THE RED NOISE POWER DENSITY ESTIMATION TECHNIQUES AND APPLICATION TO THE SOURCE SAX J2103.5+4545

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ARİF EMRE ERKOCA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

PHYSICS

JULY 2004

Approval of the Graduate School of Natural and Applied Sciences.

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Sinan Bilikmen
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Altan Baykal
Supervisor

Examining Committee Members

Prof. Dr.Ali Alpar (SABANCI UNV.) _____

Prof. Dr. Altan Baykal (METU, PHYS) _____

Prof. Dr. Ümit Kızıloğlu (METU, PHYS) _____

Prof. Dr. Nilgün Kızıloğlu (METU, PHYS) _____

Assoc. Prof. Dr. Şölen Balman (METU, PHYS) _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Arif Emre Erkoca

ABSTRACT

THE RED NOISE POWER DENSITY ESTIMATION TECHNIQUES AND APPLICATION TO THE SOURCE SAX J2103.5+4545

ERKOCA, ARİF EMRE

M.S., Department of Physics

Supervisor: Prof. Dr. Altan Baykal

July 2004, 76 pages.

In this thesis, red noise analysis techniques are presented. The necessity of the use of the window functions and the Deeter polynomial method in order to determine red noise is discussed. The method was applied to the source SAX J2103.5+4545 which showed a white torque noise with a relatively low noise strength due to its being a transient system.

Keywords: Window functions,Red Noise Determination Techniques, SAX J2103.5+4545

ÖZ

KIRMIZI GÜRÜLTÜ GÜÇ YOĞUNLUĞU BELİRLEME TEKNİKLERİ VE SAX J2103.5+4545 KAYNAĞINA UYGULAMASI

ERKOCA, ARİF EMRE

Yüksek Lisans , Fizik Bölümü

Tez Yöneticisi: Prof. Dr. Altan Baykal

Temmuz 2004, 76 sayfa.

Bu tezde, kırmızı gürültü analizi teknikleri sunulmaktadır. Kırmızı gürültü belirlemede pencere fonksiyonları ve Deeter polinom yöntemini kullanma gereklilikleri tartışılmıştır. Bu teknikler SAX J2103.5+4545 kaynağına uygulandığında beyaz dönme momenti gürültüsü düşük seviyeli olarak belirlendi.

Anahtar Sözcükler: Pencere fonksiyonları,Kırmızı Gürültü Belirleme Teknikleri, SAX J2103.5+4545

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to Prof. Dr. Altan Baykal for his help, attitude, guidance and insight throughout the research.

My great thanks are due to my mother, father and dear sister for their morale, endless help and encouragement all throughout my life.

I also would like to thank Çağdaş Sıtkı İnam, Özgür Taşkın and Elif Beklen for their patiences, help during this study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 A Brief Theory

The discovery of periodic X-ray pulsations from Cen X-3 led to a qualitative understanding of X-ray pulsars as rotating magnetized neutron stars accreting matter from a binary companion. The neutron star accretes matter either by capturing material from the stellar wind of the companion or through Roche lobe overflow of the mass donating star. The strong magnetic field controls the accretion flow close to the neutron star. Matter follows the field lines onto the magnetic poles. Mainly as the accreting material approaches the neutron star, the plasma is channeled to the magnetic polar caps, where it releases its gravitational energy as X-ray radiation; these rotating hot spots are the source of the pulsed emission. Such emission is beamed and thus produces X-ray pulsation by periodically passing through the line of sight as the neutron star rotates, if the magnetic and rotation axes of the neutron star are misaligned. Neutron stars of this type are called 'Accretion Powered Pulsars'. The resulting accretion luminosity is $L = \frac{GM_*\dot{M}}{R_*}$ which depends on the accretion rate, neutron star mass and radius.Accretion powered X-ray pulsars are classified according to the mass of companion star as either low mass X-ray binaries (LMXBs have late type or

degenerate dwarf companion with $M_c < 2.5M_\odot$) or high mass X-ray binaries (HMXB) ($M_c > 6M_\odot$). HMXB are divided into two subgroups as Be type companion star and the OB supergiant type. In these X-ray pulsars mass transfer can be seen by either disk-fed or wind-fed. In LMXB, mass transfer occur only if the companion star fills its Roche lobe and the Keplerian disk forms around the neutron star. There are only few low-mass binaries with accreting pulsars: Her X-1, 4U 1626-67, GX1+4 and GRO J1744-28. The Be systems are generally observed during transient outbursts. The episodic outbursts are often correlated with periastron passage of the neutron star in its eccentric orbit. Most OB supergiants have stellar winds driven by the radiation pressure. Vela X-1 is the best known wind-fed supergiant pulsar binary. If an accretion disk is a result of a Roche lobe overflow, much larger mass transfer rates are produced. The large persistent accretion rates in SMC X-1, Cen X-3, and LMC X-4 make them candidates for disk-fed supergiant pulsar binaries (Bildsten et al. 1997). According to the Corbet diagram (Corbet 1986), the neutron stars orbiting Roche-lobe filling supergiants have short spin and orbital periods and the opposite for the wind-fed supergiant binaries. They also show long term steady torque behavior. However, the observed accretion torque on wind-fed objects often fluctuate (even between spin-up and spin-down). The torque on the neutron star can be expressed as a specific angular momentum ($l_k = (GMr_0)^{1/2}$) added to neutron star at some radius with a certain mass accretion rate. If the accretion is from a Keplerian disk, at the inner disk edge the magnetosphere disrupts the Keplerian rotation of the disk, forcing matter to accrete along the field lines. The inner disk edge

moves inward with increasing mass accretion rate due to the relation;

$$r_0 = K\mu^{4/7}(GM)^{-1/7}\dot{M}^{-2/7}$$

(μ) is the neutron star magnetic moment and K=0.91 gives the Alfven radius for spherical accretion. Then the torque estimate is given by Ghosh & Lamb (1979) as

$$2\pi I\dot{\nu} = n(w_s)\dot{M}l_k$$

$$n(w_s) \simeq 1.4(1 - w_s/w_c)/(1 - w_s)$$

n is a dimensionless function that measures the variation of the accretion torqueas estimated by the fastness parameter

$$w_s = \nu/\nu_K(r_0) = (r_0/r_{co})^{3/2} \propto \dot{M}^{-3/7}$$

where r_{co} is the corotation radius at which the centrifugal forces balance the gravitational forces, and w_c is the critical fastness parameterat which the accretion torque is expected to vanish. The accretion torque,in fact, is the sum of the torque produced by the accretion of matter that falls onto the star (mechanical torque) and the torque contributed by the twisted magnetic field lines of the star that interact with the outer parts of the disk (magnetic torque). The mechanical torque always acts to spin up a star rotating in the same sense as the disk flow, whereas torque from the magnetic stresses can have either sign;negative if the lines thread the disk outside the corotation radius. The net torque will cause spin-up if the neutron star is rotating slowly ($w_s < w_c$) in the same sense as the circulation in the disk. Even if the neutron star is rotating in the same sense

as the disk flow, the torque will be in the direction of spin-down if the nuetron star is rotating too rapidly ($w_s > w_c$). The accreted material will produce X-ray luminosity at the neutron star surface at the rate

$$L = GM\dot{M}/R$$

The rate of spin-up is related to the X-ray intensity through

$$\dot{\nu} \propto n(w_s)L^{6/7}$$

When $\dot{M}$ becomes small enough the luminosity decreases and the magnetospheric radius exceeds the corotation radius. Then, some of the material will be accelerated to super-Keplerian velocities and will not be easily accreted (accretion is centrifugally inhibited) and material may become attached to the field lines and flung away, removing angular momentum and causing the star to spin down which is called propeller effect (Illarrionov & Sunyaev, 1975). So a slight change in the mass accretion rate can be considered to be a reason for torque reversals. Also any external torque fluctuation creates a difference between angular velocity of the core superfluid and crust. The two components are coupled by a mutual friction at crust core coupling times of the order 400 to 10^4 periods. In time scales shorter than the crust-core coupling time scales crust angular velocity derivative has larger magnitude. As a model torque variations by a sum of noise processes were represented by Lamb et.al.(1978 b). They concentrated on two different types of torque fluctuation models; one corresponds to "white noise", where the power of torque variation is independent of frequency, which represents the series of sudden transfers of angular momentum to the neutron star crust; other

describes sudden jumps in the rate at which angular momentum is transferred to the neutron star crust, and hence represents red torque noise where the power of torque variation depends inversely on the second power of the frequency. They also described the response of a neutron star to these variations by using a two component model.

The conventional wisdom suggests that Roche-lobe overflow and subsequent disk accretion is supposed to carry positive angular momentum to the neutron star and spin it up (Pringle and Rees 1972). On the other hand for wind accretion, the flow is supposed to have lower specific angular momentum and can approach the neutron star with reversals in the spin direction. Hence, it is expected that disk accretion should lead to a stronger spin-up of the neutron star and less torque noise in contrast to the wind accreting types. Observations showed that there is a strong correlation between the noise strength and the X-ray luminosity of accreting X-ray pulsars ($S \sim L_{37} 10^{-17} rad^2 s^{-3}$) ,where L_{37} is the X-ray luminosity in $10^{37} ergs^{-1}$. An interesting feature of this relation is the lack of any distinction between the sources believed to be disk type and wind type (Baykal 1993). This is due to the fact that both of these depend on the rate of the torque noise events and regardless of the sign of the torque, the mass transfer always contribute positively to the luminosity of the neutron star. In this study a list of some accretion powered X-ray binaries are presented, with their noise strengths and X-ray luminosities (Bildsten et al. 1997, Baykal et al. 1993, Baykal et al. 2002 and this study).

Early studies of four pulsating sources X-ray sources reveal pulse periods which

Table 1.1: Luminosities and Noise strengths of Accretion powered X-ray pulsars

Name	Log L(erg/sec)	Log S(rad^2/sec^3)
SMC X-1	(38.70 , 38.78)	(-13.52 , -12.33)
HER X-1	(36.85 , 37.30)	(-16.92 , -16.40)
4U 0115+634	(36.90 , 37.48)	(-17.70 , -16.68)
CEN X-3	(37.70 , 38.00)	(-15.80 , -13.70)
4U 1627-67	(36.90 , 37.48)	(-16.32 , -16.19)
LMC X-4	(38.60 , 38.85)	(-17.00 , -15.80)
OAO 1657-41	(35.60 , 37.00)	(-15.40 , -14.10)
A 0535+26	(37.30 , 37.30)	(-16.70 , -16.36)
GX1+4	(37.60 , 38.00)	(-13.77 , -13.05)
VELA X-1	(35.90 , 36.78)	(-18.10 , -17.40)
4U 1145-619	(35.00 , 36.78)	(-17.40 , -16.85)
E1145.1-614	(34.48 , 36.48)	(-18.15 , -17.15)
4U1538-52	(36.60 , 36.60)	(-19.10 , -17.49)
GX301-2	(36.00 , 37.00)	(-16.62 , -15.09)
X-PER	(33.60 , 34.00)	(-19.40 , -18.42)
SAXJ 2103.5+4545	(34.78 , 36.00)	(-20.27 , -19.0)

are not smoothly decreasing; and in fact are noisier than the Crab: Her X-1, Cen X-3 , Vela X-1 and X Per . These variations imply that the angular momentum of the crust is fluctuating. Because the neutron star is a dynamic system, such fluctuations could be caused by the (external) torque exerted by the accreting matter(Lamb et al. 1978) or internal torque exerted on the crust by neutron superfluid. Neither of these torques is expected to be smooth. Accretion theory suggests that the accretion torque contains a fluctuating component while the response of the superfluid even to a smooth accretion torque may contain a stochastic element. Consequently, the pulse period fluctuations in accreting

