Joint Analysis of X-ray Spectral and Timing using the State-Space Model: Application for the Black Hole Binary MAXI J1820+070

Tomoki OMAMA

Doctor of Philosophy

Department of Space and Astronautical Science
School of Physical Sciences

The Graduate University for Advanced Studies, SOKENDAI

March 2024

Acknowledgement

To complete this thesis, I received support from numerous individuals.

My supervisor, Prof. Masahiro Tsujimoto, provided valuable advice on applying techniques to astronomical data and guided the research direction from an astrophysics perspective. He dedicated a significant amount of time to proofreading this doctoral thesis. Not only did he respect my research interests and allow me to pursue them, but he also suggested research themes essential for achieving my final goal. He supported me in developing the necessary skills to accomplish it. Thanks to his assistance, the quality of this thesis has been greatly elevated. I express my utmost gratitude for all his contributions.

Prof. Ken Ebisawa warmly welcomed me, who transferred from the doctoral program, and provided an excellent research environment. He explained to me the history and challenges of black hole binary research, clarifying the goals the development method should aim for.

The lab members supported me in various aspects, from research to personal matters. I greatly appreciate Dr. Takuya Midooka, who continuously assisted me with X-ray data analysis, X-ray astronomy, and career-related matters. Mayu Tominaga provided advice on NICER data processing and astrophysical analysis methods, which is certainly helpful for my work. Miki Kurihara offered me a lot of opportunities to discuss Bayesian statistics and philosophical matters, contributing to organizing my thoughts. I recall engaging in discussions with Yuto Mochizuki about research and personal matters, where enjoyable and sometimes rigorous debates are invoked.

The research lab at Hiroshima University, where I spent my undergraduate and master's studies, taught me everything necessary to become a researcher from scratch. Meeting with former lab colleagues regularly, even after moving to the Institute of Space and Astronautical Science, provided a refreshing way to stay updated on each other's progress. Prof. Makoto Uemura, my previous supervisor, offered valuable advice on astrophysics and data science during both my master's and doctoral programs, playing a significant role in enabling me to conduct this research.

Apart from my enrolled lab, I engaged in discussions with many researchers and gained inspiration. Prof. Eric Feigelson and Prof. Hungsuk Tak provided valuable advice from a statistical perspective. Prof. Chris Done offered opinions from a theoretical standpoint. Prof. Mariko Kimura expressed interest in this research method and provided beneficial comments. I discussed the illustration of MAXI J1820+070 results with Prof. Megumi Shidatsu and received questions, opinions, and discussions on research directions from Dr. Tenyo Kawamura. I am deeply thankful to all the researchers involved.

I thank the examiners of this doctoral thesis, Prof. Makoto Uemura, Prof. Tadayasu Dotani, Prof. Ryuichi Fujimoto, Prof. Yukio Yamamoto, Prof. Ryo Osawa, and Prof. Masahiro Tsujimoto. Their comments and suggestions have improved my thesis.

Not only researchers but also many individuals associated with the Institute of Space and Astronautical Science (as known as ISAS) and the Graduate University for Advanced Studies supported everything involving me. I am particularly grateful to Secretary Noriko Takahashi, who supported everything necessary for the research. Thanks to her support, I could focus on research without being distracted by other matters.

Finally, I would like to express my gratitude to my family, who has supported me. When I decided to change my career path in the middle of high school and expressed my desire to study at university, they supported and cheered me with abundant love. They continued to support me with kindness during my university studies and research. Certainly, their love and support were crucial for my academic journey.

Tomoki Omama
March 2024

Software We use data observed by NICER (Gendreau et al., 2012) and processed it with HEAsoft (Nasa High Energy Astrophysics Science Archive Research Center (Heasarc), 2014) and Xspec (Arnaud, 1996).

For numerical calculations, we employed NumPy (Harris et al., 2020), SciPy (Virtanen et al., 2020), Statsmodels (Seabold and Perktold, 2010), TensorFlow (Abadi et al., 2016), and TensorFlow Probability (Dillon et al., 2017). Matplotlib (Hunter, 2007) and Seaborn (Waskom, 2021) are used for creating plots.

Grants-in-Aid for Scientific Research This work was supported by JSPS KAKENHI Grant Number 22J13440.

Abstract

A black hole binary (BHB) is a binary system consisting of a black hole and a non-degenerate star. The black holes came into existence with Cygnus X-1 in the 1970s, when two pioneers of X-ray astronomy, Minoru Oda and Ricardo Giacconi, boldly speculated that the rapid X-ray flux variability distinctively seen from this source could only be attributable to a black hole. Since then, many features of BHBs have been identified, such as spectral changes and transient behaviors, but the rapid variability from milliseconds to seconds remains one of the most distinctive features of BHBs. The origin of the rapid variability is still unknown, but it is believed to carry information about the accretion and ejection processes and the strong gravitational field in the vicinity of a black hole.

X-ray observations of BHBs record the energy and arrival time of individual X-ray photons from BHBs. Spectral and light curve analyses have been performed, but they were done independently from each other. On the one hand, traditional spectral analysis was performed for time-sliced spectra ignoring the correlation among them in time. On the other hand, traditional light curve analysis was performed without considering the changing contributions of spectral components in time. We need to develop a method for the joint spectral and timing analysis.

The statistical modeling approach provides the answer. If we denote the observed count in a time bin $n \in \{1, \cdots, N\}$ and an energy bin $m \in \{1, \cdots, M\}$ as c_{nm}, we regard c_{nm} as a realization of the probability variable C_{nm}. The goal is then to estimate the joint probability distribution of $p(C_{11}, \cdots, C_{NM})$. In this manner, timing and spectral information can be modeled jointly. This approach also has advantages including noise as a model component and using latent variables to describe the changes in the system behind observed values.

The statistical modeling approach sounds straightforward and suitable for the analysis of BHB data but has been scarcely used. Several reasons hampered the application to real data, including data quality, modeling techniques, computational resources, and physical models to interpret joint probability. However, recent advances in all of them are clearing these obstacles. It is high time to start using statistical modeling as the norm of X-ray spectral and timing analysis. In this thesis, we demonstrate that this is possible and even crucial in deriving new insights from BHBs by applying the method to the actual data of a BHB.

We use the data of MAXI J1820+070 observed with The Neutron star Interior Composition ExploreR (NICER). MAXI J1820+070 is a transient BHB discovered in 2018. The source exhibited many spectral and timing features common among BHBs in both the hard and soft states. The low interstellar extinction and the proximity made the

source extremely bright in flux reaching ∼4 Crab. NICER is the X-ray observing instrument onboard the International Space Station. The unprecedented collecting area and a large dynamic range of NICER and the brightness of the source resulted in an extreme count rate of $\mathcal{O}(10^4\ \mathrm{s}^{-1})$, which is rich enough to apply statistical modeling. We focus on a 50 s length of data during the hard state near the flux peak of the BHB.

We applied classical time series modeling to the X-ray light curves constructed at 0.5–2.0, 2.0–5.0, and 5.0–10 keV. We first used the autoregressive (AR) model for each light curve, and a reasonable fit was obtained. Because noise is included in the AR model, the univariate description functions (e.g., correlation function and power spectrum) are less noisy than those made by the traditional analysis using the raw data. Next, we used the vector autoregression (VAR) model. Because the mixture among the multi-band light curves is included, the fitting improved from the AR model. This implies the importance of the spectral mixture for the observed light curves.

We therefore proceeded with the linear Gaussian state-space modeling to the multi-band X-ray light curves in five energy bands. The observed light curves are treated as observation variables, whereas the intensity changes of the physical spectral components (Comptionized, disk blackbody, and soft excess components) were treated as latent variables. The system equation was described by the VAR model and the observation equation was described by a linear matrix. In this manner, we included both the spectral mixture and the correlation in time in a single model. As a result, we could derive the multivariate description functions (e.g., cross-covariance, cross spectra, and coherence) among the spectral components, not among the multi-band light curves. This is the advantage of using the latent variable in a model.

We produced the spectrally-decomposed power spectra and derived the break frequencies of the Comptionized, disk blackbody, and soft excess components. We also produced the spectrally-decomposed cross spectra to derive the time lags among them. From these results, we found that the three components affect each other in the causality order of the disk blackbody, Comptonized, and soft excess emission. The different break frequencies in the three components, the time lag between these components, and the mutual power contribution all point to the geometry of the truncated accretion disk.

This work is one of the first successful applications of the state-space modeling approach to BHB data analysis. We demonstrated the possibility and utility of the joint spectral and timing analysis by applying it to the actual data and obtaining new insights into BHBs. We consider that this should be one of the standard approaches to analyzing the data to come in the near future with advanced observing technologies.

Contents

Chapter 1

Introduction

Contents

1.1 Rapid X-ray Variability of Black Hole Binaries

Cygnus X-1 (Cyg X-1) was the first to be recognized as a black hole binary (BHB) in 1970's. The Uhuru satellite observed the source in the X-rays and discovered rapid flux variability in the X-ray light curves from milliseconds to seconds. Based on this, Minoru Oda and Riccardo Giacconi, the two pioneers of X-ray astronomy, speculated that the source is a black hole (Oda, 1974). Figure 1.1 shows the X-ray light curve of Cyg X-1 taken with the ANS satellite, in which rapid variability is seen despite the paucity of counts.

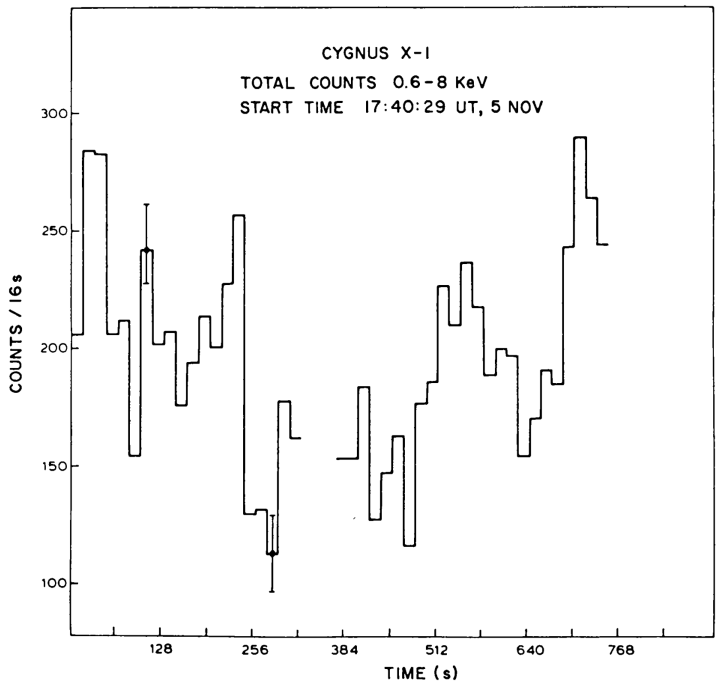


Figure 1.1: X-ray light curve of Cyg X-1 taken with the ANS satellite. Figure taken from Oda (1977)

Since then, the rapid variability in the X-ray light curve has become one of the defining characteristics of BHBs (Remillard and McClintock, 2006). The data quality has improved drastically over the past 50 years. Figure 1.2 is the X-ray light curve of MAXI J1820 + 070, which is a transient BHB discovered in 2018. This is the target of this study, and we investigate the correlation of the variability among light curves in different energy bands. At first glance, the rapid variability is strongly correlated among all light curves. However, if we look closer, we recognize that some features (e.g., low-frequency hump at 46 s) are seen preferentially in the light curve of the softer energy bands.