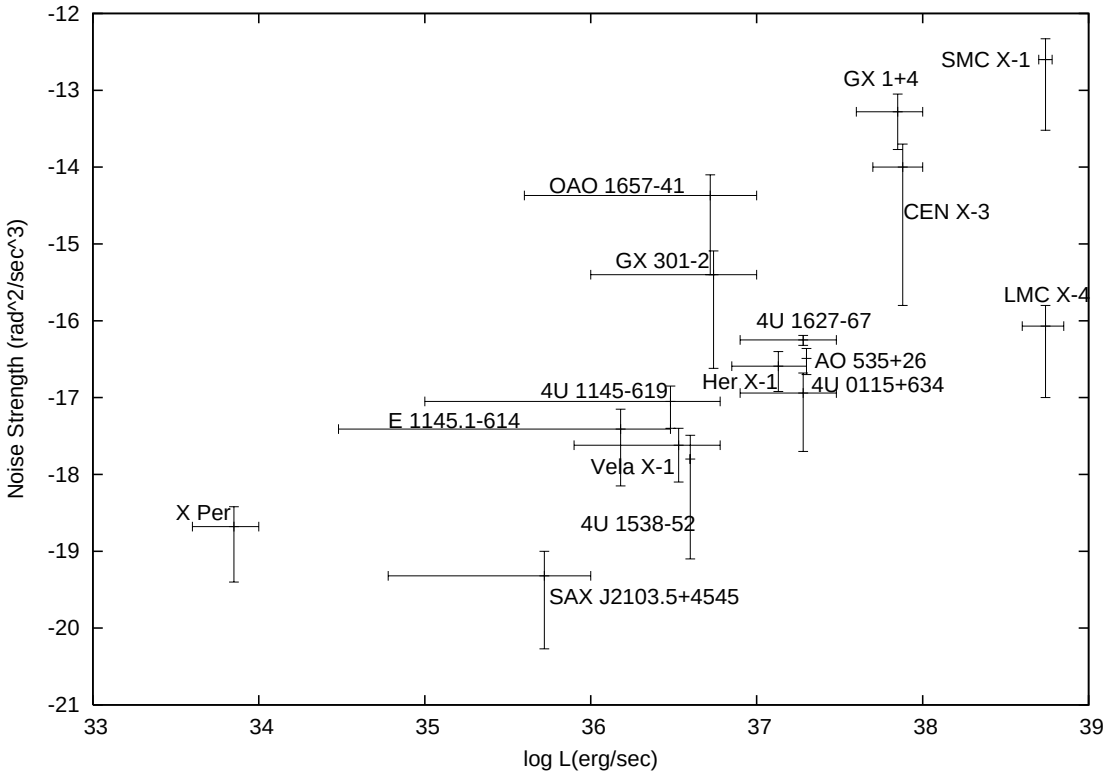

Figure 1.1: Noise Strength versus Luminosity of Accretion powered X-ray Pulsars

neutron stars are produced by torques originating outside and inside the object. The external torque is closely related to the mass accretion and the internal torque depends on the coupling between the superfluid interior and the solid outer crust. Filtered torque variations by the mutual interactions of the internal structures results the observed changes on the pulse period. A statistical analysis of such fluctuations was performed after the discovery of the Crab pulsar (Boynton et. al., 1972; Groth, 1975a; Cordes, 1980) indicating that the small scale variations can be expressed with white or red noise models. Soon afterwards, a theoretical description of torque variations in terms of power spectral analysis was developed

by Lamb et. al. (1978 a,b). The pulse timing analysis techniques has been developed to study the fluctuations in the rotation rate of a neutron star. (Deeter & Boynton 1982; Deeter 1984). They were applied to Vela X-1 and Her X-1 using the data obtained by HEAO-1 and UHURU (Boynton 1981, Deeter 1981, Boynton et al. 1984, Deeter et al. 1989). The results showed that the angular velocity time series of these sources can be modelled as a random walk (or white noise in the angular accelerations). Also the restless behaviour has been observed in the rapidly spinning down Crab Pulsar (Groth 1975a) and possibly in several other pulsars (Cordes & Helfand 1980). A similar analysis has also been applied to derive the power spectra of timing noise in radio pulsars (Cordes 1980, Cordes & Helfand 1980, and Alpar et. al. 1986).

1.2 About This Study

This study mostly deals with the methodology of determining the red noise processes present in the timing data. We have used some routines written in C language at Numerical Recipes such as during the generation of a random number series, finding the linear least square fit, cubic spline, and FFT (Fast Fourier Transform) routines are also used while discrete Fourier Transforming. The other routines were written by ourselves (some of which are present in the appendix). In Chapter 2 the need for the use of window functions are presented. In Chapter 3, time-domain properties and the non-stationarity of a red noise process is introduced. In Chapter 4, low-resolution technique which mixes the time-domain properties and the frequency-domain properties in determining the

red noise is presented. In all chapters, results of our computer simulations are also present. In the fifth chapter, an application to the phase residual of source SAXJ2103.5+4545 is shown. Finally, in chapter 6 we discuss our results.

CHAPTER 2

FOURIER TECHNIQUES AND POWER SPECTRUM

In most of the scientific and engineering problems, power spectrum analysis are required. In this sense, Fourier techniques are widely used.

2.1 Fourier Transform

A Fourier Transform gives a decomposition of a function introduced in time domain into sine waves in frequency domain. Two types of transforms can be considered; continuos and discrete. The continuous transform decomposes an infinitely extended continuous function $x(t)(-\infty \leq t \leq \infty)$ into an infinite number of sine waves:

$$a(f) = \int_{-\infty}^{\infty} x(t) e^{2\pi i f t} dt$$

$$x(t) = \int_{-\infty}^{\infty} a(f) e^{-2\pi i f t} dt$$

In the real world the data are neither infinitely extended nor continuous, so discrete transform is used.

$$a_j = \sum_{j=-\frac{N}{2}}^{\frac{N}{2}} x_k e^{\frac{2\pi i j k}{N}}$$

$$x_k = \frac{1}{N} \sum_{j=-\frac{N}{2}}^{\frac{N}{2}} a_j e^{\frac{-2\pi i j k}{N}}$$

where $k = 0, ..., N$ and $j = -N/2, ..., N/2 - 1$.So, the discrete Fourier transform decomposes the time series into N sine waves.If the time series is an equidistant one of length T, so x_k refers to a time $t_k = \frac{kT}{N}$.Then, the transformation is an equidistant frequency series. Each j refers to a frequency $w_j = 2\pi j/T$. So it is clear from this construction that how we can write the exponential factors in the transformation pair.

It is interesting to note that inserting N numbers as input, N numbers as output are generated.How we limit the maximum analysis frequency strongly depends on the sampling interval, defined as $\Delta = T/N$. This highest frequency, equal to half the sampling frequency, $f_{\frac{N}{2}} = \frac{1}{2\Delta}$, is called the Nyquist frequency. Hence, $a_{-j} = a_{N-j}$ that is Fourier Transform coefficients are periodic in j with period N. With this convention in mind, one can let j vary from 0 to $N - 1$. So we can work with only positive frequencies. Mathematically speaking, in general, the coefficients can be independent unless we require certian constraints such as a real time series which is the situation in reality. Then with a constraint of a real time series; it is obvious that $|a_{-j}| = |a_j|$ which shows the correlation between the coefficients. Then it is wise to use one-sided power spectrum definition for the analysis;

$$P_j = \frac{2}{a_0}|a_j|^2 \tag{2.1}$$

where $j = 0, ...N/2$. An important feature is observed that power is a quantity that can be calculated with both of the transform pairs (Persavel's theorem). This

property resembles the conservation of the length of a vector in space rotations.

$$\int |x(t)|^2 dt = \int |a(f)|^2 df$$

or for the discrete case

$$\sum_{k}^{N-1} |x_k|^2 = \frac{1}{N} \sum_{j}^{N-1} |a_j|^2$$

2.2 The relation between the Discrete and Continuous Fourier Transform

In order to obtain a finite and discretely sampled time series from a continuous one, double multiplication must be applied. The first multiplication is with a "window function"

$$w(t) = \begin{cases} 1, & 0 \leq t \leq T \\ 0, & \text{otherwise.} \end{cases}$$

which makes the series finite.(with the above form it is also called "square window") Second multiplication is with a "sampling function" which discretizes the series;

$$i(t) = \sum_{k=-\infty}^{\infty} \delta \left(t - \frac{kT}{N} \right).$$

Consequently, the discrete series, x_k can be reached after two multiplications $x(t)w(t)i(t)$. The Fourier Transform of the resulting series can be found easily by using the Convolution Theorem which states that the Fourier Transform of the product of two functions is the convolution of the corresponding transforms (van der Klis,M. 1988). For simplicity let's show the transform pairs with $\Longleftrightarrow$, then the convolution theorem indicates that;

$$x(t) \iff a(f)$$

$$y(t) \iff b(f)$$

$$x(t)y(t) \iff a(f) * b(f) = \int a(f')b(f-f')df'$$

Most of other types of transform pairs can be found from the Campbell and Foster Table.

Then we need the Fourier Transforms of window and sampling functions. They are found as;

$$W(f) = \sin(\pi f T)/\pi f.$$

for a symmetric w(t), and

$$I(f) = \frac{N}{T} \sum_{l=-\infty}^{\infty} \delta\left(f - l\frac{N}{T}\right)$$

2.2.1 Aliasing

The convolution of an arbitrary function with a delta function at f_0 is a shifted version of the original function; $a(f) * \delta(f - f_0) = a(f - f_0)$. So it is obvious that convolution with the Fourier Transform of the sampling function results in a function in frequency space which repeats itself every N/T, twice the Nyquist frequency,frequency units. Since the time series are real, the transforms are also symmetric around $f = 0$. Then, this causes features with a frequency exceeding the Nyquist frequency by f_x (i.e. located at $f = f_{N/2} + f_x$) to also appear at a frequency $f_{N/2} - f_x$, a phenomenon known as aliasing. The reflected feature is called the alias of the original one (van der Klis 1988).

2.2.2 Windowing

Now let's suppose we have a noise process whose Fourier Transform is given by $F(f) = \frac{C}{f^r}$ (like the rth order red noise) where C is a constant and 2r is the power-law index. Then, due to the use of a window function, the convolution of its transform with that of the original noise process must be considered before finding the resultant transform. This can be written as;

$$F_w(f) = \int \frac{C}{f'^r} \frac{\sin[\pi(f - f')T]}{\pi(f - f')} df'$$

Consequently, the problem called "leakage" is observed due to the large, but finite response at the low frequencies. The sine function inside the integrand have considerable sidelobe effects, so each power estimate is dominated by the response at the low analysis frequencies. The sidelobe characteristic can easily be guessed since there is a sharp increase from 0 to 1, so apart from the peak at zero frequency the presence of a considerable response at larger frequencies is obvious. This effect flattens the power spectrum and unfortunately the expected power law can not be observed. Because of this fact square window is sometimes called "no window". In order to recover this, the window function must be chosen to sufficiently reduce the sidelobe response of the power density estimator so that leakage is not a major problem (Scott et.al. 2003). Practically, every function that rises from zero to a peak and then falls again has been named after someone (Press 1992). A few of the more common are as follows:

$$w_j = 1 - \left| \frac{j - \frac{N}{2}}{\frac{N}{2}} \right| \text{ Bartlett}$$

$$w_j = 1 - \left(\frac{j - \frac{N}{2}}{\frac{N}{2}} \right)^2 \text{ Welch}$$

$$w_j = \frac{1}{2} \left[1 - \cos \left(\frac{2\pi j}{N} \right) \right] \text{ Hann.}$$

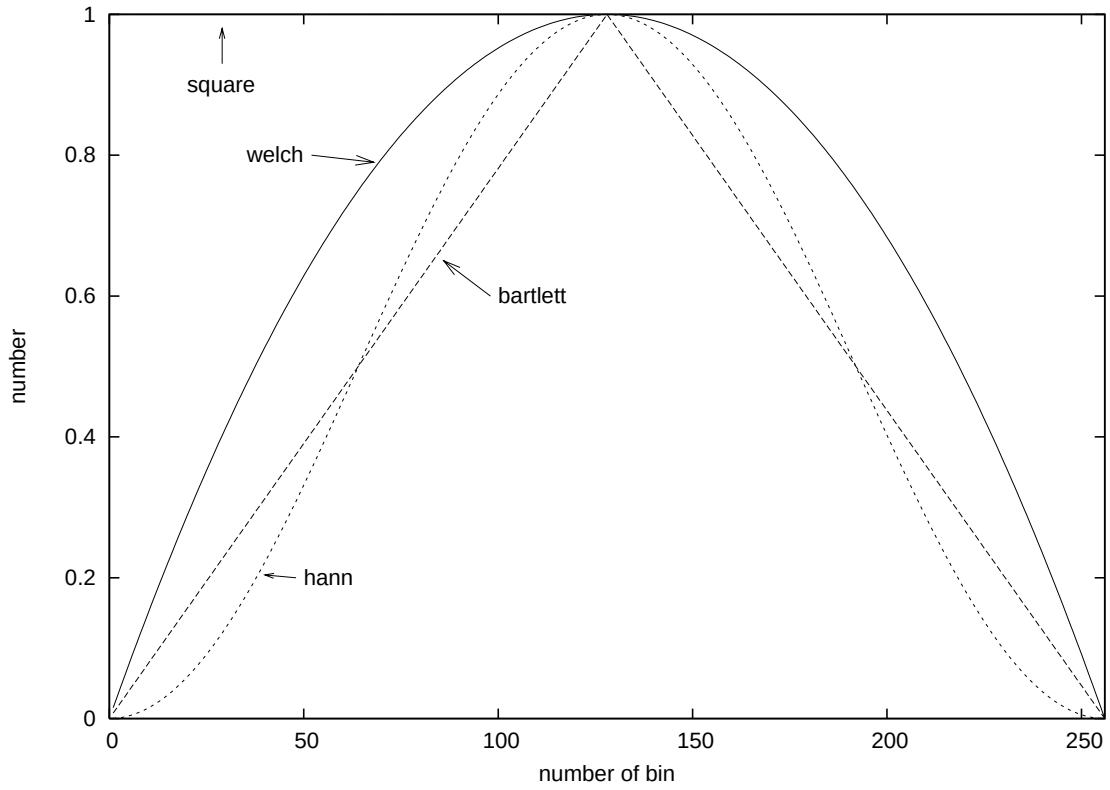

Figure 2.1: Window Functions

2.3 Simulations

In order to have simulations we should construct a kind of white noise process which is ,by definition, a stationary process ($< \epsilon(t)\epsilon(t+\tau) >= {\sigma_\epsilon}^2 \delta(\tau)$) which has

a flat power spectrum. $S = {\sigma_\epsilon}^2$ is called the noise strength and $< |F_\epsilon(f)|^2 >= S$ from the Persavel's theorem, where $F_\epsilon(f)$ is the Fourier Transform of $\epsilon(t)$. First integral of a white noise process is a first order red noise process with a power-law of form P$\sim f^{-2}$; repeating the same procedure, after each integration a factor of f^{-2} is multiplied in the power spectrum. The Figure 2.2 shows the time series of one of the white noise realizations having 100000 points produced by a gaussian deviate random number generator and its histogram. Also the gaussian fit to this histogram and the power spectrum of the realization are also presented. We found the noise strength of that noise as 0.999 from the power spectrum and the standart deviation from the histogram as 0.996 whose square ($\sim$ 0.992) is also a test for the noise strength. These two close numbers suggest us that we can generate a normally deviated unit strength white noise process.