To better understand these differences, we turn to the energy spectrum, as shown in Figure 1.3. The energy spectrum reveals that one component dominates the entire spectrum, while several more subdominant components contribute to the softer bands. These spectral components have different variability, which is the reason why we observe different variability in different energy bands.

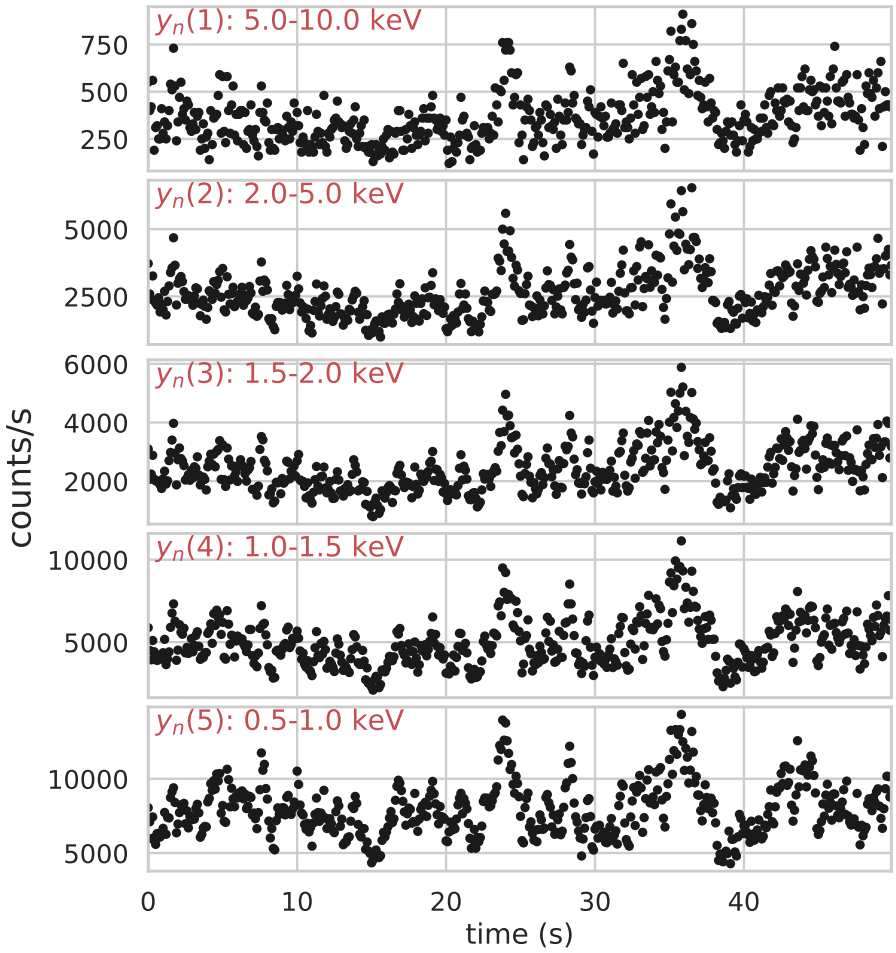


Figure 1.2: multiband light curve of MAXI J1820 + 070.

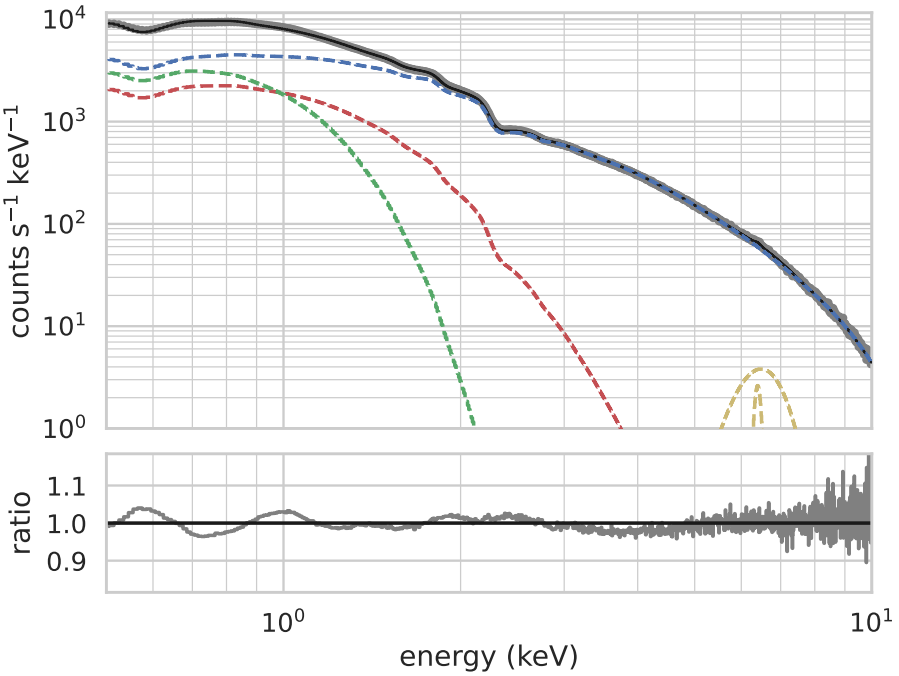


Figure 1.3: Energy spectrum of MAXI J1820 + 070.

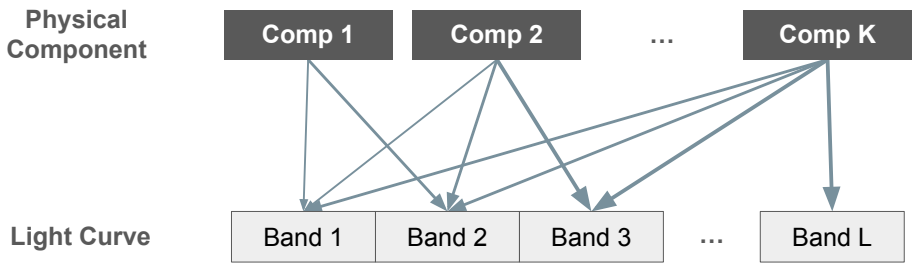


Figure 1.4: Schematics of the relation between physical components and energy bands.

On the one hand, light curve analysis has conventionally been performed for the observed light curves in multiple energy bands. However, what we really want to know is the variability of the spectral components, such as those shown in Figure 1.3. The observed light curves are just a mixture of them. Figure 1.4 schematically depicts the relationship between the spectral components and the multiband light curves. The goal of this study is to demonstrate how we can construct the light curves of the spectral components based on the statistical modeling approach. We treat the spectral components as latent variables and model the observed multiband light curves as a superposition of these latent variables.

On the other hand, the spectral analysis has traditionally been performed without considering the correlation in time. The spectra were constructed either by using the entire observation duration or a piece of it without introducing the correlation between them in time. In reality, the spectral components are correlated in time.

The goal of this thesis is to fill these gaps using statistical modeling.

1.2 Light Curve Analysis vs Time Series Modeling

Throughout this thesis, we use the following distinction.

Light curve analysis to derive features of the observed light curve without statistical modeling.

Time series modeling to estimate the probability distributions of the variability based on statistical modeling.

Classical time series modeling for statistical modeling without latent variables.

State-space time series modeling for statistical modeling with latent variables.

Traditionally, light curve analysis has been the norm for studying the X-ray variability of BHBs since the pioneering work by Oda (Oda et al., 1971). In this thesis, we demonstrate the need for the statistical modeling approach for further investigation.

Below, we clarify the differences between the light curve analysis and the time series modeling, and give an example to show why the statistical modeling approach is necessary. Through X-ray observations of BHBs, we obtain the arrival time and energy of individual X-ray photons, which we refer to as raw data hereafter. For demonstration, we synthesized 10,000 raw data of time and energy in panel (a) of Figure 1.5. These synthetic data represent 10,000 realizations from a constant distribution in time and a power-law distribution of an index of -2 in energy.

1.2.1 Light Curve Analysis

The traditional light curve analysis starts with constructing the light curve. We bin the raw data in the grid over time and energy. In panel (b) of Figure 1.5, the synthetic raw data are binned with the grid outlined with red lines in panel (a), and their counts per bin are shown with a color code. A plot depicting the time variation of the power spectrum is commonly called a spectrogram. Here, we refer to panel (b) as "energy spectrogram" by analogy. By labeling the binned time as $n = \{1, \cdots, N\}$ and the binned energy as $m = \{1, \cdots, M\}$, we denote the observed counts per bin as $c_{11}, \cdots, c_{NM}$. Integrating $c_{11}, \cdots, c_{NM}$ over energy produces the light curve (panel (d) in Figure 1.5)

$$x_n = \sum_{m=1}^{M} c_{nm}, \tag{1.1}$$

and integrating over time yields the time-averaged spectrum (panel (c) in Figure 1.5)

$$y_m = \frac{1}{N} \sum_{n=1}^{N} c_{nm}. \tag{1.2}$$

The light curve analysis consists of extracting the features of x_n by numerically manipulating them. It can be done in either the time domain or frequency domain after

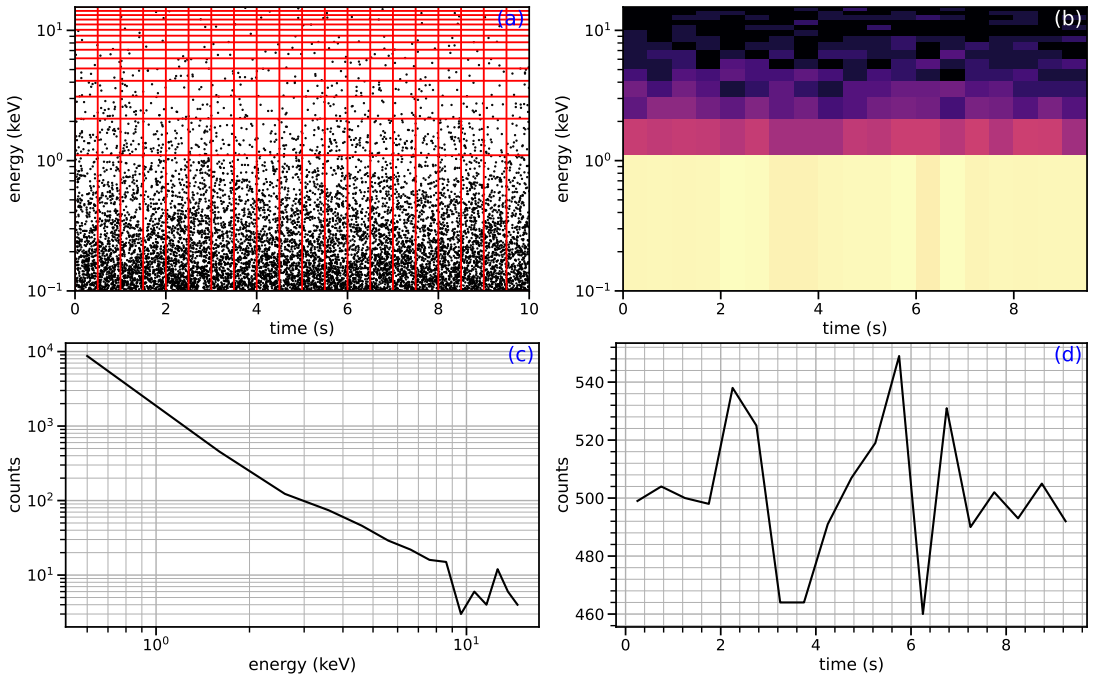


Figure 1.5: Synthetic X-ray photon data and approximated representations. (a) the synthesized X-ray photons. (b) The two-dimensional histogram is divided by the red lines in (a). (c) energy spectrum by summing over time. (d) light curve by summing over energy.