In order to test the strength of windowing, with the help of a generated normal deviated white noise process, red noise processes with different orders were generated with repeated integrations and different subsets from these realizations were taken, multiplied with window functions as described above, Fourier Transformed; logP versus logf graphs were drawn and the slopes were found and then averaged.

According to the simulations, it is observed that Hann window is a good choice to recover the red noise processes; particularly the low order ones, rather than other window functions. It is also obvious that square window(no window) is the worst choice. Regardless of the order of the noise process, the slope of the logP

Table 2.1: Power Law index (m=2r) estimates for the first three order red noises

window function	r=1	r=2	r=3
hann	2.065∓0.001	4.311∓0.076	6.78∓0.02
welch	2.141∓0.019	3.570∓0.002	3.563∓0.001
bartlett	2.12∓0.01	3.685∓0.002	3.718∓0.007
no window	1.779∓0.003	1.787∓0.002	1.789∓0.001

versus logf graphs saturates around -2 when square window is used. Also for other window functions this method has limitations to recover higher order red noise processes($r > 3$). Also during our simulations it is observed that removing a low degree polynomial from the noise recovers the spectrum in a more convenient way. In addition, the number of points used in the process significantly affects the results. For the first order red noise, the results are not significantly distinct. However, starting with the second order red noise, the number of points and the removal of a polynomial trend have great significance. The following results are after removing cubic polynomial trends and using a larger window length (i.e. more points). In order to determine the second order red noise process, again hann window is the best choice to be applied whereas here are presented the best results obtained from the simulations. For several other noise processes some kinds of deviations of the order 20% from the expected power-law index are also occured and even the histograms of power-law indices obtained from different

noise realizations with the same order and equal length show significant changes.

Table 2.2: Power Law index (m=2r) estimates for the first three order red noises after cubic polynomial removal

window function	r=1	r=2	r=3
hann	2.057∓0.001	4.011∓0.075	6.57∓0.01
welch	2.109∓0.017	3.430∓0.001	3.563∓0.001
bartlett	2.081∓0.008	3.582∓0.002	3.643∓0.005
no window	1.774∓0.003	1.785∓0.002	1.788∓0.001

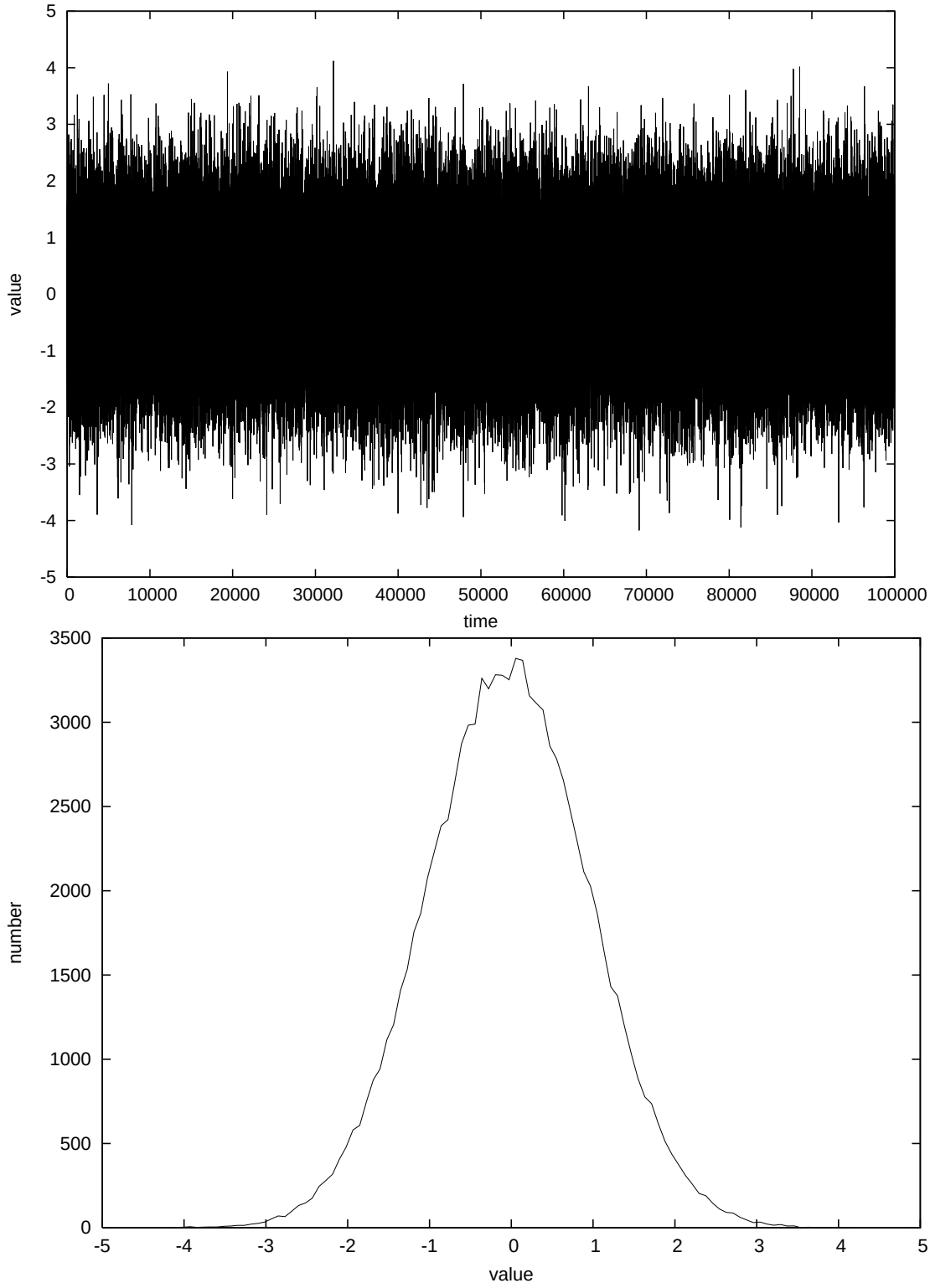

Figure 2.2: White Noise realization (upper panel) and its histogram(lower panel)

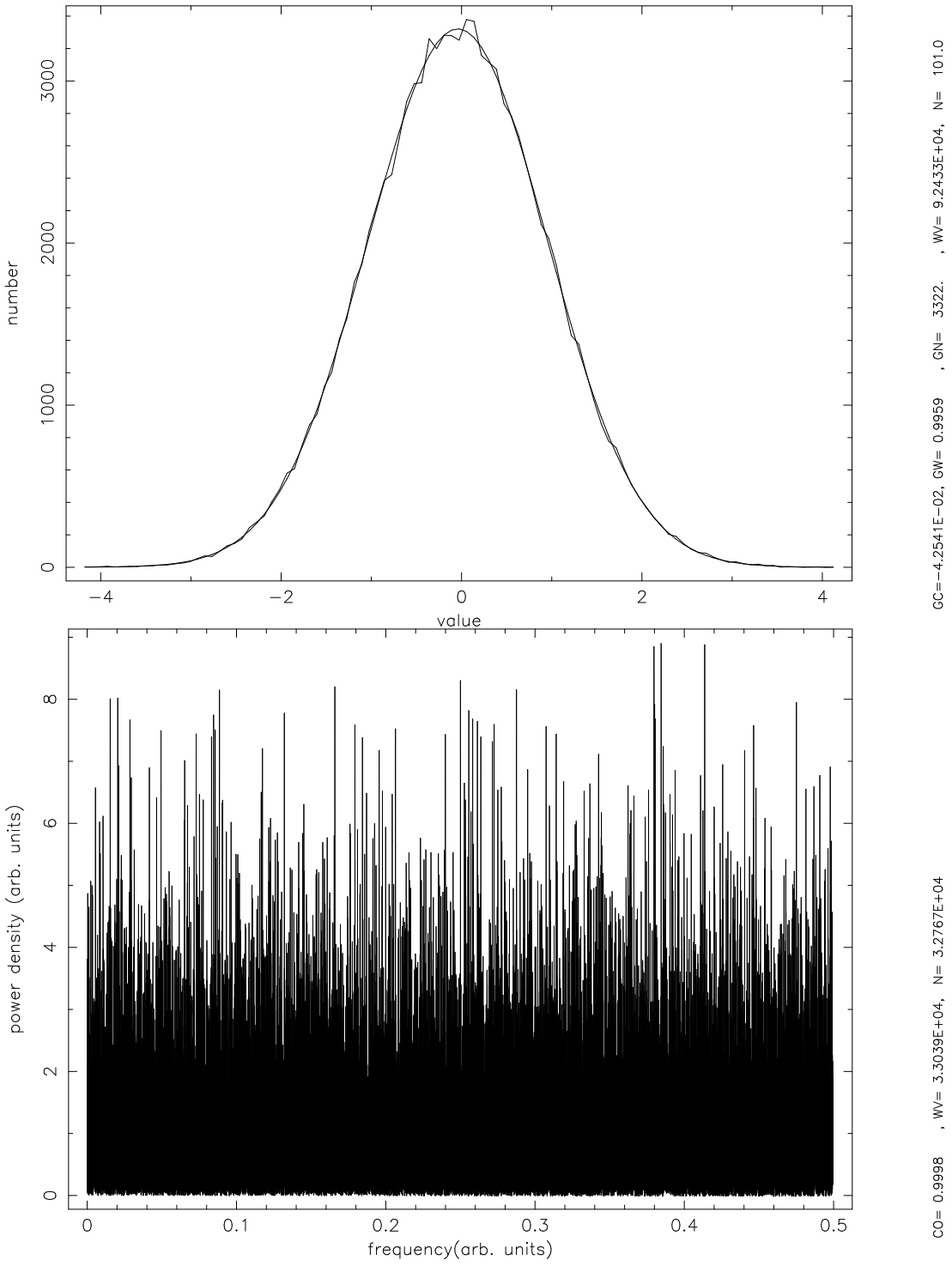

Figure 2.3: Gaussion Fit to the histogram of a white noise process(upper panel). Power spectrum of the white noise(lower panel)

CHAPTER 3

TIME DOMAIN PROPERTIES OF NOISE PROCESSES

3.1 Generation of Red Power-Law Noise

As it is also stated in the previous chapters, power-law noise(PLN) processes with negative even integer spectral indices can be produced by repeated integration of a white noise process $\epsilon(t)$. For instance, a random walk, with a $1/f^2$ power spectrum is defined as (Scott et al. 2003);

$$r_2(t) = \int_{-\infty}^{t} \epsilon(t')dt'.$$

It is also easy to show that there is an alternative way to define the above random walk process from the same white noise process by the convolution of white noise with the unit step function, $H(t)$:

$$r_2(t) = \epsilon(t) * H(t) = \int_{-\infty}^{+\infty} \epsilon(t')H(t-t')dt' = \int_{-\infty}^{t} \epsilon(t')dt'$$

where $H(t)$ is the standard Heaviside step function defined as : $H(t) = 1; t \geq 0$ and $H(t) = 0; t < 0$. In a similar way, $1/f^4$ noise can be made by double convolution with the step function.

$$H(t) * (H(t) * \epsilon(t)) = (H(t) * H(t)) * \epsilon(t).$$

The self convolution of $H(t)$ results in a ramp function; $tH(t)$.

$$\begin{aligned} H(t) * H(t) &= \int_{-\infty}^{+\infty} H(t')H(t-t')dt' \\ &= \int_{0}^{+\infty} H(t-t')dt' \\ &= \int_{0}^{t} dt' \\ &= tH(t) \end{aligned}$$

Then $r_4(t)$ can be generated directly as the convolution of $tH(t)$ with white noise:

$$\begin{aligned} r_4(t) &= (tH(t)) * \epsilon(t) \\ &= \int_{-\infty}^{t} (t-t')\epsilon(t')dt'. \end{aligned}$$

Similarly, the convolution of $\epsilon(t)$ with $t^2H(t)$ will generate $1/f^6$ noise. In general, even-integer red power-law noise can be generated directly from white noise (Scott, Finger & Wilson 2003) as:

$$r_m(t) = \frac{1}{k!}\int_{-\infty}^{t} (t-t')^k \epsilon(t')dt' \tag{3.1}$$

where m=2(k+1) and is called the spectral index. For integer values of k, m=2,4,6,... So, without the need of repeated integration of a white noise process, we can generate a red noise process with a desired spectral index,m, by using the equation 3.1. This formulation for $r_m(t)$ can also be generalized for noninteger k values. An example of this will be given shortly. The above equation is in the form of a general linear noise process

$$r_m(t) = \int_{-\infty}^{t} \omega(t-t')\epsilon(t')dt'$$

in which case a weight function of the form $w(t)H(t)$ is convolved with the white noise.It is then easy to find PDS by simply multiplying the fourier transform of the weight function with that of white noise. When the discrete form of a white noise process is concerned, the weight function can be viewed as a "step" so that the noise process can be considered as an accumulation of random steps. The duration of the step is important in order to say that whether the noise process is stationary or not(i.e. whether two values of the noise process separated by a sufficiently large time interval are correlated or not). For red power-law noise processes a correlation exists between a given value and all past values of the time series even if two values of the basis white noise time-series are uncorrelated at all times. So red power-law noise is intrinsically nonstationary. Using the Fourier Transform Techniques and the Convolution Theorem, it is easy to show that the PDS of a red power-law noise can be found as:

$$P(f) = \frac{\sigma_\epsilon^2}{2\pi f^{2(k+1)}} \tag{3.2}$$

where σ_ϵ^2 is the mean power level of the underlying white noise.

3.2 Time Domain Properties of Red Power-Law Noise

The time series of any kind of noise process can be divided into two prts: a physically realizable part starting at time $t = 0$ and a part which refers to the past starting at time $t = T$ (T can range to $-\infty$) and the latter contributes a smooth function to the time series after $t = 0$.

$$r_m(t) = \frac{1}{k!}\int_T^0 (t - t')^k \epsilon(t')dt' + \frac{1}{k!}\int_0^t (t - t')^k \epsilon(t')dt'$$

and m=2(k+1). As an example, for the random walk (k=0; m=2), the smooth function will be a constant:

$$\int_T^0 \epsilon(t')dt' = C_0$$

Similarly, the smooth function will be a linear trend for the time series of $1/f^4$ type noise and a quadratic trend for that of a $1/f^6$ type noise process. These smooth functional behaviours are also presented in this study for the first three integrals of white noise.

3.2.1 Ensemble Averaged Expectation and Variance

Since the generation of a red power-law noise depends on a white noise process, the statistics will be directly related to the white noise statistics. The expectation $\langle r_m(t)\rangle$ will be given by:

$$\langle r_m(t)\rangle = \frac{1}{k!}\langle\int_{-\infty}^{t}(t-t')^k\epsilon(t')dt'\rangle = \langle\epsilon(t)\rangle\frac{1}{k!}\int_{-\infty}^{t}(t-t')^k dt'.$$

If this time series is divided into two finite portions as before the result will be:

$$\langle r_m(t)\rangle = \frac{\langle\epsilon(t)\rangle}{(k+1)!}(t+T)^{k+1}.$$

Then a zero mean white noise ($\langle\epsilon(t)\rangle = 0$) time series implies that $\langle r_m(t)\rangle = 0$ as well. The variance of $r_m(t)$ as a function of time can also be calculated from:

$$\begin{aligned}\langle r_m(t)^2\rangle &= \left(\frac{1}{k!}\right)^2\left\langle\left(\int_{-\infty}^{t}(t-t')^k\epsilon(t')dt'\right)^2\right\rangle \\ &\vdots \\ &= \left(\frac{\sigma_\epsilon}{k!}\right)^2\int_{-\infty}^{t}(t-t')^{2k}dt'\end{aligned}$$

where for white noise $\langle \epsilon(t)\epsilon(t') \rangle = {\sigma_\epsilon}^2 \delta(t-t')$ is used. $\delta(t)$ is the Dirac delta function. If, again, the integral is divided into two finite portions the variance is finite and reduces to:

$$\langle r_m(t)^2 \rangle = \frac{1}{(2k+1)} \left(\frac{\sigma_\epsilon}{k!} \right)^2 (t+T)^{2k+1} \propto (t+T)^{m-1}$$

The physical unit of noise strength (S=${\sigma_\epsilon}^2$) can be seen easily from the above relation. If $r_m(t)$ has the unit of radian, for example, it is $rad^2 sec^{-(2k+1)}$. So if we had a second order red noise in phase (i.e. k=1 or m=4), the unit of noise strength would be $rad^2 sec^{-3}$.

The autocovariance can also be found in a similar manner:

$$\langle r_m(t) r_m(t+\tau) \rangle = \left(\frac{\sigma_\epsilon}{k!} \right)^2 \left(\int_{-\infty}^{t} (t-t')^k (t+\tau-t')^k dt' \right)$$

with $\tau \geq 0$. Hence the autocovariance function depends on both the current time t and the lag τ.After a long algebraic calculation it can also be proven that the fourier transform of this autocovariance function (from τ space to frequency space) approach the power-law form in equation 3.2.

3.2.2 Examples

For the three cases (k=0,1,2 i.e. for $1/f^2$, $1/f^4$, $1/f^6$ power-law noise), after removing the past smooth function (i.e. taking the lower limits of integrals as 0 rather than $-\infty$), the autocovariance functions can be found as:

$$< r_2(t) r_2(t+\tau) >= {\sigma_\epsilon}^2 t$$

$$< r_4(t) r_4(t+\tau) >= \frac{1}{6} {\sigma_\epsilon}^2 t^2 (2t+3\tau)$$

$$< r_6(t) r_6(t+\tau) >= \frac{1}{120} {\sigma_\epsilon}^2 t^3 \left(6t^2 + 15t\tau + 10\tau^2\right)$$

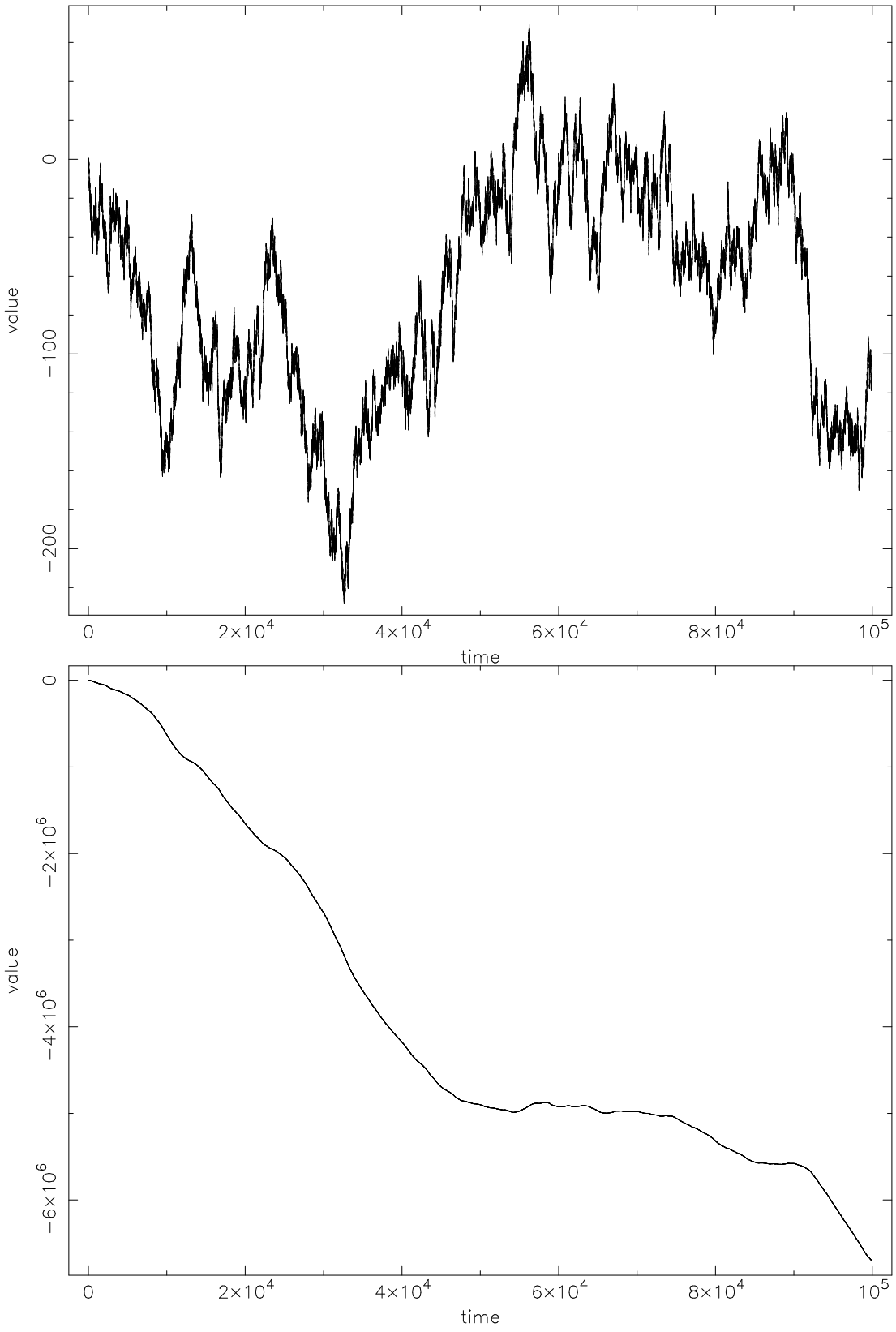

Figure 3.1: Simulated Red Noise Time Series for r=1(upper panel), r=2(lower panel)

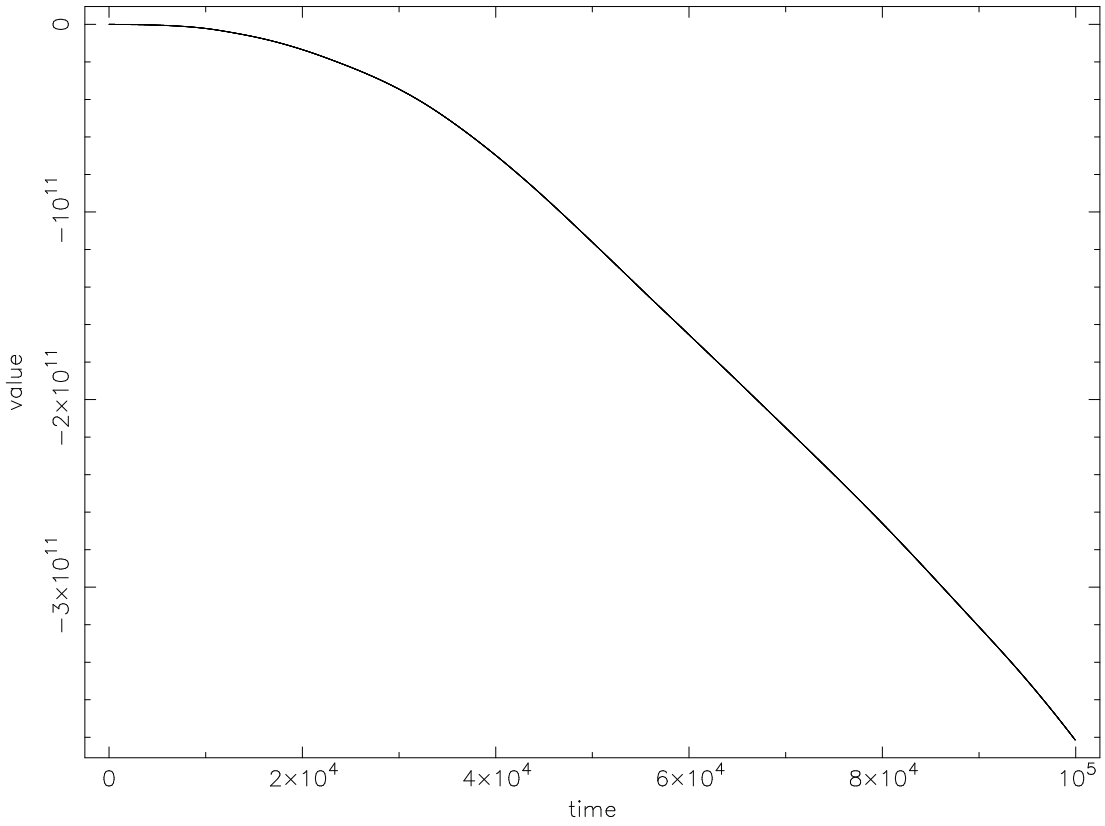

Figure 3.2: Simulated Red Noise Time Series for r=3

CHAPTER 4

LOW RESOLUTION TECHNIQUE

In the rapidly spinning down Crab pulsar a kind of restless behaviour was observed (Groth 1975a) and a new method of analysis for time series like Crab pulsar timing data was presented (Groth 1975b). Same behaviour was also observed in several other radio pulsars (Cordes and Helfand 1980). This phenomenon is thought to arise from fluctuating internal torques associated with the nature of coupling between the neutron star crust and the superfluid interior. In the case of accreting neutron stars, rotational spin-up suggests the possibility of similar behaviour from internal torques, but enhanced restlessness may result from fluctuating external torques associated with the process of matter accretion. Theoretical studies showed that the response of a neutron star to fluctuating torques can be quiet complex, however they can be described by a simple noise process which has a definite power spectrum.The complexity can also be revealed through the power spectrum of pulse phase, pulse frequency or frequency derivative. Not only there are problems including the non-equispaced sampling of time series arising from real astronomical data, but there are also problems due to the nature of the noise process present in the data.

4.1 Nonstationarity

For the red noise determinations the most important problem is the nonstationarity. By definition, the autocorrelation function $< x(t_1)x(t_2) >$ of a stationary process $x(t)$ is a function of one variable, $t_1 - t_2$. So, the process is in some sense independent of local time; only the time difference between the data points are important. Since the power spectrum is the cosine transform of the autocorrelation fuction (Freeman 1958 page 35), then it is only a fuction of one frequency. For a non-stationary process, the autocorrelation function not only depends on the time difference; but it also depends on the local time through $t_1 + t_2$ term. So, a useful power spectrum is not obtained.

After removing the secular trends from the data we can end up with only the noise component which is discussed by Groth (Groth 1975b). He expands the noise process with orthonormal polynomials (Legendre polynomials, P_j, where j is the degree of the polynomial). Covariances of the expansion coefficients give rough estimates of the power spectrum; however non-stationarity must be removed. Non-stationarity is the characteristic of a red noise process. Three types of noise models can be considered; random walk in phase,phase noise (PN); random walk in frequency, frequency noise (FN); random walk in frequency derivative, slowing-down noise (SN).

4.1.1 PN,FN,SN

With the PN model,

$$\phi = \sum_i \delta\phi_i H(t - t_i)$$

where t_i is the time of ith step, $\delta\phi_i$ is the size of the ith step, $H(t)$ is the unit step function, the autocorrelation function is found to be

$$< \phi(t_1)\phi(t_2) >= R < \delta\phi^2 > t_<$$

R is the average rate of steps; RT is the number of steps (T is the total time span). $< \delta\phi^2 >$ is the second moment of the step size distribution. $t_<$ is the smaller of t_1 and t_2. In a more symmetric form;

$$t_< = t_a - \tau/2$$

where $t_a = (t_1+t_2)/2$ and $\tau = |t_1-t_2|$. So t_a term brings the non-stationarity. It is found that the non-stationary part only contributes to the first three covariances ($j < 1$,i.e. non-stationarity is confined to a few of the expansion coefficients rather than spread out among all the coefficients). So, for $j > 1$, the power spectrum which is due to only the stationary part can be calculated with inserting $f_j = j/2T$ since jth degree Legendre polynomial has j real zeros between $0 < t < T$. Then f_j corresponds to "quasifrequency". In a way this resembles the Fourier transforming. In the end, $Power \sim {f_j}^{-2}$.