Fourier conversion. In the time domain, techniques such as autocorrelation functions or shot analysis are used. In the frequency domain, techniques such as power spectra are used. This is good enough to derive basic characteristic features such as the mean and standard deviation of the count rates or the coherence signals of a pulsar. Some results of BHBs obtained through such techniques are summarized in § 4.

This approach is prone to noise. Looking at the light curve (panel (d) in Figure 1.5), it is tempting to suggest that the light curve exhibits variability with a characteristic time scale of $\mathcal{O}(1\text{ s})$. This is not true, as the data are synthesized from the non-variable model. In addition, there is no way to introduce latent variables, which is essential to access changes in the spectral components behind the variation of the light curve.

In the realm of the statistical modeling, the light curve analysis corresponds to what is known as exploratory data analysis (EDA). This is typically a step performed before starting the statistical modeling. Features identified with EDA are incorporated into the subsequent statistical model.

1.2.2 Time Series Modeling

In the statistical modeling approach, we view $c_{11}, \cdots, c_{NM}$ as realizations of random variables following particular probability distributions $C_{11}, \cdots, C_{NM}$. The essence of statistical modeling is to estimate the joint probability distribution $p(C_{11}, \cdots, C_{NM})$ from the raw data $c_{11}, \cdots, c_{NM}$ that are realizations of the distribution. In this manner,

we can include noise into the model and separate observation and latent variables. We can exploit the established methods to do so in the statistical modeling approach.

If we model the entire data directly in panel (a) of Figure 1.5, we need to model the joint probability distribution of $10,000$ dimensions, which is unrealistic. By integrating the energy spectrogram C_{nm} over energy, we obtain the random variable as

$$X_n = \sum_{m=1}^{M} C_{nm}. \tag{1.3}$$

This defines the set $\{X_1, \cdots, X_M\}$ corresponding to the light curve. The process of estimating the probability distribution $p(X_1, \cdots, X_M)$ and modeling the light curve is known as time series modeling.

By integrating the energy spectrogram C_{nm} over time, we obtain the random variable

$$Y_m = \frac{1}{N} \sum_{n=1}^{N} C_{nm} \tag{1.4}$$

This defines the set $\{Y_1, \cdots, Y_M\}$ corresponding to the time-averaged energy spectrum. The process of estimating the probability distribution $p(Y_1, \cdots, Y_M)$ and modeling the energy spectrum would be called energy spectral modeling.

1.3 Joint Spectral and Timing Modeling

What we aim is to model C_{nm}, not X_n in Equation (1.3) nor Y_m in Equation (1.4). We have the observed light curve in multiple energy bands (Figure 1.2) in our hands. The joint probability distribution for the multiband light curves is equivalent to the one for the energy spectrograms as shown below. If we represent the multiband light curve at time n as a vector random variable X_n, its joint probability distribution is expressed as

$$p(X_1, \cdots, X_N) \tag{1.5}$$

On the other hand, the joint probability distribution of the energy spectrogram C_{nm} can be expressed using the random variable vector $X_n = [C_{n1} \cdots C_{nM}]^T$ as

$$p \begin{pmatrix} C_{11} & C_{21} & \cdots & C_{N1} \\ C_{21} & C_{22} & \cdots & C_{N2} \\ \vdots & \ddots & \ddots & \vdots \\ C_{M1} & C_{M2} & \cdots & C_{NM} \end{pmatrix} = p(X_1, \cdots, X_N). \tag{1.6}$$

It is evident that they share the same structure. We will focus on estimating the joint probability distribution for the multiband light curves in this thesis, which we term as the "joint spectrum and timing modeling".

This approach improves the modeling of not only the light curve but also of the energy spectrum. In the traditional energy spectrum analysis, the energy spectra are constructed in each time slice as y_m in Equation (1.2) and fitted independently from each other. From the statistical modeling point of view, this implicitly assumes conditional independence. To simplify conditional independence, consider three variables a, b, c. If a and b are conditionally independent given c, their joint probability distribution satisfies

$$p(a, b|c) = p(a|c)p(b|c).$$

For the counts in the energy spectrum bin at a time n, $X_n = [C_{n1}, \cdots C_{nM}]^T$, with the energy spectrum model $\mathcal{M}_E$, the conditional independence assumes

$$p(X_{n-1}, X_n|\mathcal{M}_E) = p(X_{n-1}|\mathcal{M}_E)p(X_n|\mathcal{M}_E).$$

This assumption is invalid in reality as the random variables are correlated in time. Since there is no link between X_{n-1} and X_n, the random variables are assumed to be independent and identically distributed. In the joint modeling approach with the model $\mathcal{M}_J$, the correlation between their variables X_n and X_{n-1} is included as

$$p(X_n|X_{n-1}, \cdots, X_1, \mathcal{M}_J), \tag{1.7}$$

hence the conditional independence assumption is invalid. Here, we restrict the model in Equation (1.7) in a particular form, which is called the Markov models (Bishop, 2006). This is a common choice for the estimation of the joint probability distribution, which we

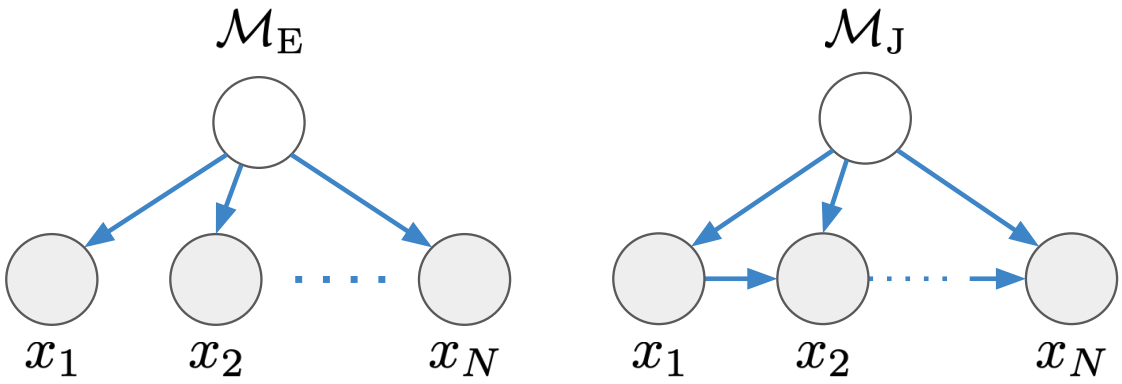


Figure 1.6: Graphical model of the energy spectrum (left) and the joint model (right). A gray-filled circle represents an observable variable. The random variables of the energy spectrum is independent and identically distributed. The joint model follows the markov model.

adopt in this thesis. The graphical model of the energy spectrum and the joint models are indicated in Figure 1.6.

1.4 Advances for Statistical Modeling

The statistical modeling approach appears very straightforward and intuitive, and the advantages over conventional analysis are obvious. However, this approach has not been explored in X-ray astronomy, including BHB data (Pottschmidt et al., 1998). If we retrieve the literature using the Astrophysics Data System (ADS) database, the number of papers including the keywords "black hole" and "autoregressive" model, which is a representative time series model, is only 22 as of writing. This should be compared with the entire volume of the literature by retrieving the keywords "black hole" and "power-law" model, which is a representative energy spectrum model, of 4893. Why is this? We point out three reasons in the following that hampered the extensive use of statistical modeling and recent advances to clear these obstacles.

1.4.1 Observation Data Quality

One reason is the lack of photon counts to explore the utility of the statistical modeling. The paucity of counts is evident in the early days (Figure 1.1) with $\mathcal{O}(10 \text{ counts s}^{-1})$. The situation gradually improved over the years, and a giant leap was made with the advent of the Neutron star Interior Composition Explorer (NICER) in 2017. The unprecedented collecting area and precision in arrival time determination of X-ray photons make it possible to accumulate observed data of BHBs uncompromised by the detector dynamic range and the Poisson statistics up to the frequency of $\mathcal{O}(1 \text{ kHz})$. The count rate in Figure 1.2 is $\mathcal{O}(10^4 \text{ counts s}^{-1})$, which is a 10^3-fold increase from the early days. With this improvement in the data quality, it has now become practical to apply the statistical modeling approach even including the joint spectral and timing modeling.

1.4.2 Development of Modeling and Computational Resources

Statistical modeling requires massive computer resources. With recent advances in computing technology, it is now feasible to experiment with multiple models and parameters of statistical models, unless they are very complex. Parallel CPUs and GPUs are available for end users, which are indeed used in this work. This hardware availability is further enhanced by software development that allows the construction of statistical models. Languages such as R and Python provide access to statistical analysis libraries and probabilistic programming languages, making statistical model analysis straightforward.

A field that recognized this change early is the field of economics, in which a large knowledge base is being built upon application of the statistical modeling to the actual data, including multivariate time-series data. We should follow this in the field of astronomy to open new avenues for data exploration.

1.4.3 Physical Interpretation

Yet another reason against the time series modeling is the difficulty in relating the results with physical models, unlike energy spectral analysis (Gilfanov, 2010). Both light curve analysis and time series modeling are predominantly phenomenological, lacking a direct derivation from specific physical models. Nevertheless, some aspects of the BHB can only be achieved by analyzing the variability. The quasi-periodic oscillation (QPO) and the time lag are the two most important examples (reviewed in Remillard and McClintock, 2006; Uttley et al., 2014). We should explore them more seriously on the basis of physical models; after all, the rapid variability is a distinctive feature of BHBs from other sources.

Recently, physical models have been developed for these well-known phenomena (e.g., Fabian et al., 1989; Remillard and McClintock, 2006). In the near future, these physical models will be able to predict both the spectral and timing behaviors of BHBs, which we can compare with observations. The parameters of the physical models will be treated as random variables in the statistical modeling approach and their probability distribution will be derived through application to the actual data.

Given these advancements in data quality, modeling techniques, computational resources, and theories, it is high time to overcome the difficulties of statistical modeling and make it a new norm of data analysis in X-ray astronomy. We are going to demonstrate this to be possible with the joint spectral and timing modeling of a BHB, MAXI J1820 + 070, observed with NICER.

Chapter 2

Design of Research

Contents

2.1 Purpose

As outlined in Chapter 1, the conventional spectral and light curve analyses of BHBs have many limitations. On the one hand, energy spectral analysis is powerful for deriving spectral components of physical entities, but it is often performed for time-slice spectra without considering their correlation in time. On the other hand, the light curve analysis describes the correlation in time, but ignores the spectral changes that make varying contributions of spectral components in time. These limitations can be overcome only by the joint modeling of the spectra and light curves. The statistical modeling presents a comprehensive approach that leverages all available information on the subject, including raw data, theories, and rules of thumb, to achieve tasks such as information extraction, knowledge discovery, and prediction. This is the approach that we should take.

The time series counterpart of the statistical modeling is commonly referred to as time series modeling. This is typically accomplished through the use of a state-space model, which describes the phenomena using system equations and observation equations. The system equation describes the development of the states, whereas the observation equation describes the process of observing these states. The novel idea for this research is to relate the former with the traditional light curve analysis and the latter with the traditional spectral analysis, so that we can model both in a single framework. The primary goal of this research is to demonstrate that such a method is possible and even useful in obtaining new insights into BHBs by applying the method to actual data.

2.2 Scope

To achieve the purpose, the investigation will be made up of the following steps.

a. Application of the classical time series modeling: In this step, we employ the classical time series models, such as the autoregressive model and its variant, to model the observed time series data. This method has not been widely used for studying the X-ray light curves of BHBs. The goal is to gain insight into the information obtained from these models and understand their limitations. This step serves as a preliminary exploration before constructing more advanced models.

b. Development of the state-space models for the joint spectral and timing analysis: In the state-space models, the state and observation variables are distinguished. The success of state-space modeling lies in the appropriate choice of these variables, especially the state (or latent) variables. We use the physical components as the state variable and multiband light curves as the observation variables. We demonstrate that this method works efficiently for the joint spectral and timing models.

c. Demonstration of the method using actual data of BHBs and gain insight into their physical nature: We applied the joint spectral and timing models to real astronomical data. Specifically, we apply the model focusing on the rapid variability of a

BHB observed with NICER in the low hard state and show that we can extract new insights based on the joint spectral and timing modeling.

In this research, we focus on the best and minimum data set to achieve the goals. We do not distract ourselves by dealing with all the available data. We focus on one BHB taken with one instrument for a short period. This alone contains sufficiently rich information that can only be unveiled through the joint spectral and timing analysis.

2.3 Structure

The rest of the thesis proceeds as follows.

Chapter 3. Time Series Analysis

We provide a concise review of the time series analysis. This includes the general considerations of statistical modeling (such as model inference, selection, inspection, and classification) in § 3.1, and description functions in § 3.2. In this thesis, we call cross spectra, cross correlation functions, etc all together as "description functions". They are defined mathematically and used throughout the thesis to present the results of various modeling. We also describe the mathematical background of the classical time series modeling (autoregression model and its variant) in § 3.3 and the state space modeling in § 3.4.

Chapter 4. X-ray Variability of BHBs

We review the X-ray variability of BHBs. We show that BHBs have several typical states depending on the physical nature of the accretion disk, which can be distinguished by the spectral and timing characteristics (§ 4.1). We present some results obtained by the traditional light curve analysis focusing on time lags. We clarify the limitations of the traditional approach and argue for the need for latent variables. (§ 4.2).

Chapter 5. Observing Facility and Target

We describe the observing facility NICER (§ 5.1) and the target MAXI J1820+070 (§ 5.2). We summarize the result of MAXI J1820 + 070 observations mainly made with NICER. We pick up one data set that is best suited to demonstrate our joint spectral and timing analysis.

Chapter 6. Analysis and Results

This is the main part of this thesis. We start with the generation and description of the time series data (§ 6.1). We then apply the statistical modeling for both the classical modeling (§ 6.2) and the state-space modeling (§ 6.3). Description functions are derived based on these approaches. We will show that the joint spectral and timing approach successfully decomposes the spectral and timing variations mixed in the data.

Chapter 7. Discussion

We extract information from the joint spectral and timing modeling and translate it into the physics of BHBs. In particular, we focus on the geometrical structure of BHBs in the low hard state, such as the accretion disk and accretion flow. From the variability time scale and time lags of individual spectral components, we give an interpretation to consistently explain the observed findings based on the truncated accretion disk geometry.

Chapter 8. Summary

We summarize the key findings of this work and conclude the successful application of the joint spectral and timing analysis. We also present some perspectives for future extensions of the statistical modeling approach.

Chapter 3

Time Series Analysis

Contents

3.1 Statistical Modeling

In this section, we describe the statistical modeling framework that incorporates prior information, focusing on the design, estimation, and interpretation of statistical models. We provide a concise overview of probability distributions and statistical models, which form the fundamental basis for statistical modeling. We also discuss model estimation and selection. More technical descriptions can be found in Kitagawa (2022) and Bishop (2006).

3.1.1 Pre-processing of Data

When analyzing data with time series models, the observed data often do not meet the conditions assumed by the model. In such cases, pre-processing of data is effectively for successful modeling. Some of the commonly-used pre-processing are shown below.

Normalization Standardization in time series analysis refers to the process of transforming the values of a time series to a common scale, making it easier to compare and analyze different variables or series. This is important because time series data often have different units or scales. The standardization process typically involves subtracting the mean μ of the series and dividing by the standard deviation σ, that is $z_n = (y_n - \mu)/\sigma$.

Scale Transformation For data with fluctuations that span multiple orders of magnitude, the scale transformation is useful for consistent treatments of both large and small variations. A frequently used technique involves applying a logarithmic transformation to the data y_n, such as $z_n = \log y_n$.

Differencing For time series data with a linear trend, it is useful to remove the trend by taking the differences. When a time series y_n with a linear trend is expressed as $y_n = a + bn$. The resulting difference series z_n is given by:

$$\begin{aligned} z_n &= y_n - y_{n-1} \\ &= b, \end{aligned} \tag{3.1}$$

where a is canceled. The same repeats for higher order trends.

Signal Filtering To smooth a highly fluctuating time series, low-pass filtering is often used. The most simple low-pass filter is the moving average. The moving average t_n for a time series y_n, comprising $2k + 1$ terms, is calculated as the average of the sum of k terms both preceding and following y_n:

$$t_n = \frac{1}{2k+1} \sum_{j=-k}^{k} y_{n+j} \tag{3.2}$$

3.1.2 Probability Distribution and Statistical Model

A random variable is a variable whose values follow a probability distribution. If we denote a random variable as Y and its value as y, the probability distribution function is defined as:

$$G(y) = \text{Prob}(Y \leq y) \tag{3.3}$$

In the case of continuous random variables, the probability distribution function $G(y)$ can be expressed as the integral of the probability density function $g(t)$ over the interval $(-\infty, y)$:

$$G(y) = \int_{-\infty}^{y} g(t)dt. \tag{3.4}$$

Here, $g(y)$ is referred to as the probability density function. The probability that the random variable Y falls within the interval $a \leq Y \leq b$ is calculated as:

$$\text{Prob}(a \leq Y \leq b) = G(b) - G(a) = \int_{a}^{b} g(t)dt \tag{3.5}$$

Various probability distributions are used to describe the characteristics of data. Values generated in accordance with the magnitude of the probability density function are referred to as realizations of a random variable. The probability density function of the Gaussian distribution is given by:

$$g(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}, \quad -\infty < x < \infty. \tag{3.6}$$

Here, μ is the mean and σ^2 is the variance, both of which are parameters of the distribution. This distribution is denoted as $\mathcal{N}(\mu, \sigma^2)$. The probability density function of the Gaussian distribution forms a bell-shaped curve centered around the mean, and realizations of this distribution are more likely to be found near the mean (Figure 3.1).

When a probability distribution is given, realizations of that distribution can be generated from random numbers. In statistical analysis, when observations $y_1, \cdots, y_N$ are obtained, they are considered realizations from a certain random variable. The distribution $g(y)$ that characterizes this random variable is referred to as the true distribution. Typically, the true distribution is not known. Estimating the true distribution from observed data is called statistical modeling. In the example of the Gaussian distribution illustrated in Figure 3.1, this corresponds to estimating the parameters μ and σ^2 of the true Gaussian distribution from the observed data.

When observations are considered as realizations from the same distribution independently from each other, it is sufficient to estimate a single probability distribution. However, for the time series analysis, the data are not independent. It is thus necessary to determine the joint probability distribution $f(y_1, \cdots, y_N)$.

Representing a time series $y_1, \cdots, y_N$ by a sample mean $\hat{\mu}$ and sample auto covariance matrix $\hat{C}_k$ assumes that y_i is the realization from a multivariate normal distribution with

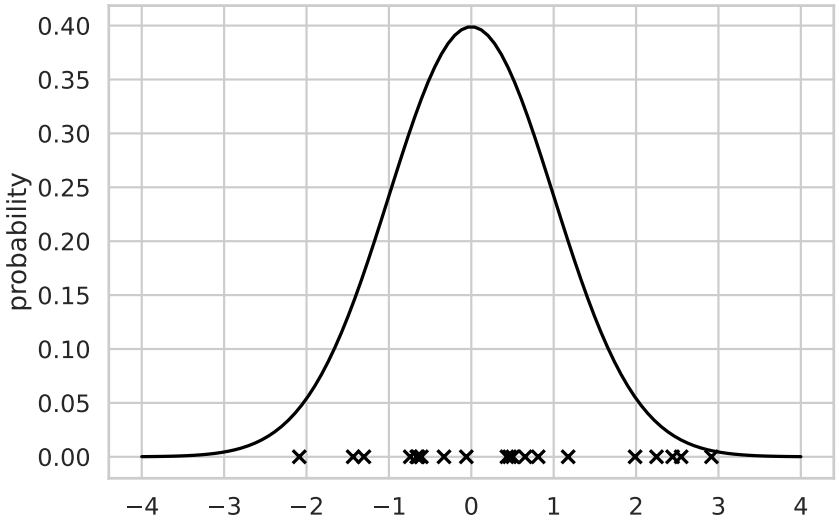


Figure 3.1: Gauss distribution and its realization. The mean and variance are zero and one. Twenty realizations are generated from the distribution.

a mean vector $(\hat{\mu}, \cdots, \hat{\mu})^T$ and a covariance matrix

$$\hat{C} = \begin{bmatrix} \hat{C}_0 & \hat{C}_1 & \cdots & \hat{C}_{N-1} \\ \hat{C}_1 & \hat{C}_0 & \cdots & \hat{C}_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{C}_{N-1} & \hat{C}_{N-2} & \cdots & \hat{C}_0 \end{bmatrix}. \tag{3.7}$$

Here, $\hat{C}_k = \mathrm{E}[(y_n - \hat{\mu})(y_{n-k} - \hat{\mu})]$, in which $\mathrm{E}[\cdot]$ is the expected value. This requires estimating $N + 1$ values from N data points. This approach does not efficiently capture the information in the data for too many parameters. Therefore, we often use more constrained modeling in the time series analysis. For example, in the m-dimensional autoregressive model (§ 3.3.1) with the variance of the error term following a Gaussian distribution, the time series information is aggregated into $m + 1$ values.

3.1.3 Model Inference

The goal of statistical modeling is to estimate a statistical model that approximates the true distribution. To achieve this, we need criteria to measure how close the model is to the true distribution. In this context, we use the Kullback-Leibler (K-L) information as a criterion. Using this criterion, we naturally derive the log-likelihood as a measure of the relative goodness of fit for models. Here, we define the K-L information and transform it into an estimable form to derive the log-likelihood, which can be used in model estimation.

The K-L information is a measure of the difference between two distributions $g(y)$ and $f(y)$. It is calculated as follows:

$$I(g;f) = \mathrm{E_Y}\left[\log \frac{g(Y)}{f(Y)}\right] = \int_{-\infty}^{\infty} \log\left\{\frac{g(y)}{f(y)}\right\} g(y)dy, \tag{3.8}$$

where $\mathrm{E}_Y(\ \cdot\ )$ is the expected value over Y. The second equation is applicable when the random variable is continuous. Essentially, it computes the expected value of the logarithmic difference between $g(y)$ and $f(y)$ weighted by $g(y)$. K-L information equals zero when $g(y) = f(y)$, and it takes positive values when $g(y) \neq f(y)$. Smaller K-L information values indicate that $g(y)$ and $f(y)$ have distributions that are closer to each other. The negative of the K-L information is known as the Boltzmann entropy $B(g;f) = -I(g;f)$. The model that maximizes this entropy is considered the best approximation of the true distribution, which is referred to as the entropy maximizing principle in statistical modeling.