Similarly for FN stationarity is found for $j > 3$, and $j > 5$ for SN. And for FN, $Power \sim {f_j}^{-4}$. ; for SN $Power \sim {f_j}^{-6}$. FN model is given as;

$$\phi = \sum_i \delta\nu_i (t - t_i) H(t - t_i)$$

SN model is given as;

$$\phi = \sum_i (\delta\nu_t)_i (t - t_i)^2 H(t - t_i)$$

Hence, the time derivative of PN brings delta functions in the summation since the derivative of step function is a delta function. The power spectra of that process turns out to be independent on the frequency and it is flat; then this is called white noise. Most of the time the term "order" is used. If power spectrum of a noise process goes as f^{-2r}, this process is called "rth order red noise" (i.e. it is the rth integral of a white noise process).

4.2 Polynomial Estimator Method

This method takes its roots from the previous section and from the previous chapters. Up to now, we have encountered two important problems while trying to find the power spectrum of a red noise; one is the "leakage problem", the other is the "non-stationarity". Both of them are solved by this technique. Sampling our time series with a function which is considered to take care of the "leakage" problem;the necessity to remove the non-stationarity and the use of Legendre polynomials are the basic clues of this method.

The power estimate can easily be written as (Deeter et al. 1982)

$$P = \left[\int_a^b g(t)x(t)dt \right]^2 \tag{4.1}$$

Here x(t) is a time series, g(t) is a sampling function. If x(t) is a stationary random process like a white noise, the ensemble average of the power spectrum

can be found as (Freeman 1958 page 105)

$$< P >= \int |G(f)|^2 S(f) df \tag{4.2}$$

G(f) is the transfer function, fourier transform of g(t). S(f) is the power density which obeys, for a rth order red noise, $S = K_r(2\pi f)^{-2r}$. Here this time K_r is what we call "noise strength". Again the leakage problem must be solved. This arises since the expectation of power estimate may not be dominated by power at the centroid of G(f) but rather by power at the lower frequency sidebands, if the spectrum is red and sufficiently steep. This problem can be avoided by requiring G(f) to go to zero (for small f) at least as rapidly as $f^{r-\frac{1}{2}}$. It is assured if the first $r + \frac{1}{2}$ terms in the Maclaurin expansion of G(f) vanish. G(f) can be written as the Fourier transform of g(t), the terms in the Maclaurin expansion can be found by taking the derivatives with respect to frequency and after setting f equal to zero. After each derivative there appears a factor of t coming from the derivative of the exponential. Then the requirement here produces the following so called r moment conditions.

$$\int_a^b g(t) t^i dt = 0 \tag{4.3}$$

for $0 \leq i < r$.Here there seems to be an ambiguity that we require x(t) to be stationary, however we can take S(f) to be a power spectrum of a red noise process. In fact the implemented moment conditions also solves the puzzle; since a kind of sampling function which satisfies the above moment conditions automatically removes the non-stationarity in the noise (remember that non-stationarity only contributes to the low degree polynomials depending on the order of noise, Groth

1975b). As might be expected, imposing the moment conditions to order r on sampling function is closely related to removing a polynomial trend of degree r-1 from the observed data. Thus any sampling function g(t) applied to the data with a polynomial of degree r-1 removed will produce a valid power estimate for rth order red noise. For example, to estimate a second order red noise cubic, quartic,... polynomials can be used as sampling functions. In fact in our study we have used Legendre-like polynomials such as what Groth had used.

Integrating the equation 4.1 by parts,the power estimate in the time domain becomes

$$P = \left[\int_a^b g^{-r}(t)x^r(t)dt\right]^2 \tag{4.4}$$

$g^{-r}(t)$ is the particular rth integral of g(t) and $x^r(t)$ (rth derivative of x(t)) is the underlying white noise process if x(t) corresponds to a rth order red noise. Considering 4.2, a similar form of average power estimate can be obtained

$$< P >= K_r \int_a^b \left[g^{-r}(t)\right]^2 dt \tag{4.5}$$

or

$$< P >= K_r \int_a^b \left[G^{-r}(f)\right]^2 df \tag{4.6}$$

Noting that $G^{-r}(f) = (i2\pi f)^{-r}G(f)$, the nominal frequency of the estimate is in fact given by the centroid of $f^{-2r}|G(f)|^2$. This means that,to some extent, the nominal frequency also depends on the order of the noise process. Deeter, in the appendix part of his paper (Deeter 1984) repeats the calculation, makes use of

4.2 and ends up with for discrete sampling $g(t) = \sum_j g_j \delta(t - t_j)$;

$$\langle \left(\sum_j g_j x_j \right)^2 \rangle = K_r \frac{(-1)^r}{2(2r-1)!} \sum_j \sum_k g_j g_k |t_j - t_k|^{2r-1}. \tag{4.7}$$

t_j's are the discrete times of sampling, need not be equally spaced which is the power of this low-resolution method.

4.2.1 Bandpasses of Sampling Functions

The selection of a sampling function for the recovery of red power spectra hinges largely on the features of its frequency bandwidth. The effective frequency response can be very sensitive to the noise process itself. Due to 4.6, the effective frequency response of g(t) is given by the transfer function of $g^{-r}(t)$ which can be obtained in a direct manner;

$$|G^{-r}(f)|^2 = (2\pi f)^{-2r} |G(f)|^2 \tag{4.8}$$

4.2.1.1 Simulation Results

Using equation 4.8 the bandpass responses of the sampling functions can be found. We have generated linear, quadratic and cubic polynomials each of which spans a unit time interval. In the Figure 4.1 the responses of these polynomials are shown. Horizontally to the right, the columns corresponds to the degree of the noise process going as r=1,r=2,r=3. Vertically to the bottom, the lines correspond to the polynomial order;linear,quadratic, cubic.

Since the rth integral of the sampling function has a primary concern when finding the frequency response, in the time domain, the three moments with

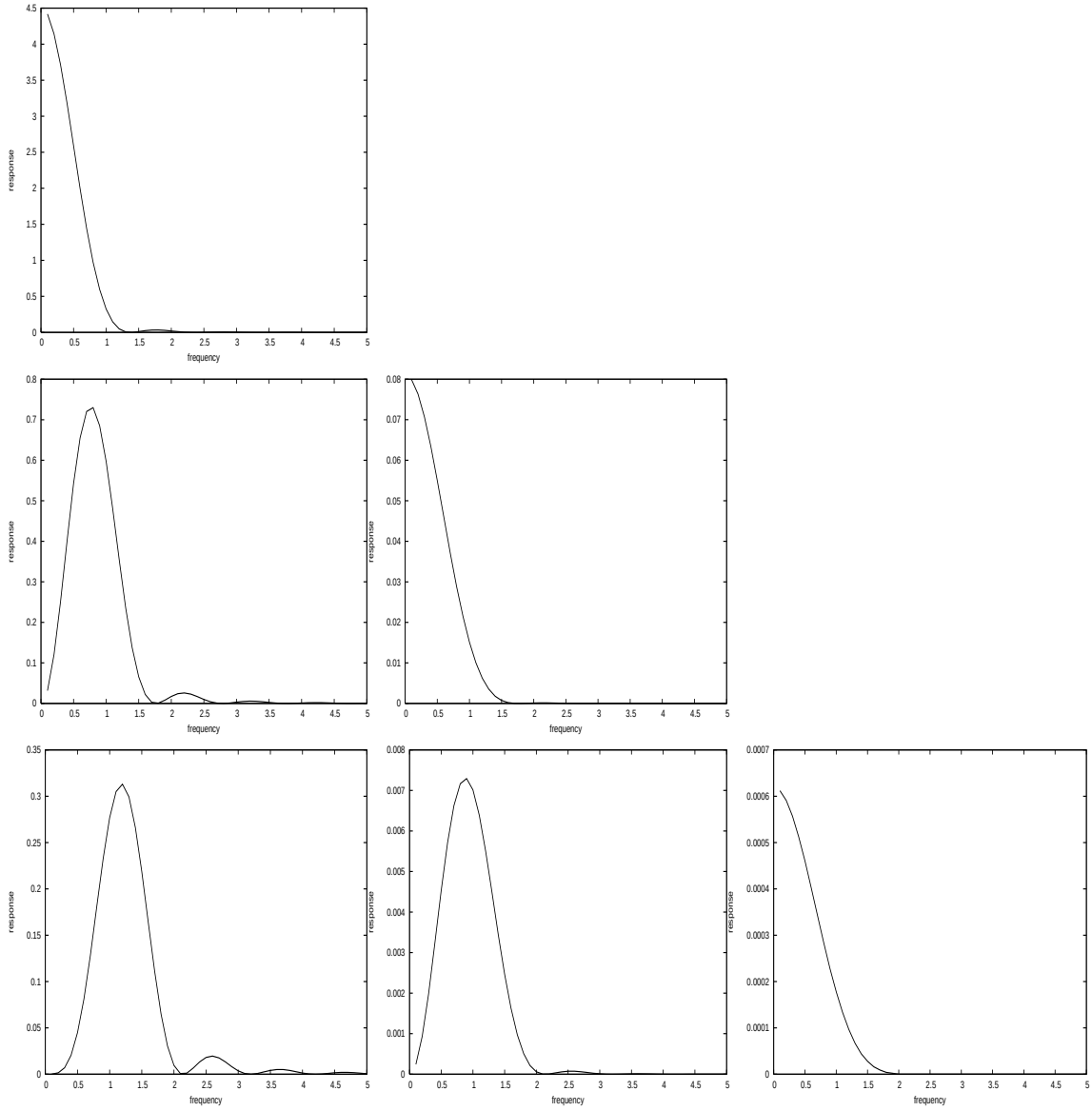

Figure 4.1: Bandpass responses of first three polynomials

respect to this rth integral of the sampling function can be found. The kth moment can be written as

$$M_k = (-1)^r \frac{k!}{(m+k)!} \sum_j g_j {t_j}^{r+k}.$$

Finally the half-width can be written as;

$$\Delta T = \left[M_2/M_0 - (M_1/M_0)^2 \right]^{1/2}$$

From this uncertainity in the time domain, one can switch back to the frequency domain by introducing a β factor, which is the product of ΔT and median frequency obtained from the frequency response of the rth integral of the sampling function;

$$\beta(r, m) = \Delta T f_{med}(r, m)$$

where m designates the order of the noise process. The calibration of this factor is given by Deeter (Deeter 1984). And in this study the calibration of Deeter is also checked and modified for our simulations. So if one can calculate ΔT from the data, median frequency can also be found from the calibrated β factor. This is what we have done in the calculations.

4.2.2 How to use the Method?

From a set of noise data, a single noise strength estimate can be obtained at the longest time scale (corresponding to the lowest frequency) by applying the estimator to the entire data set.The data set can then be divided into two equal parts, and a scaled version of the estimator applied to each part to obtain

a noise strength estimate at twice the lowest frequency ; in turn, the set can be divided into fourths, eights,and so forth. A single noise strength estimate can be obtained by averaging the individual estimates. Not each estimate for a given time scale is the same, so averaging of the median frequencies is also required. For the nonoverlapping estimators, the stability of each average estimate for each time scale is given by the width of the χ^2 distribution whose number of degrees of freedom equals the number of individual estimates (Deeter 1984 and also Appendix A). While displaying the results graphically, it is a choice to plot log-log graphs of noise strength estimate versus frequency. However, log distribution differs with that of our usual distribution. A kind of bias is introduced by Deeter et al. (1989):

$$\begin{aligned} B &= < logK > - log < K > \\ &\simeq -\left[\frac{1}{n} + \frac{1}{n^2}\right] log\ e \end{aligned}$$

and the variance is given as

$$var(logK) = \left[\frac{2}{n} + \frac{2}{n^2} + \frac{4}{3n^2}\right] (log\ e)^2$$

n here is the number of degrees of freedom. If the logarithms of the noise strength estimates are used in model fitting, this bias must be subtracted so that all the estimates of log P will have the same expectation for the assumed noise process.

Hence in this method, the order of the noise comes into the picture as an input. So it must be assumed and supplied by the user and the linearity of the final result must be negligible for the assumption to be correct (flat spectrum) since according to equation 4.7 K_r is constant.

4.3 Simulations

4.3.1 Model Test

The generated, equally spaced (unit time sampling) first order and second order noise realizations (with different seed numbers in the generation algorithm) are used to test the model.

4.3.1.1 r=1 type noise

We have generated a r=1 type noise process, whose corresponding white noise variance is 1 (i.e. the noise strength), with 100000 data points. Firstly the data set was divided into subsets for which cubic and quadratic polynomials, which were found to satisfy the desired moment conditions, were used as sampling functions. The average power of subsets showed that quadratic polynomial is a better sampler for recovering a r=1 type noise process with a closer average noise strength ($\simeq$ 1) and with an uncertainity calculated by dividing the standart deviation of the distributions by the square root of the number of subsets (Bevington 1992)(see Table 4.1).

The results also show that the noise strength can be recovered more precisely when the number of averaged values are increased (compare the 2nd and 3rd lines in the Table 4.1).

4.3.1.2 r=2 type noise

Similarly, for the second order red noise process check the Table 4.2.