Let's calculate the K-L information for two Gaussian distributions. Suppose the true distribution $g(y)$ and the model distribution $f(y)$ are both Gaussian distributions defined as follows:

$$g(y|\mu,\sigma^2) = \frac{1}{2\pi\sigma^2}\exp -\frac{(y-\mu)^2}{2\sigma^2} \tag{3.9}$$

$$f(y|\xi,\tau^2) = \frac{1}{2\pi\tau^2}\exp -\frac{(y-\xi)^2}{2\tau^2} \tag{3.10}$$

In this case, the KL information can be calculated as:

$$\begin{aligned} I(g;f) &= \mathrm{E_Y}\left[\log\frac{g(y)}{f(y)}\right] \\ &= \frac{1}{2}\left\{\log\frac{\tau^2}{\sigma^2} - \frac{\mathrm{E_Y}[(Y-\mu)^2]}{\sigma^2} + \frac{\mathrm{E_Y}[(Y-\xi)^2]}{\tau^2}\right\} \\ &= \frac{1}{2}\left\{\log\frac{\tau^2}{\sigma^2} - 1 + \frac{\sigma^2+(\mu-\xi)^2}{\tau^2}\right\} \end{aligned} \tag{3.11}$$

It is evident that when the distributions are the same, i.e., $\mu = \xi$ and $\tau = \sigma$, then $I(g;f) = 0$. As an example, if the true distribution $g(y)$ follows a standard normal distribution $\mathcal{N}(0,1)$ and the model distribution $f(y)$ is chosen to be $\mathcal{N}(0.1, 1.5)$, then $I(g;f) = 0.03940$.

It is possible to evaluate the quality of a model using the K-L information. In real problems, however, the true distribution is rarely known. We only have observed data $y_1, \ldots, y_n$, which can be considered as realizations of the true distribution. Let's consider how to compute K-L information using these data. K-L information can be decomposed as follows:

$$I(g;f) = \mathrm{E_Y}[\log g(Y)] - \mathrm{E_Y}[\log f(Y)] \tag{3.12}$$

The first term on the right-hand side cannot be calculated without knowing the true distribution. However, since it does not depend on $f(y)$, we can ignore it. The K-L

information is minimized when the second term on the right-hand side is maximized, so a model that maximizes this term is considered a good model. This term is known as the average log-likelihood. For continuous models with the probability density function $g(y)$, it can be expressed as:

$$\mathrm{E_Y}[\log f(Y)] = \int \log f(y) g(y) dy \tag{3.13}$$

This allows us to assess the goodness of the model in terms of its capability to capture the data distribution.

This integral includes the true model, $g(y)$, making it impossible to compute directly from the data. However, due to the fact that the data y_n are generated according to the true model $g(y)$, the law of large numbers tells us that, as the number of data points N approaches infinity,

$$\frac{1}{N} \sum_{n=1}^{N} \log f(y_n) \longrightarrow \mathrm{E_Y}[\log f(Y)] \tag{3.14}$$

In other words, when we maximize $\sum_{n=1}^{N} \log f(y_n)$, we can minimize the K-L information. This quantity is known as the log-likelihood when observations are independently obtained. It can be denoted as $\ell = \sum_{n=1}^{N} \log f(y_n)$.

When the model is a parametric model with parameters θ and is represented as $f(Y) = f(Y|\theta)$, the log-likelihood becomes a function of the parameters θ. This is referred to as the log-likelihood function of θ, and finding θ that maximizes ℓ is considered a way to select a good model.

3.1.3.1 Frequentist Inference

In this research, we employ both frequentist and Bayesian approaches to estimate statistical models, taking into account various factors, including the availability of information and computational resources. We now provide a summary of these two approaches.

The maximum likelihood method aims to maximize the likelihood function, which is obtained by taking the exponent of the log-likelihood function. This is the frequentist approach that involves estimating parameters that maximize the probability of the observed data. The likelihood function, which is a conditional probability of the model parameters θ given the observed data Y, is represented as $p(Y|\theta)$. In the frequentist perspective, θ is considered as fixed parameters, and it is assumed that the observed data Y are obtained from the probability distribution governed by these parameters. In many cases, optimization is performed to find the parameters that maximize the log-likelihood.

One computational method for maximum likelihood estimation is the quasi-Newton method. This method starts by selecting an initial value for the parameters θ_0 and using this value to compute the log-likelihood function. The parameters θ are updated using the first derivative of the log-likelihood, $\partial\ell/\partial\theta$, as follows:

$$\theta_k = \theta_{k-1} + \lambda_k H_{k-1} \frac{\partial \ell}{\partial \theta} \tag{3.15}$$

where the step size λ_k and the Hessian matrix H_{k-1} are determined automatically. If the first derivative of $\ell(\theta)$ can be analytically computed, the process is relatively straightforward. However, in many cases, $\ell(\theta)$ takes a complex form, which makes direct computation difficult. In practice, numerical differentiation is often used.

To estimate the uncertainty of the maximum likelihood estimates, one can consider the probability distribution of the available data set Y while taking into account the distribution of the parameters. One approach is bootstrapping, which involves repeated random sampling, with replacements, of N data points from the original data set $Y = y_1, \cdots, y_N$ a total of L times. Maximum likelihood estimation is performed for each of the bootstrap samples of a size N, and the distribution of the estimated parameters is used to assess the error.

3.1.3.2 Bayesian Inference

Bayesian model estimation is a method that involves interpreting the data and the parameters based on the Bayes' theorem to estimate the probability distribution of the parameters. In Bayes' theorem, when we have two random variables, A and B, and we want to find the probability of B given that A has occurred, it is expressed as follows:

$$p(B|A) = \frac{p(A|B)p(B)}{p(A)} \tag{3.16}$$

Here, $p(B|A)$ and $p(A|B)$ represent the conditional probability of B given A and that of A given B, respectively. Now, if we consider B as the model parameter θ and A as the observed data Y, we can rewrite Bayes' theorem as follows:

$$p(\theta|Y) = \frac{p(Y|\theta)p(\theta)}{p(Y)} \tag{3.17}$$

In this context, $p(Y|\theta)$ represents the probability of obtaining the observed data Y when the model parameters θ are known, which is reffered to as the likelihood function. $p(\theta|Y)$ represents the conditional probability of the model parameters θ after observing the data, which is referred to as the posterior distribution. $p(\theta)$ is called the prior distribution, which is the probability distribution of the model parameters before observing any data. These terms, such as the prior distribution and the posterior distribution, are named to reflect the Bayesian idea that we want to find the distribution of the model parameters θ given the observed data Y.

In the frequentist maximum likelihood estimation, we seek to maximize the likelihood function $p(Y|\theta)$ to find θ. In the Bayesian approach, we aim to estimate θ that maximizes the posterior distribution $p(\theta|Y)$. Since the denominator in Equation (3.17) represents the probability of the observed data, which becomes constant once the data are obtained, we only need to maximize the numerator. The numerator is the product of the likelihood function and the prior distribution, so we consider not only the likelihood of the data, but also the prior knowledge of the parameters θ. This can be advantageous in cases where incorporating prior knowledge is important. For example, if one tosses a coin three

times, and it comes up with the head all three times, the frequentist maximum likelihood estimation suggests that the probability of the head is 1. In the Bayesian approach, we can avoid such extreme conclusions by imposing a prior that represents fairness.

The process of estimating θ that maximizes the posterior probability is known as Maximum A Posteriori (MAP) estimation. However, the MAP estimation focuses on finding the maximum value of the posterior, and thus is not a full Bayesian approach. A full Bayesian approach involves calculating the entire posterior distribution, which can be achieved using techniques such as the Markov Chain Monte Carlo (MCMC). These calculations involve integrating probability densities and sums, and they often require significant computational resources. Consequently, traditional statistical analyzes have mainly relied on frequentist approaches. In recent years, however, the advances in computing power and algorithms have made it feasible to estimate posterior probability distributions, and methods such as variational inference have been developed, making the Bayesian estimation more accessible.

The Metropolitan-Hastings (MH) algorithm is a basic technique in the MCMC methods, employing any distribution as its proposal distribution (Hastings, 1970). It calculates the acceptance probability of the proposed samples, determining acceptance or rejection in a single step using random numbers. Hamiltonian Monte Carlo (HMC) takes advantage of the principles of the Hamiltonian mechanics in physics (Duane et al., 1987). Notably, HMC allows for more efficient sampling by facilitating jumps to more distant samples compared to the MH algorithm. An extended version of HMC is the No-U-Turn Sampler (NUTS) (Hoffman and Gelman, 2014). HMC requires adjustment of parameters such as step size and number of steps, and suboptimal values can hinder convergence. NUTS addresses this by dynamically adjusting these parameters during the sampling process. Additionally, NUTS automatically avoids U-turn sampling, enhancing the efficiency of estimation for subject experts.

3.1.3.3 Choice between Frequentist and Bayesian Approaches

There are no simple criteria for choosing between a frequentist and a Bayesian approach. It should be based on factors such as the desired level of interpretation and computational costs. Here, we will discuss general points to consider when deciding which approach to adopt.

One distinguishing feature of the Bayesian approach is its ability to incorporate prior knowledge as a prior distribution. If there is substantial prior knowledge, such as strong theoretical predictions for the problem at hand, the Bayesian approach might be preferable. However, it is essential to note that the posterior probability includes prior knowledge as a prior distribution, requiring careful consideration. Conversely, in cases with limited prior knowledge or significant impact from the prior distribution, the frequentist approach can be more practical.

The degree of interpreting the results obtained can be another deciding factor between the two approaches. In the Bayesian approach, estimated parameters are expressed as probability distributions, simplifying interpretation by representing uncertainty through

the spread of the distribution. In contrast, the frequentist approach relies on the point estimation, and assessing uncertainty usually involves more complex procedures like confidence intervals.

Computational costs play a role in choosing between the two approaches. Generally, Bayesian computations to determine the posterior distribution can be intricate, often requiring numerical approximations or sampling methods. Conversely, the frequentist approach, with methods like maximum likelihood estimation, tends to have lower computational costs. The availability of software tailored to the problem is crucial. The frequentist approach are often better supported than the Bayesian approach. However, the advent of the "probabilistic programming languages" (e.g., Stan (Stan Development Team, 2024), PyMC (Salvatier et al., 2015), TensorFlow Probability (Dillon et al., 2017)) has made the Bayesian approach more accessible. These languages often incorporate features such as automatic differentiation and GPU utilization, facilitating fast computation. However, due to their unique syntax and the need to understand the Bayesian statistics in detail, the learning cost of users is high.

3.1.4 Model Selection

The Akaike Information Criterion (AIC) is one of the commonly used model selection criteria used to choose the most appropriate model among multiple candidate models to explain the data. It might seem logical to compute the log-likelihood function for each model and compare them, selecting the model that maximizes it. However, in the case of models defined by the maximum likelihood estimates $\hat{\theta}$, the quantity $N^{-1}\ell(\hat{\theta})$ exhibits a positive bias as an estimate of $\mathrm{E}_{\mathrm{Y}}[\log f(Y|\hat{\theta})]$, making it unsuitable for model comparison. This bias arises from the use of the same data twice for parameter estimation and evaluation.

To address this issue, AIC was developed, and it is expressed as:

$$\mathrm{AIC} = -2\ell(\hat{\theta}) + 2k \tag{3.18}$$

Here, $\ell(\hat{\theta})$ represents the maximum log-likelihood, and k is the number of parameters. In this research, we use AIC as the criterion to compare different statistical models when selecting models.

3.1.5 Model Inspection

To assess the accuracy of the estimated model in describing the observed data, many methods have been developed. We review some of the commonly used methods below.

3.1.5.1 White Noise Tests for Residuals

For the best model, the residual of the data from the model follows white noise; no further useful information is obtained from the white noise to improve the model. To assess whether the residuals follow white noise, it is crucial to verify that their autocorrelation

(see § 3.2) is zero. White noise is examined by testing the null hypothesis $R_k = 0$ against the alternative hypothesis $R_k \neq 0$ for the autocorrelation coefficient $\hat{R}_k$ at the desired lag k. The asymptotic distribution of $\hat{R}_k$ follows a normal distribution $\mathcal{N}(0, 1/N)$, where N represents the number of data points used for the sample autocorrelation function calculation. For example, with $N = 100$, the standard deviation is 0.1. The critical value for a two-sided 95 % confidence interval is 1.96 for the normal distribution. Therefore, if $|\hat{R}_k| > 0.196$, the null hypothesis $R_k = 0$ is rejected, indicating a significant correlation at a 5 % significance level.

It should be noted that the earlier test examines whether each autocorrelation coefficient is zero individually for k. When testing if all k's are zero, which we really want to investigate, different types of tests are required. The objective is to test the null hypothesis $R_1 = R_2 = \cdots = R_m = 0$ against the alternative hypothesis that $R_k \neq 0$ at least for one $k \in [1, m]$. This test, known as the portmanteau test, can be performed using various statistics (Hamilton, 1994). A commonly employed test in this context is the Ljung-Box test. The statistic defined as

$$\begin{aligned} Q(m) &= N(N+2) \sum_{k=1}^{m} \frac{\hat{R}_k{}^2}{N-k} \\ &\sim \chi^2(m) \end{aligned} \tag{3.19}$$

asymptotically approaches to a chi-square distribution (Ljung and Box, 1978). Here, χ^2 represents the chi-square distribution. By comparing the 95 % point of $Q(m)$ with that of $\chi^2(m)$, if $Q(m)$ is greater, the null hypothesis is rejected at a 5 % significance level. The portmanteau test allows the evaluation of whether the residuals follow white noise, offering insight into the accuracy of modeling.

3.1.5.2 Q–Q Plot

A Q–Q plot is a graphical tool used in statistics to assess the similarity between the distribution of a sample dataset and a theoretical distribution, such as the normal distribution. In a Q–Q plot, the quantiles of the observed data are plotted against the quantiles of the expected theoretical distribution. The x-axis of the Q–Q plot represents the quantiles of the theoretical distribution, while the y-axis represents the quantiles of the observed data. If the data follow the theoretical distribution, the points on the Q–Q plot align along the diagonal line. Deviations from this line indicate departures from the expected distribution. Especially, when the theoretical distribution is the normal distribution, the Q–Q plot is called a normal Q–Q plot.

Q–Q plots are particularly useful for identifying deviations from normality in a dataset. For example, if the points in the Q–Q plot systematically deviate from the straight line, it suggests that the data may not be normally distributed. This graphical method provides a visual assessment of the assumption for the distribution of given data sets.

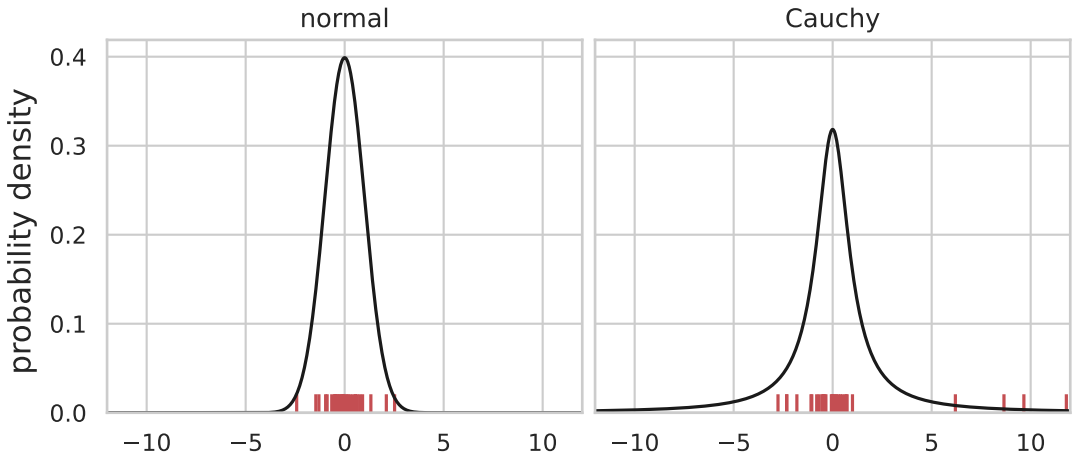


Figure 3.2: Samples from the normal and Cauchy distribution. The parameterization of both distributions is identical, with a location parameter of 0.0 and a scale parameter of 1.0.

In the Python context, SciPy offers the function to compute and plot Q–Q plot as `scipy.stats.probplot` (Virtanen et al., 2020). As an example, using the standardized samples from the normal and Cauchy distributions (Figure 3.2), we compute the normal Q–Q plots (Figure 3.3). In the normal Q–Q plot of samples from the normal distribution, the data points align closely with the diagonal line. However, in the normal Q–Q plot of Cauchy samples, noticeable deviations occur, especially in the distant part from the center. This characteristic highlights that while the center of the Cauchy distribution resembles the normal distribution, the tail parts differ.

3.1.6 Model Classification

Time series analysis involves various aspects. Each aspect leads to a specific classification. Understanding the unique features of the time series analysis is crucial for conducting suitable analyses and developing effective models. In this context, we provide an overview of some of the classification of time series modeling (Kitagawa, 2022).

Linear Time Series versus Non-Linear Time Series Linear time series analysis deals with sequences of observations generated by linear models. When the observed time series does not follow a linear model, it falls into the category of non-linear time series.

Stationary Time Series versus Non-Stationary Time Series A stationary time series is a sequence of observations that appears to have random fluctuations but can be seen as values derived from a probability distribution that remains unchanged over time. Stationary time series can be categorized as either weak stationary or strong stationary. In the weak stationary time series, the mean, variance, and covariance remain constant across time; i.e.; they only depend on the relative time shifts. In the strong stationary time series, the probability distribution remains unchanged even with time shifts. Time

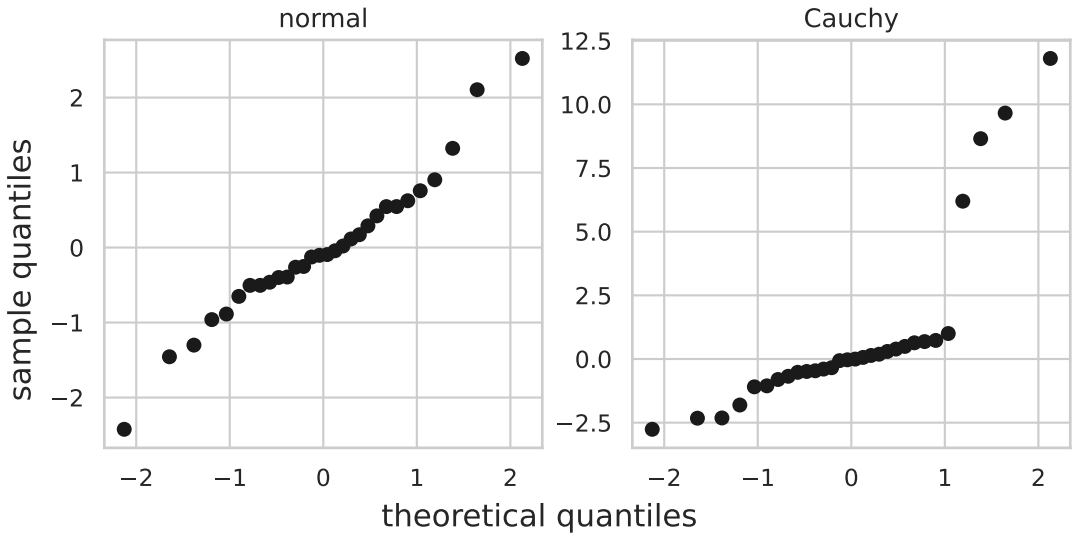


Figure 3.3: Normal Q–Q plots for the samples generated from the normal and Cauchy distributions.

series that deviate from these stationary characteristics are referred to as non-stationary time series.

Gaussian Time Series versus Non-Gaussian Time Series A Gaussian time series is the one in which the distribution of random variables follows a Gaussian (normal) distribution. Conversely, if the distribution deviates from a Gaussian form, it is termed a non-Gaussian time series.

3.2 Description Functions

In this thesis, we call functions that characterize the data as "description functions", which we define below separately for univariate and multivariate cases. The description functions calculated directly from the data are called "sample description function", hereafter.

3.2.1 Univariate

Autocovariance Function and Autocorrelation Function For weak stationary time series, the mean function μ_n remains constant over time and is defined as

$$\mu_n = \mathrm{E}[y_n], \tag{3.20}$$

representing the mean of the time series. The covariance between y_n and y_{n-k}, denoted as C_k, is a function solely dependent on the lag k and can be expressed as

$$\begin{aligned} C_k &= \mathrm{Cov}(y_n, y_{n-k}) \\ &= \mathrm{E}[(y_n - \mu)(y_{n-k} - \mu)], \end{aligned} \tag{3.21}$$

This function is known as the autocovariance function, which is an even function $C_{-k} = C_k$ satisfying $|C_k| \leq C_0$. The autocorrelation function is defined by

$$\begin{aligned} R_k &= \mathrm{Cor}(y_n, y_{n-k}) \\ &= \frac{\mathrm{Cov}(y_n, y_{n-k})}{\sqrt{\mathrm{Var}(y_n)\mathrm{Var}(y_{n-k})}} \\ &= \frac{C_k}{C_0}, \end{aligned} \tag{3.22}$$

where $\mathrm{Var}(y_n)$ is the variation of y_n defined as $\mathrm{Var}(y_n) = \mathrm{Cov}(y_n, y_n)$.

Partial Autocorrelation Function In the autocorrelation function, the correlation between e.g., y_n and y_{n-3} incorporates the effects of intermediate correlations such as between y_n and y_{n-1} or y_{n-2}, thus it is not a pure correlation function between y_n and y_{n-3}. The partial autocorrelation refers to the correlation between y_n and y_{n-k} after excluding the effects of the intermediate correlations. This is known as the Partial Autocorrelation Function (PACF), which is a function of the lag k.