Table 4.1: Low Resolution Technique applied to the first order red noise

num. of Subsets	Seed	Estimator	Noise Strength
1000	-2	cubic	0.94 ∓ 0.04
1000	-2	quadratic	0.98 ∓ 0.04
10000	-2	quadratic	1.00 ∓ 0.01
100	-5	cubic	1.09 ∓ 0.12
100	-5	quadratic	0.98 ∓ 0.11

In Figure 4.2 are there two histograms of the noise strengths of two different noise processes (r=1 and r=2 types). Quadratic and cubic polynomials were used respectively.

The figures show a characteristic of a distribution of a χ^2 with one degree of freedom. This is also justified in Figure 4.3 with scaling the corresponding χ^2 distribution (dashed lines) for the second histogram in the previous figure. This also proves experimentally that the stability of the estimates really depend on chi-square distribution. This is expected because the red noises are produced from a white noise which has standart normal distribution and then its modulus square must have a chi-square distribution which is related to the noise strength (Appendix A).

Similar simulation was also applied to the third order red noise process to correctly recover the power spectum with a noise strength of 1 (the variance of the basis white noise process). For that, cubic and quartic polynomials were used

Table 4.2: Low Resolution Technique applied to the second order red noise

num. of Subsets	Seed	Estimator	Noise Strength
1000	-2	cubic	1.01 ∓ 0.05
1000	-2	quadratic	0.95 ∓ 0.04
100	-5	cubic	1.02 ∓ 0.09
100	-5	quadratic	1.05 ∓ 0.08

as sampling functions. It is seen that quartic polynomial for the third order red noise process is a good choice to recover the spectrum. This simulation also shows the strength of the method over the windowing method which has a limitation when the order increases.

Table 4.3: Low Resolution Technique applied to the third order red noise

num. of Subsets	Estimator	Noise Strength
1000	cubic	0.95 ∓ 0.05
1000	quartic	0.99 ∓ 0.04

4.3.2 β factor calibration

The β factor values for a given noise process and the estimator are given in detail by Deeter (Deeter 1984). We have also checked the validity of his results and they turned out to be applicable. As in this study we have predominantly worked with a second order red noise process and ,as a consequence, worked with cubic polynomial estimators, the β factor for that case was crucial. Considering a cubic polynomial on the unit interval, we have found it to be 0.159 corresponding to a frequency of 0.921 (unit of frequency) whereas it is given by Deeter as 0.158 with a corresponding frequency 0.918 (unit of frequency) which is not so distinct from ours. This calculation is in fact easy but the point is that the polynomial must be constructed over an interval of type (-a,a), i.e. for the unit interval it should be (-0.5,0.5). The standard deviation from the mean frequency was found as 0.49 which is approximately half of the median frequency. That also gives an intuitive answer for why we use the median frequency in determining the β factor because it is nearly the uncertainity in the frequency domain. As an other simulation, we have tried to find the frequency response of the cubic polynomial, lasting for two unit time, to the second order red noise for two different sampling densities; one with 25 points and other with 50 points in that interval. Both of the simulations gave a median frequency around 0.5 frequency unit (which is expected for a 2 unit time interval) and a standart deviation half of that median frequency; consequently the same β factor as above. This shows the similarity property for different analysis frequencies and the stability of the β factor.

4.3.3 Window function responses

Deeter explained that any sinusoid can satisfy only those moment conditions forced by parity consideration; however this problem can be circumvented by polynomial trend removal . Unfortunately, simple trend removal does not produce sinusoid estimators with good bandpasses. Except for first order red noise, the responses are mostly bimodal and concentrated at very low frequencies rather than at nominal frequencies determined beforehand. So, for red noise processes sinusoids are excluded as sampling functions (Deeter 1984). In the previous chapter we deal with window functions while determining the order of noise. However, some problems arised when the order gets larger. In windowing method, the data is multiplied by the window and the fourier transform of the product is taken. This is equivalent to multiplying the data with the product of window and a sine function. So this product can be,in fact, considered to be a sampling function. In order for this function to satisfy the desired moment conditions polynomial trends up to desired order can be removed and then the result can be used as a sampling function. We have taken the Hann window with a unit time interval, multiplied it with a sine function with a unit time period, removed the cubic trend from the data and found the frequency responses of this sampling function for three different order noises (r=1,r=2,r=3). The results are given in Figure 4.4. The results show that the responses are mostly unimodal and highly centered at the frequencies larger than the nominal frequency. The presence of window function reduced the bimodal structure of pure sinusoid. Horizontally to

the right the order of the noise is increasing.

A similar procedure was applied to the product of Hann window and a sine function, without removing the polynomial trend, defined in a unit time interval (i.e. with a nominal frequency f=1). The results show that the frequency responses are unimodal but concentrated at very low frequencies, especially for the higher order noise processes. As a result, the study here is a good indication of the limitation of the use of window functions in the previous chapters.

4.3.4 Experimental Sensitivity

There is also a question that must be answered that why we use cubic polynomials in the case of the determination of the second order red noise or quartic polynomial for third order red noise. While using the method, the noise strength at a set of analysis frequencies with octave spacing are estimated. Systematic use of the same degree polynomial from an orthogonal set on every analysis time scale yields a set of local noise strength estimators that would have identical relative bandpasses if applied to equispaced data. This property of similarity is important for the recovery of power-law power spectra with large, negative exponents. The minimum polynomial degree is set by the power-law exponent and by the level of protection desired against contamination of the estimator. For instance, for a second order red noise in pulse phase, one would need to use at least the second-degree polynomial from the orthogonal sequence. However, the cubic polynomial would have to be used in order to provide protection against the possibility that the noise might be slightly redder than second order (one level of

protection). There are several reasons for employing the lowest possible degree consistent with adequate protection (Deeter et.al. 1989). First, higher degree polynomials can not be applied to the small number of data points. Depending our observations, contamination of quartic polynomial is more dominant (after 0.042 unit frequency, i.e. below 24 unit time) than that of cubic polynomial (after 0.13 unit frequency, i.e. below 8 unit time) while determining the second order red noise. This means that one can construct power density estimators, not contaminated from the estimator, up to larger analysis frequencies with the use of a lowest possible degree estimator. Also, responses of the estimators with one level of protection are more likely to be unimodal and centered approximately at the inverse of the sampling interval (i.e. at $f \simeq 1/T$).

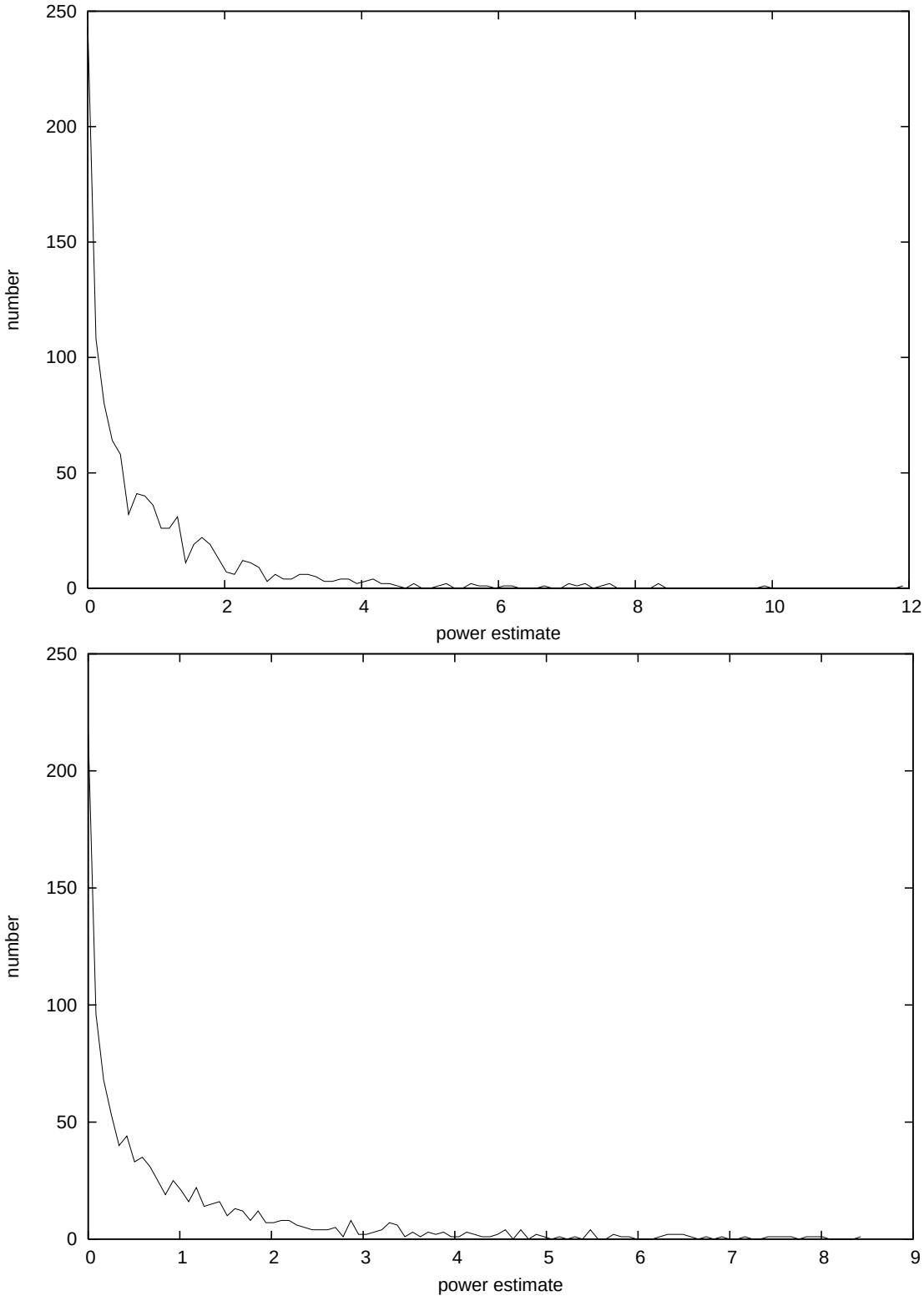

Figure 4.2: Histogram of the noise strength(power) estimates for r=1(upper panel), r=2(lower panel)

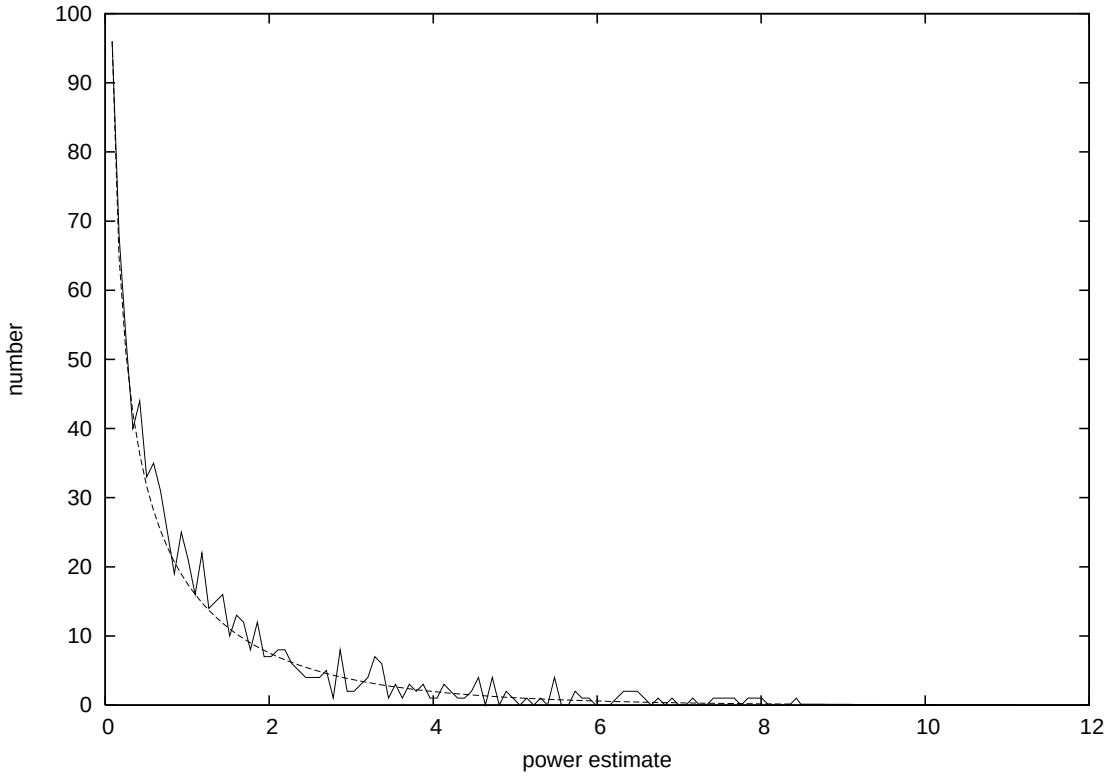

Figure 4.3: chi-square with one degree of freedom fit to the histogram of the noise strength(power) estimate for r=2

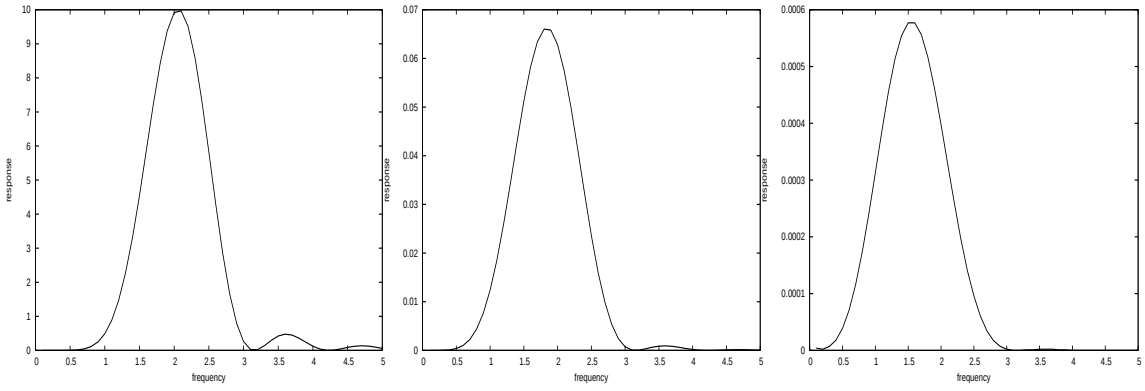

Figure 4.4: Frequency responses of the product of Hann and sine functions after cubic polynomial trend is removed