The PACF can be computed by decomposing the covariance between y_n and y_{n-k} into the covariance with all intermediate observations. Consider the scenario where the standardized time series is represented by a noise-free k-th order autoregressive representation given by:

$$y_n = a_1^k y_{n-1} + a_2^k y_{n-2} + \cdots + a_k^k y_{n-k} \tag{3.23}$$

Here, a_i^k denotes the i-th coefficient in the k-th order AR model. Multiplying both sides by y_{n-k} and taking the expected value, we have:

$$\mathrm{E}[y_n y_{n-k}] = a_1^k \mathrm{E}[y_{n-1} y_{n-k}] + a_2^k \mathrm{E}[y_{n-2} y_{n-k}] + \cdots + a_k^k \mathrm{E}[y_{n-k} y_{n-k}] \tag{3.24}$$

This expression decomposes the covariance between y_n and y_{n-k} into covariances with the intermediate observations, providing the desired form. Using covariances $C_i = \mathrm{E}[y_n y_{n-i}]$, the above equation can be expressed as:

$$C_k = a_1^k C_{k-1} + a_2^k C_{k-2} + \cdots + a_k^k C_0 \tag{3.25}$$

This equation involves k unknown variables, making it unsolvable. However, by applying the same procedure to Equation (3.23) and multiplying both sides by y_{n-i} for $i = 1, \cdots, k$, we obtain:

$$\begin{aligned} C_1 &= a_1^k C_0 + a_2^k C_1 + \cdots + a_{k-1}^k C_{k-1} \\ C_2 &= a_1^k C_1 + a_2^k C_0 + \cdots + a_{k-1}^k C_{k-2} \\ &\vdots \\ C_k &= a_1^k C_{k-1} + a_{k-2}^k C_{k-2} + \cdots + a_{k-1}^k C_0 \end{aligned} \tag{3.26}$$

This allows us to compute the values for k. In the matrix form,

$$\begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_k \end{bmatrix} = \begin{bmatrix} C_0 & C_1 & \cdots & C_{k-1} \\ C_1 & C_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & C_1 \\ C_{k-1} & \cdots & C_1 & C_0 \end{bmatrix} \begin{bmatrix} a_1^k \\ a_2^k \\ \vdots \\ a_k^k \end{bmatrix} \tag{3.27}$$

Thus, the partial autocorrelation is calculated by

$$\begin{bmatrix} a_1^k \\ a_2^k \\ \vdots \\ a_k^k \end{bmatrix} = \begin{bmatrix} C_0 & C_1 & \cdots & C_{k-1} \\ C_1 & C_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & C_1 \\ C_{k-1} & \cdots & C_1 & C_0 \end{bmatrix}^{-1} \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_k \end{bmatrix}. \tag{3.28}$$

Power Spectrum When the condition

$$\sum_{k=-\infty}^{\infty} |C_k| < \infty,$$

is met signifying the rapid decay of the autocovariance function C_k, the Fourier transform of C_k can be computed. The function defined over the range $-1/2 \leq f \leq 1/2$

$$p(f) = \sum_{k=-\infty}^{\infty} C_k e^{-2\pi i k f} \tag{3.29}$$

is denoted as the power spectrum. Since the autocovariance function is an even function with $C_k = C_{-k}$, (3.29) can be reformulated as

$$p(f) = \sum_{k=-\infty}^{\infty} C_k \cos 2\pi k f = C_0 + 2\sum_{k=1}^{\infty} C_k \cos 2\pi k f. \tag{3.30}$$

Conversely, the autocovariance function is reconstructed from the power spectrum $p(f)$ as

$$C_k = \int_{-\infty}^{\infty} p(f) e^{-2\pi i k f} df = \int_{-\infty}^{\infty} p(f) \cos 2\pi i k f df. \tag{3.31}$$

3.2.2 Multivariate

More generalized description functions are used for multivariate cases. This is particularly relevant for this work, in which we analyze the multiband X-ray light curves. We introduce covariance and cross-spectrum below, which are commonly used as description functions for multivariate time series. It is important to note that they can also be applied to univariate cases, yielding results like autocovariance and power spectrum. For further details, refer to Kitagawa (2022).

Cross-Covariance and Cross-Correlation Functions Covariance serves as a basic analytical function to explore the characteristics of time series. For a multivariate time series $y_n = [y_n(1) \cdots y_n(M)]^T$, the mean function is defined as

$$\mu(i) = \mathrm{E}[y_n(i)], \tag{3.32}$$

resulting in the mean vector $\mu = [\mu_i \cdots \mu_M]$. The covariance between $y_n(i)$ and $y_{n-k}(j)$ is given by

$$\begin{aligned} C_k(ij) &= \mathrm{Cov}(y_n(i), y_{n-k}(j)) \\ &= \mathrm{E}[(y_n - \mu(i))(y_{n-k} - \mu(j))]. \end{aligned} \tag{3.33}$$

The $M \times M$ matrix

$$C_k = \begin{bmatrix} C_k(1,1) & \cdots & C_k(1,M) \\ \vdots & \ldots & \vdots \\ C_k(M,1) & \cdots & C_k(M,M) \end{bmatrix} \tag{3.34}$$

represents the covariance for the lag k. C_k as a function of the lag k is known as the cross-covariance function. The diagonal elements $C_k(i,i)$ of this function are referred to as the autocovariance function. Introducing the correlation coefficient between $y_n(i)$ and

$y_{n-k}(j)$ as

$$\begin{aligned} R_k(ij) &= \mathrm{Cor}(y_n(i), y_{n-k}(j)) \\ &= \frac{\mathrm{Cov}(y_n(i), y_{n-k}(j))}{\sqrt{C_0(i,i)C_0(j,j)}}, \end{aligned} \tag{3.35}$$

the matrix

$$R_k = \begin{bmatrix} R_k(1,1) & \cdots & R_k(1,M) \\ \vdots & \cdots & \vdots \\ R_k(M,1) & \cdots & R_k(M,M) \end{bmatrix} \tag{3.36}$$

is given, which is known as the cross-correlation function. The diagonal elements $R_k(i,i)$ are the autocorrelation function. Note that, for univariate time series, only autocovariance functions and autocorrelation functions are applicable.

Autocovariance functions and autocorrelation functions exhibit symmetry, making it necessary to calculate only positive values. However, this symmetry does not apply to the cross-covariance functions and the cross-correlation functions. Yet, a useful property is

$$C_k = C_{-k}^T, \quad R_k = R_{-k}^T \tag{3.37}$$

which means that we can focus on computing only for the positive k part of these functions as well.

When we have a set of observed values $y_n, \cdots, y_N$, the following calculations are performed using these values:

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{3.38}$$

$$\hat{C}_k(i,j) = \frac{1}{N}\sum_{n=k+1}^{N} (y_n(i) - \hat{\mu}(i))(y_{n-k}(j) - \hat{\mu}(j)) \tag{3.39}$$

$$\hat{R}_k(i,j) = \frac{\hat{C}_k(i,j)}{\sqrt{\hat{C}_0(i,i)\hat{C}_0(j,j)}} \tag{3.40}$$

We refer these as the sample mean, sample autocovariance function, and sample autocorrelation function, respectively.

Cross-Spectrum and Coherency The Fourier transform of the covariance matrix $C_k(s,j)$ is called the cross-spectral density function and is expressed as:

$$\begin{aligned} p_{sj}(f) &= \sum_{k=-\infty}^{\infty} C_k(s,j) e^{-2\pi i k f} \\ &= \sum_{k=-\infty}^{\infty} C_k(s,j)\cos 2\pi k f - i\sum_{k=-\infty}^{\infty} C_k(s,j)\sin 2\pi k f. \end{aligned} \tag{3.41}$$

When we define

$$P(f) = \begin{bmatrix} p_{11}(f) & \cdots & p_{1\ell}(f) \\ \vdots & \ddots & \vdots \\ p_{\ell 1}(f) & \cdots & p_{\ell\ell}(f) \end{bmatrix} \tag{3.42}$$

as the matrix of the cross-spectra for every pair of time series data, the relation between C_k and $P(f)$ can be expressed as:

$$\begin{aligned} P(f) &= \sum_{k=-\infty}^{\infty} C_k e^{-2\pi i k f} \\ C_k &= \int_{-\frac{1}{2}}^{\frac{1}{2}} P(f) e^{2\pi i k f} df \end{aligned} \tag{3.43}$$

The cross-spectrum is also represented as:

$$p_{jk}(f) = \alpha_{jk} e^{i\phi_{jk}(f)}, \tag{3.44}$$

where

$$\begin{aligned} \alpha_{jk}(f) &= \sqrt{(\Re\{p_{jk}(f)\})^2 + (\Im\{p_{jk}(f)\})^2} \\ \phi_{jk}(f) &= \arctan \frac{\Im\{p_{jk}(f)\}}{\Re\{p_{jk}(f)\}}. \end{aligned} \tag{3.45}$$

Here, $\Re$ and $\Im$ represent the real and imaginary parts. $\alpha_{jk}(f)$ is called the amplitude spectrum and $\phi_{jk}(f)$ is called the phase spectrum. The coherency is an equivalent value to the square of the correlation coefficient between the frequency components of $y_n(j)$ and $y_n(k)$ at frequency f, which is expressed as

$$\mathrm{coh}_{jk}(f) = \frac{\alpha_{jk}(f)^2}{p_{jj}(f) p_{kk}(f)}. \tag{3.46}$$

3.3 Autoregressive Models

In this section, we provide a concise overview of the classical time series model with examples. We refer to Kitagawa (2022) as a reference for this section.

3.3.1 Autoregressive Model

3.3.1.1 Definition

The autoregressive (AR) model is a phenomenological model that uses a linear combination of observed values in the past to express the observed value y_n at the time n. It is given by the equation:

$$y_n = a_1 y_{n-1} + \cdots + y_j x_{n-j} + v_n \tag{3.47}$$

$$= \sum_{j=1}^{m} a_j y_{n-j} + v_n, \tag{3.48}$$

where, a_j is the AR coefficient that corresponds to the j-th past value, and v_n is the white noise error term that is assumed to be normally distributed with a mean of zero and a constant variance. The AR model is a weakly stationary model, which implies that its mean and autocovariance are constant. The AR coefficients are estimated by autocovariance using observed data.

Stationarity The characteristic equation of the AR model is given by:

$$1 - \left(\sum_{j=1}^{m} a_j z^j \right) = 0, \tag{3.49}$$

The roots of this equation, termed characteristic roots, are closely related to the stationarity of the AR model. If these roots lie outside the unit circle, the AR model is non-stationary. For example, in the case of the first-order AR model, the characteristic equation is expressed as:

$$1 - a_1 z = 0, \tag{3.50}$$

leading to the stationarity condition:

$$|z| = |a_1^{-1}| > 1, \tag{3.51}$$

or $|a_1| < 1$.

Random Walk and Stationarity Test In the first-order AR model, the model with $a_1 = 1.0$ is specifically called random walk. It is expressed as $y_n = y_{n-1} + w_n$, where w_n denotes white noise. This type of random walk is termed a unit root process because its characteristic equation has a root on the unit circle. The Dickey-Fuller (DF) test exploits the fact that the root of a stationary time series lies within the unit circle.

The null hypothesis of the test asserts that the time series follows a unit root process, specifically with $a_1 = 1.0$. Rejecting the null hypothesis implies that the time series is likely stationary.

Although the DF test is designed for time series following first-order AR models, the augmented Dickey-Fuller (ADF) test is applicable to time series following higher-order AR models. Readers interested in a thorough understanding of the DF and ADF tests can refer to Hamilton, 1994.

3.3.1.2 Description Functions

In the AR model, various information can be extracted by transforming its coefficients into description functions. Here, we introduce representative examples including impulse response function, autocovariance function, partial autocorrelation function, and power spectrum.

In the traditional light curve analysis, these description functions, except for the impulse response function, are calculated from the data and commonly used to examine the characteristics of time series data. However, with this approach, the functions are influenced by noise, which can obscure the true features of the signal. In contrast, the AR model incorporates noise as model parameters, thus the AR coefficients create results that are less influenced by the noise than those directly computed from the data. In this section, we provide examples of these calculations using synthetic data.