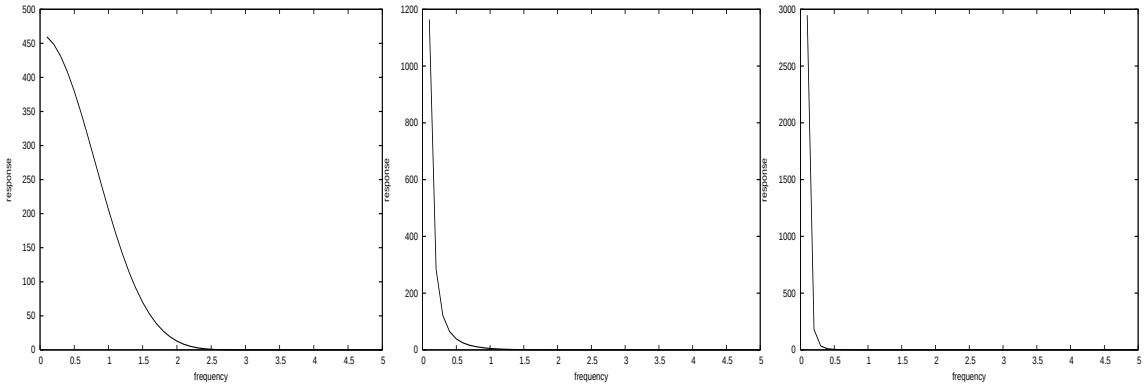

Figure 4.5: Frequency responses of the product of Hann and sine functions

CHAPTER 5

APPLICATION

5.1 Sax J2103.5+4545

The transient X-ray source SAX J2103.5+4545 was discovered by the Wide Field Camera on the BeppoSAX X-ray satellite during the outburst between 1997 February and September(Hulleman et.al. 1998). The source showed 358.61 s pulsations. The pulse arrival times provided by the RXTE proportional counter array (PCA) yielded an eccentric orbit (e=0.4$\mp$0.2) (Baykal et al. 2000). The X-ray spectrum was consistent with a power-law model. The photon index was 1.27 ∓ 0.14 and the absorption column density was $(3.1 \mp 1.4) \times 10^{22} cm^{-2}$. The orbital parameters suggest that the source has a high-mass companion. Another outburst was detected 2 years later by the all-sky monitor (ASM) on the Rossi X-Ray Timing Explorer (RXTE). SAX J2103.5+4545 continued to be active more than a year after the ASM detectors detected it in 1999 November. The 358 s pulsar spun up for 150 days, at which point the flux dropped quickly by a factor of 7, the frequency saturated, and, as the flux continued to decline, a weak spin-down began. The spin-up/flux correlation can be fitted to the Ghosh and Lamb derivations for the spin-up caused by accretion from a thin, pressure dominated disk (Ghosh et al. 1979). A distance of 3.2 ∓ 0.8 kpc and a magnetic field of

$(12 \mp 3) \times 10^{12}$G were obtained(Baykal et.al. 2002). The outburst could have been caused by an episode of increased wind from a Be star, such that a small accretion disk was formed during each periastron passage. The distance estimate implied that the X-ray luminosity observed was between 1×10^{36} and 6×10^{34} erg s^{-1}.

5.2 Application of the Low Resolution Technique

For the timing noise analysis of SAX J2103.5+4545, the phase noise data extending over a time interval of 380 days was used. The data set was divided octave by octave, cubic polynomial was used as a sampling function for each portion assuming the presence of the second order noise process, for each analysis frequency logarithmic averages of the noise strength estimates found were used and at the end the correction for the biasses was considered (as it was discribed in the previous section). For the lowest possible analysis frequency, obviously we have 1 degree of freedom, and for the larger frequencies the degrees of freedoms go on like 2,4,8,12,17. The weight of each estimate was calculated depending on the degree of freedom as mentioned before. After the fifth division measuremental error tends to dominate.

The log-log graph of noise strength versus analysis frequency shows that the assumption that there may be a second order red noise process turned out to be favourable since after linear fitting (taking into account the weights of all points), the linear coefficent is around 0.048457 with an error 0.271781. The noise strength turns out to be -19.319009 with an error 1.718067. So the linear trend is somewhat negligible and the application of the low resolution method to

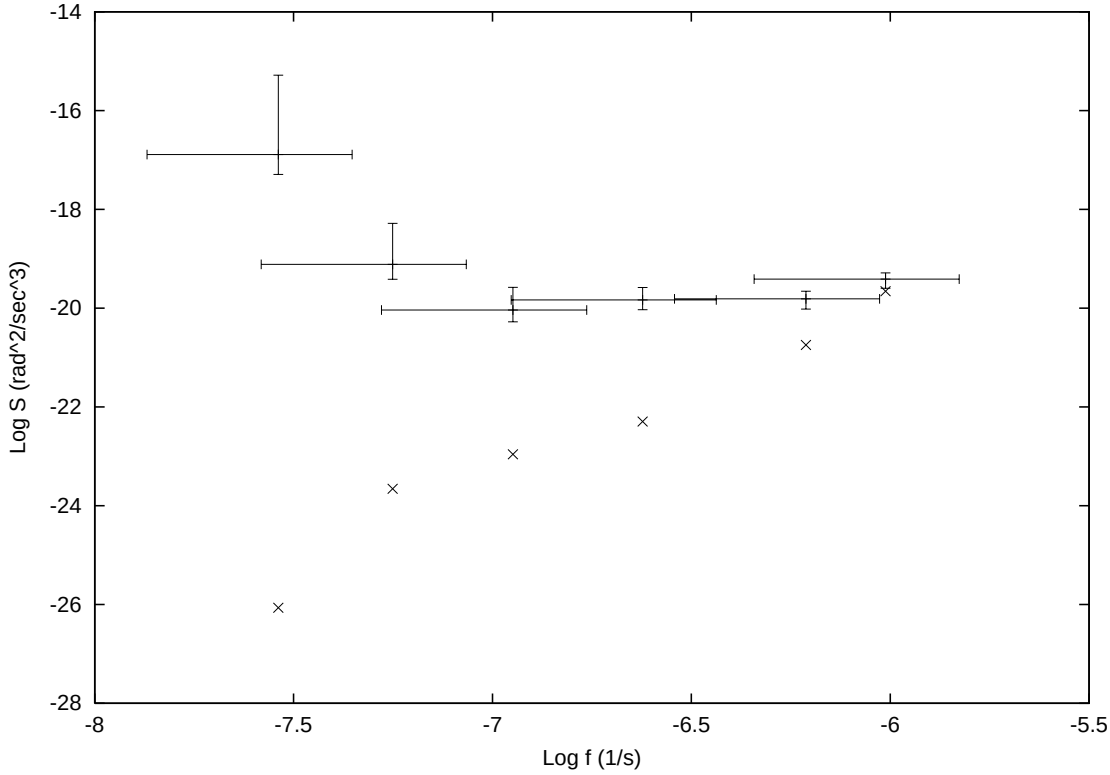

Figure 5.1: log-log graph of noise strength versus analysis frequency for SAX J2103.5+4545 using r=2, crosses show the measuremental error

this source revealed that the timing noise of SAX J2103.5+4545 can be modelled by a frequency noise (FN) or a random walk in frequency. This is also equivalent to a white torque noise.

Application of several windows to recover the expected power-law for this source (interpolating the data with a cubic spline routine in order to have a equispaced data, Press 1992) revealed that the power-law index, m, should be greater than 3.5 (m must be 4 for a second order red noise process). Since

Table 5.1: Noise Strength versus Analysis Frequency of SAX J2103.5+4545 using r=2

Log f(1/sec)	Log S(rad^2/sec^3)	Measuremental Error
(-7.868616 , -7.352865)	(-17.292905 , -15.282905)	-26.065305
(-7.581515 , -7.065764)	(-19.413962 , -18.283962)	-23.657153
(-7.278862 , -6.763111)	(-20.27789 , -19.57789)	-22.959295
(-6.952960 , -6.437210)	(-20.032835 , -19.582835)	-22.295984
(-6.542362 , -6.026611)	(-20.018142 , -19.658142)	-20.744804
(-6.342198 , -5.826447)	(-19.596841 , -19.286841)	-19.658635

they were just single estimates they can be simply from the far end tail of the distribution. In addition, we could not apply the windows to whole data due to the large gaps between the data points (preventing us from interpolating the data thoroughly); i.e. we used at most half of the data in this application (remember the sensitivity of this windowing method to the number of data points, dense sampling and length of the window, especially to the red noise processes with order greater than 2).

CHAPTER 6

DISCUSSION AND CONCLUSION

In the last part of this study the results and accuracies of the techniques that are used to determine the red noise processes and their application to the source SAX J2103.5+4545 will be discussed. The simplest way to determine a power spectrum is mostly done by using the well-known Fourier Techniques whereas it is the worst choice while determining the red noise due to the leakage problem discussed in Chapter 2. However using an appropriate window function can be a remedy to recover this problem. Simulations show that this is applicable;especially after removing a polynomial trend from the time series, for low order red noise realizations (compare Table 2.1 and Table 2.2). The strong dependence on the number of points, the need of removing a polynomial trend and the applicability up to at most second order red noise costitutes the defects of the windowing method. Being a more convenient and powerful method, low resolution technique is reviewed; the use of Legendre-like function which satisfies certain moment coditions (equation 4.3) and has a narrow bandpass peaked at the nominal frequency (corresponds to one-level of protection, Figure 4.1) is required as a sampling function. The low resolution technique turns out to be a mixed type, a bridge between the time domain and frequency domain. The tables

4.1, 4.2, and 4.3 indicate the accuracy of the method and Figure 4.3 reflects the stability of a single noise strength estimate that is turned out to be a χ^2 distribution. As the order of the red noise increases, the smooth functional behaviour is also apparent (Figures 3.1 and 3.2) and so is non stationarity (section 4.1) which must be removed in timing analysis. The non-stationarity can also be removed by the low resolution technique (equation 4.7),i.e. while determining the noise strength estimate only time differences between the data points are important, not the local times. Application of this technique to the transient source SAX J2103.5+4545 reveals the presence of a white torque noise behaviour with a noise strength value in between $(5.4 \times 10^{-21}, 1 \times 10^{-19}) rad^2 sec^{-3}$. We found that this source appears also on the linear trend in the Noise strength versus Luminosity graph (Figure 1.1) like the other accretion powered X-ray sources shown in Chapter 1. The low noise strength observed for this source may also be a signiture of its being a transient system.

REFERENCES

[1] Alpar, A. et al. 1986, ApJ, 311, 197

[2] Baykal, A., Ögelman H. 1993, 267, 119

[3] Baykal, A. 1997, A&A, 319, 515

[4] Baykal, A. et al. 2000, ApJ, 544, L129

[5] Baykal, A. et al. 2002, ApJ, 569, 903

[6] Baykal, A. et al. 2001, ApJ, 327, 1269

[7] 7) Bevington, P.R., Robinson, D.K. 1992 Data Reduction And Error Analysis For The Physical Sciences

[8] Bildsten, L. et al. 1997, ApJSS, 113, 367

[9] Boynton, P.E. et al. 1984, ApJ, 283, L53

[10] Corbet, R.H.D. 1986, MNRAS, 220, 1047

[11] Cordes, J.M. 1980, ApJ, 237, 216

[12] Cordes, J.M., Helfand, D.J. 1980, Apj, 239, 640

[13] Deeter, J.E. 1981, PhD Thesis, Univ. of Washington

[14] Deeter, J.E., Boynton, P.E. 1982, ApJ, 261, 337

[15] Deeter, J.E. 1984, ApJ, 281, 482

[16] Deeter, J.E., Boynton, P.E., Lamb, F.K., Zylstra, G. 1989, ApJ,336,376

[17] Freeman J.J. 1958, Principles of Noise

[18] Ghosh P., Lamb, F.K. 1979a, ApJ, 232, 259

[19] Ghosh P., Lamb, F.K. 1979b, ApJ, 234, 296

[20] Groth, E.J. 1975a, ApJSS, 29, 431

[21] Groth, E.J. 1975b, ApJSS, 29, 443

[22] Groth, E.J. 1975c, ApJSS, 29, 453

[23] Hulleman, F.et al. 1998, A&A, L25

[24] Illarionov, A.F., Sunyaev R.A. 1975, A&A, 39, 185

[25] Lamb, F.K. et al. 1978a, ApJ, 224, 969

[26] Lamb, F.K. et al. 1978b, ApJ, 225, 582

[27] Miller, I.,Miller M. 1999, John E. Freund's Mathematical Statistics

[28] Press, W.H. et al. 1992, Numerical Recipes in C

[29] Pringle, J.E., Rees, M.J. 1972, A&A, 21, 1

[30] Scott, D.M., Finger M.H., Wilson C.A. 2003, MNRAS, 334, 412

[31] van der Klis, M. 1988, in Timing Neutron Stars, NATO Advanced Study Institue, ed. H.Ögelman and E.P.J van der Heuvel (Kluwer Academic Publishres, Dordrecht), 28-42

APPENDIX A

χ^2 DISTRIBUTION

If X is a random variable, the function given by $f(x) = P(X = x)$ for each x within the range X is called the probability distribution of X which has the following properties: (a) $f(x) \geq 0$ for each value within its domain, (b) $\sum_x f(x) = 1$ where the summation extends over all the values within its domain. Using this function the mean, the variance and several other statistical properties of X can be found easily. So, the functional form of $f(x)$ should be known for any random variable X.In the continuous case, the summation turns out to be an integral. As an other definition, the function

$$F(x) = P(X \leq x) = \sum_{t \leq x} f(t)$$

is called the distribution function, or the cumulative distribution of X. From the definitions it follows that

$$F(-\infty) = 0, F(\infty) = 1;$$

$F(a) \leq F(b)$ for any real numbers a and b and if $a \leq b$

In the continuous case $P(a \leq X \leq b) = \int_a^b f(x)dx$ where f(x) is now the probability density function. And hence

$$\int_{-\infty}^{\infty} f(x)dx = 1$$

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(t)dt$$

From these definitions;

$$P(a \leq X \leq b) = F(b) - F(a)$$

$$f(x) = \frac{dF(x)}{dx} \text{where derivative exists}$$

A.0.1 Moment Generating Functions

The rth moment about the origin of a random variable X, denoted by $\mu_r^{'}$ is the expected value of X^r:

$$\mu_r^{'} = E(X^r) = \sum_x x^r \cdot f(x)$$

These moments can easily be found by introducing a moment generating function as:

$$M_X(t) = E(e^{tX}) = \sum_x e^{tX} \cdot f(x)$$

and expanding e^{tX}, inserting the expansion into summation and eventually finding the coefficients of $\frac{t^r}{r!}$. The coefficients turn out to be the desired moments of the distribution. It is also easy to see that the rth derivative of the moment generating function with respect to t at $t = 0$ is the rth moment about the origin (see Miller M., page 148).

A.0.2 Gamma Distribution

A random variable X has a gamma distribution and it is referred to as a gamma random variable if and only if its probability density is given by

$$g(x;\alpha,\beta) = \frac{1}{\beta^{\alpha}\Gamma(\alpha)} x^{\alpha-1} e^{-x/\beta}$$

for $x > 0$ and $\alpha > 0, \beta > 0$.

$$\Gamma(\alpha) = \int_0^{\infty} y^{\alpha-1} e^{-y} dy$$

for $\alpha > 0$, is the well-known gamma function.When $\alpha = \nu/2$ and $\beta = 2$ the distribution has a special name called chi-square distribution:

$$f(x) = \frac{1}{2^{\nu/2}\Gamma(\nu/2)} x^{\frac{\nu-2}{2}} e^{\frac{-x}{2}}$$

for x¿0. The parameter ν is referred to as the number of degrees of freedom. The mean and the variance of the chi-square distribution are given by;

$$\mu = \nu$$

$$\sigma^2 = 2\nu$$

The moment generating function can be found as:

$$M_X(t) = (1-2t)^{-\nu/2}$$

A.0.3 Normal Distribution

The distribution function

$$n(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

is called the normal distribution function where μ and σ are the mean and the variance respectively. If the former is 0 and the latter is 1 the distribution is called the standard normal distribution.

APPENDIX B

FINDING THE PROBABILITY DENSITY FUNCTIONS

By the help of three techniques; distribution function technique (DFT), transformation technique (TT) and the moment-generating function technique (MGFT) the probability density function of a random variable $Y = u(X_1, X_2, \cdots, X_n)$. can be found. As an example, by using DFT and TT, it can be shown that the probability density distribution of $Z = X^2$ is a chi-square distribution with 1 degree of freedom if X has the standard normal distribution. (see Miller, page 248)

B.1 Moment Generating Function Technique

This technique is powerful when we deal with the linear combinations of independent random variables (i.e. the multivariate distribution can be separated $f(x, y, z, ...) = f(x).f(y).f(z)...$). By the help of this technique, it can be shown that the moment generating function of $Y = X_1 + X_2 + X_3 + \cdots + X_n$ where X_i are all independent can be given as:

$$M_Y(t) = \prod_{i=1}^{i=n} M_{X_i}(t)$$

where $M_{X_i}(t)$ is the moment-generating function of $X_i(t)$(see Miller, page 262). So, if all the X_i's have chi-square distributions, the distribution of Y is again a

chi-square distribution with a number of degree of freedom which is equal to the addition of individual degrees of freedom. At this point we should remember the moment-generating function of chi-square distribution and notice that the multiplication of several of this brings an exponential which is equal to the addition of the all ν's.

$$M_X(t) = (1 - 2t)^{-\nu/2}$$

APPENDIX C

C ROUTINES

C.1 Finding The Average Noise Strength of Uncorrelated Sets Of Distinct Noise Processes With Equal Length And Order

```
#include <stdio.h>
#include <math.h>
#define N //number of the points that will be read from the whole data
#define B //number of subsets
#define M //must be greater than D
#define D // D-1 is the degree of polynomial that will be used as a
              sampling function
#define R //order of the noise


int main(void) {
int l,k,i,n,j,s,r,jj;
double m,q,sum,sum1,norm;
double g[N+1][M+1];
double w[N+1];
double t[N+1];
```

```
double x[N+1];

double diff,c,power,ortapower,ave,max,min,ortalogpower;

double p[B+1];

FILE* inp;


inp=fopen("noisedata.txt","r");


// read the data

for(k=1;k<=N;k++) {

               fscanf(inp,"%d%lf",&l,&m);

                t[k]=(double)(l);

                x[k]=m;

                  }

s=N/B;

for(j=1;j<=B;j++) {

     q=0.0; sum=0.0;

              for(i=1;i<=s;i++) {

                  sum=sum+t[(j-1)*s+i];

                  q++;

                                  }

      ave=sum/q;

              for(i=1;i<=s;i++) {

                   t[(j-1)*s+i]=t[(j-1)*s+i]-ave;
```

```
                                }
// construct orthonormal polynomials
            n=M-1;
                for(i=1;i<=s;i++) {
                     w[(j-1)*s+i]=1.0;
                   }
                for(r=1;r<=n+1;r++){

                       if(r==1) {
                           for(i=1;i<=s;i++) g[(j-1)*s+i][1]=1.0;
                                 }
             else     {

            for(i=1;i<=s;i++) g[(j-1)*s+i][r]=pow(t[(j-1)*s+i],(r-1));

            for(jj=1;jj<=r-1;jj++) {
                  sum=0.0;
                  for(i=1;i<=s;i++)
                        g[(j-1)*s+i][r]=g[(j-1)*s+i][r]-sum*g[(j-1)*s+i][jj];
                  }
               }
            sum=0.0;
            for(i=1;i<=s;i++)
```

```
                sum=sum+w[(j-1)*s+i]*g[(j-1)*s+i][r]*g[(j-1)*s+i][r];
          sum=sqrt(sum);
          for(i=1;i<=s;i++) g[(j-1)*s+i][r]=g[(j-1)*s+i][r]/sum;
                                        }

// estimate the noise strength

        sum=0.0;
        for(i=1;i<=s;i++) {
                  sum=sum+x[(j-1)*s+i]*g[(j-1)*s+i][D];
                           }
                  sum=sum*sum;
                        sum1=1.0;
                        for(k=1;k<=2*R-1;k++) sum1=sum1*k;
                        q=(double)(pow(-1,R))/(2.0*sum1);
                        c=0.0;
       for(k=1;k<=s;k++) {
                 for(i=1;i<=s;i++) {
                     diff=fabs(t[(j-1)*s+k]-t[(j-1)*s+i]);
 c=c+g[(j-1)*s+i][D]*g[(j-1)*s+k][D]*(pow(diff,2*R-1));
                                        }
                          }
```

```
        norm=q*c;


        power=sum/norm;
p[j]=power;
        printf("%f\n",power);
        }
.
.   // for each subset noise strength is found. You can take the average
.   // or the logarithmic average of those, check for the bias,plot the
.   // histogram of your results,....
.
.
return(0);
}
```

C.2 Plotting Histogram

```
#include <stdio.h>
#include <math.h>
#define N // number of points in the data
#define BLM // number of bins


int main(void) {
```

```
int i,t;

double y[BLM+1];

double p[N+1];

double m,max,min,fark;

int out[BLM+1];


FILE *inp;



inp=fopen("gts.out","r");


// read the data

for(i=1;i<=N;i++) {

                   fscanf(inp,"%d%lf\n",&t,&m);

                            p[i]=m;

                  }

min=p[1];

for(i=2;i<=N;i++) {

       if(p[i]<min) min=p[i];

                  }

// find the minimum value and subtract it from the whole values

                for(i=1;i<=N;i++) {

                        p[i]=p[i]-min;
```

```
                                    }
// find the maximum of the values after the above subtraction
                         max=p[1];
                  for(i=2;i<=N;i++) {
                         if(p[i]>max) max=p[i];
                                     }
// reset the counter
for(i=0;i<=BLM;i++){
                  out[i]=0;
                    }



for(i=1;i<=N;i++) {
               out[(int)((double)(BLM)*(p[i]/max))]++;
                   }
// construct the bins
                  for(i=0;i<=BLM;i++) {
                         y[i]=min+(double)(i)*(max/(double)(BLM));
                     }


for(i=0;i<=BLM;i++){
             printf("%f%c%c%d\n",y[i],' ',' ',out[i]);
                    }
```

```
fclose(inp);

return(0);

}
```

C.3 Finding Noise Strength By Using Orthonormal Polynomials With Implementing The Required Division For Each Analysis Frequency

```
#include <stdio.h>

#include <math.h>

#define N  \\number of data points

#define Mo \\ choose larger than D

#define D  \\ D-1 is the polynomaial that is used for sampling

#define R  \\ order of the noise process

int main(void) {

int l,k,i,n,j,s,r,jj,a,z,m2,u;

double m,q,sum,sum1,sum2,norm,m1,l2,T;

double f[ ];

double b[ ];

double f1[ ];

double sigma[ ];

double p1[ ];

double M[ ];

double g[ ][ ];

double w[ ];
```

```
double t[ ];

double t2[ ];

double x[ ];

double diff,c,power,ave;

double p[ ];

FILE* inp;

FILE* inp1;

inp=fopen("data.txt","r");

inp1=fopen("divide.txt","r");

for(k=1;k<=N;k++) {

                fscanf(inp,"%lf%lf%lf\n",&l2,&m,&m1);

                t2[k]=l2;

                t2[k]=(double)(j);

                x[k]=m;

                 }


// we have read the data


for(j=1;j<="number of lines in divide.txt";j++) {

              fscanf(inp1,"%d%d\n",&l,&a);


// find the mid time for each set and subtract it from all the times in

//that set
```

```
 q=0.0; sum=0.0;
         for(i=a;i<=l;i++) {
          q=q+1.0;
          t[i]=t2[i];
          sum=sum+t[i];
              }
            ave=sum/q;

           for(i=a;i<=l;i++) {
                        t[i]=t[i]-ave;
                             }

// for each set construct orthonormal polynomials

                        n=Mo-1;
               for(i=a;i<=l;i++) {
                        w[i]=1.0;
                  }
               for(r=1;r<=n+1;r++){

                     if(r==1) {
                         for(i=a;i<=l;i++) g[i][1]=1.0;
```

```
                                }
            else     {

           for(i=a;i<=l;i++) g[i][r]=pow(t[i],(r-1));

           for(jj=1;jj<=r-1;jj++) {
                    sum=0.0;
           for(i=a;i<=l;i++) sum=sum+w[i]*g[i][r]*g[i][jj];
                 for(i=a;i<=l;i++) g[i][r]=g[i][r]-sum*g[i][jj];
                 }
              }
sum=0.0;
           for(i=a;i<=l;i++) sum=sum+w[i]*g[i][r]*g[i][r];
           sum=sqrt(sum);
           for(i=a;i<=l;i++) g[i][r]=g[i][r]/sum;
                                    }
// choose the polynomial with order D-1 by inserting D
// find the noise strength

                    sum=0.0;
                    for(i=a;i<=l;i++) {
                             sum=sum+x[i]*g[i][D];
                                     }
```

```
                        sum=sum*sum;


            // normalization


                    sum1=1.0;

                    for(k=1;k<=2*R-1;k++) sum1=sum1*k;


                q=(double)(pow(-1,R))/(2.0*sum1);


                c=0.0;
        for(k=a;k<=l;k++) {
                for(i=a;i<=l;i++) {
                    diff=fabs(t[k]-t[i]);
                    c=c+g[i][D]*g[k][D]*(pow(diff,2*R-1));
                                }
                    }


        norm=q*c;
        power=sum/norm;
        p[j]=power;


// using second moments find f(med)!
         m2=D-1;
```

```
        for(k=0;k<=2;k++) {
                        sum1=1.0; sum=1.0;
                        for(i=1;i<=k;i++) {
                              sum1=sum1*i;
                                        }

                         for(i=1;i<=(k+m2);i++) {
                              sum=sum*i;
                                        }
                         c=(double)(pow(-1,m2))*sum1/sum;

                         sum2=0.0;
                         for(i=a;i<=l;i++) {

                             sum2=sum2+g[i][D]*pow(t[i],m2+k);
                                       }
                   M[k]=c*sum2;
                                    }

          T=(M[2]/M[0])-((M[1]*M[1])/(M[0]*M[0]));
          T=sqrt(T);
f[j]=0.1584/T;   // from the beta factor calibration(for cubic polynomial
//and second order noise process)
```

```
sigma[j]=(f[j]/0.92)*0.49;    // bandpass for the above case due to

//simulations

}

.

.

.

.

.

.

return(0);

}
```