Impulse Response Function The impulse response function quantitatively evaluates the impact of an impulse at a specific time point upon the data at subsequent time points. In the AR model, it can be expressed as an infinite sum of the product of previously added noise and impulse response function g_i, as follows:

$$y_n = \sum_{i=0}^{\infty} g_i v_{n-i} \tag{3.52}$$

where g_i is calculated iteratively using the following recursive formulas:

$$g_0 = 1 \tag{3.53}$$

$$g_i = \sum_{j=1}^{i} a_j g_{i-j}, \ i = 1, 2, \cdots \tag{3.54}$$

By performing sequential substitutions, we obtain the value of g_i.

Autocovariance and autocorrelation functions By multiplying y_{n-k} to Equation (3.48) and taking expectation, we obtain:

$$\mathrm{E}[y_n y_{n-k}] = \sum_{j=1}^{m} a_i \mathrm{E}[y_{n-i} y_{n-k}] + \mathrm{E}[v_n y_{n-k}]. \tag{3.55}$$

Using the equation of

$$\mathrm{E}[v_{n-i}y_{n-k}] = \sum_{j=0}^{\infty} g_j \mathrm{E}[v_{n-i}y_{n-k-j}] = \begin{cases} 0, & i > k \\ \sigma^2 g_{i-k}, & i \leq k \end{cases} \tag{3.56}$$

and defining $C_k \equiv \mathrm{E}[y_n y_{n-k}]$ with a mean of zero (according to Equation (3.21)), the autocovariance function of the AR model can be expressed as:

$$C_0 = \sum_{i=0}^{m} a_i C_{-i} + \sigma^2 \tag{3.57}$$

$$C_k = \sum_{i=0}^{m} a_i C_{k-i} \tag{3.58}$$

These equations are known as the Yule-Walker equations.

Partial Autocorrelation function We denote the AR coefficient a_i of the m-th order AR model as a_i^m. The PACF measures the effective correlation between y_n and y_{n-k} while removing the influence of correlations with intermediate observations $y_{n+1}, \cdots, y_{n+k-1}$, which corresponds to a_m^m. Typically, computing the PACF involves estimating a_i^i for $i = 1, \cdots$. However, an efficient method exists by using equations that link a_i^m and a_i^{m-1}. Specifically, this relation holds:

$$a_i^m = a_i^{m-1} - a_m^m a_{m-i}^{m-1}, \quad i = 1, \cdots, m-1. \tag{3.59}$$

Rearranging this equation yields:

$$a_i^m = a_i^{m-1} - a_m^m (a_{m-i}^m + a_m^m a_i^{m-1}) \tag{3.60}$$

$$a_i^{m-1} = \frac{a_i^m + a_m^m a_{m-i}^m}{1 - (a_m^m)^2}. \tag{3.61}$$

Using these relations, we can compute $a_1^{m-1}, \cdots a_{m-1}^{m-1}$ from $a_1^m, \cdots, a_m^m$. By repeating this process iteratively, we can calculate all PACF values, $a_1^1, \cdots, a_m^m$.

Power Spectrum The power spectrum represents the strength of fluctuations at different frequencies. Using the AR coefficients, it can be expressed as

$$p(f) = \frac{\sigma^2}{\left|1 - \sum_{j=1}^{m} a_j e^{-2\pi i j f}\right|^2} \tag{3.62}$$

In the power spectrum of the AR models, the number of peaks is determined by the AR order, and their locations are determined by the minima of $\left|1 - \sum_{j=1}^{m} a_j e^{-2\pi i j f}\right|$. The model can create peaks equal to half of the order. The fact that the AR model can generate peaks in the power spectrum means that it is possible to utilize it for periodicity search in astronomy. This allows us to examine periodic signals that may be buried in

noise, making it a valuable tool for detecting and studying periodic phenomena in time series data.

3.3.1.3 Example

In this example, synthetic data are generated using an AR model given by the equation $y_n = 0.9y_{n-1} + v_n$, where v_n represents white noise following a Gaussian distribution with a mean of 0 and a variance of 1.0 (Figure 3.4). Here, we assume that the sampling rate is 1 Hz.

The impulse response function and the autocorrelation function of this AR model exhibit exponential decay (panels (a) and (b) of Figure 3.5). However, even after a lag of 30, a significant power remains in the autocorrelation function. This is because the large coefficient $a_i = 0.9$ allows y_n to retain past information for a long period. On the other hand, the PACF shows a value of 0.9 at lag 1 and 0 for the lags greater than 1 as we modeled (panel (c) of Figure 3.5). This indicates that the prolonged correlation in the autocovariance function is not real.

In the power spectrum (Panel (d) of Figure 3.5), a flat shape is observed in the low-frequency range, which decays towards higher frequencies. This behavior, known as a "broken power law," is often seen in the power spectrum of random fluctuations such as accretion phenomena. It implies that the AR model can be utilized for these time series models. The functions, except for the impulse response function, are calculated directly from the data (shown in blue lines), but they are noisy due to the influence of the noise. In contrast, the AR model provides smooth curves, as noise is included in the modeling.

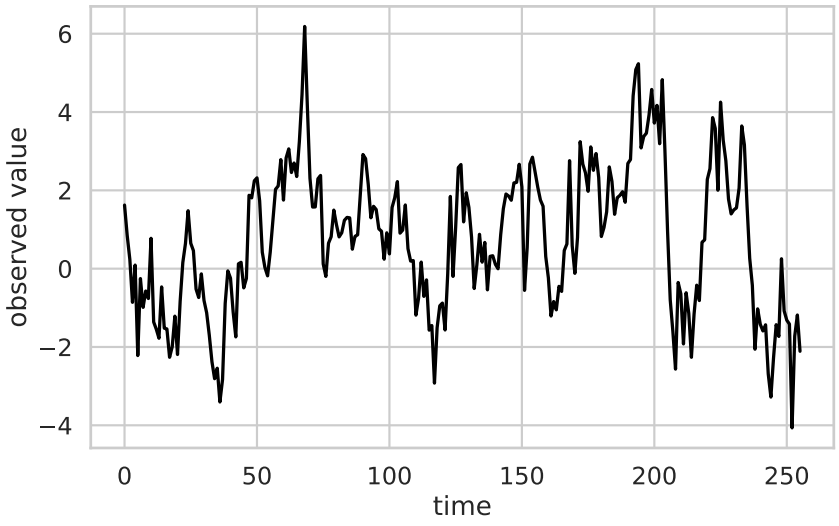


Figure 3.4: Synthetic data generated by the AR model with an order 1.

In summary, the AR model parameters allow us to calculate the description functions without being affected too much by noise. Here, we should note that this approach assumes that the noise follows a Gaussian distribution.

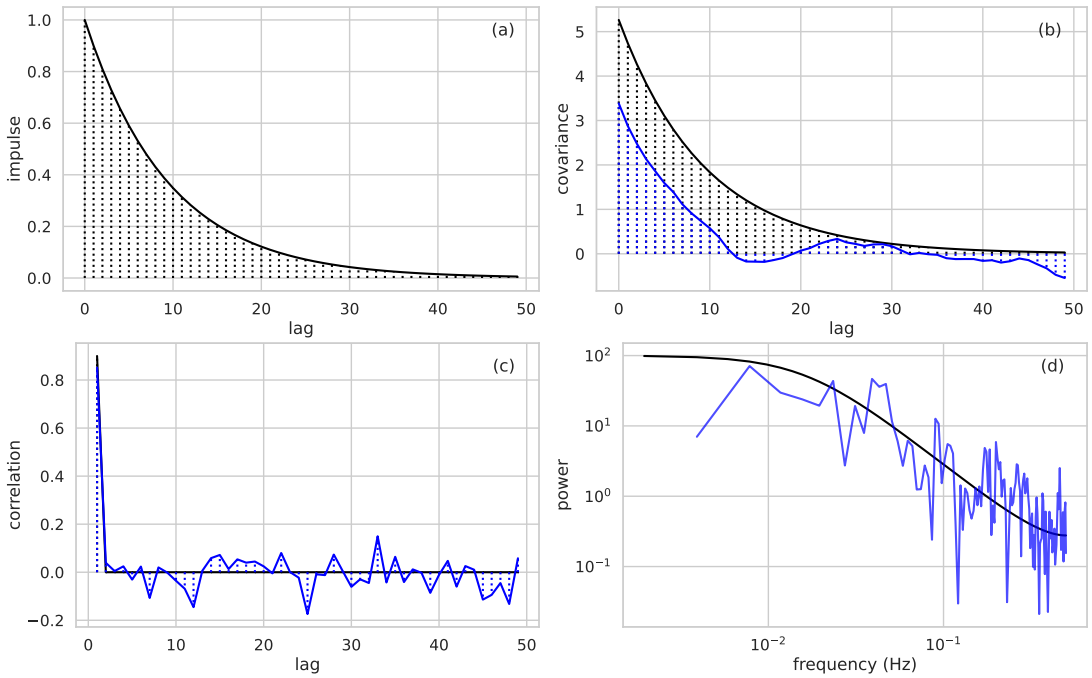


Figure 3.5: (a) Impulse response function, (b) covariance function, (c) partial-autocorrelation function, and (d) power spectrum for the synthetic data. The black curves indicate the description functions calculated from the true model parameters, while the blue curves are those calculated from the data directly.

3.3.2 Vector Autoregressive Model

3.3.2.1 Definition

The Vector Autoregressive (VAR) model is a regressive model for multivariate time series $y_n = [y_n(1), \cdots, y_n(\ell)]^T$, which utilizes previous values $y_{n-1}, \cdots, y_{n-M}$ and ℓ-dimensional white noise v_n, expressed as:

$$\begin{aligned} y_n &= A_1 y_{n-1} + \cdots + A_j y_{n-j} + v_n \\ &= \sum_j^m A_j y_{n-j} + v_n, \end{aligned} \tag{3.63}$$

where A_j is the coefficient matrix and m is the order of the VAR model.

Stationarity The characteristic equation for the VAR model is derived from the generalization of the one for the AR model (Equation (3.49)) as:

$$\left| I - \left(\sum_{j=1}^m A_j z^j \right) \right| = 0, \tag{3.64}$$

where I represents the identity matrix with the same dimensions as A_j. Similar to the AR model, the stationary condition for the VAR model requires that all characteristic

roots exceed 1, i.e., $|z| > 1$. For example, the characteristic equation of a first-order VAR model:

$$|I - A_1 z| = 0 \tag{3.65}$$

This can be reformulated as:

$$|z^{-1} I - A_1| = 0 \tag{3.66}$$

Essentially, this equation aligns with the eigen equation for A_1, indicating that the stationary condition for a first-order VAR model is met when the absolute values of all eigenvalues of A_1 are less than 1.

3.3.2.2 Description Functions

The cross-covariance of the VAR model is calculated as

$$\begin{aligned} C_0 &= \sum_{j=1}^{M} A_j C_{-j} + W \\ C_k &= \sum_{j=1}^{M} A_j C_{k-j}, \quad k = 1, 2, \cdots, \end{aligned} \tag{3.67}$$

which is called the Yule-Walker equation in the same manner as the AR model. The cross-correlation function can be calculated using Equation (3.35).

The cross-spectrum can be calculated as:

$$P(f) = A(f)^{-1} W (A(f)^{-1})^* \tag{3.68}$$

Here, $A(f)$ is the matrix whose element (j, k) is expressed as:

$$A_{jk} = \sum_{m=0}^{M} a_m(j, k) e^{-2\pi i m f}, \tag{3.69}$$

where

$$a_0(j, k) = \begin{cases} -1 & (j = k) \\ 0 & (j \neq k) \end{cases} \tag{3.70}$$

Thus, p_{ii} corresponds to the power spectrum. The amplitude spectra, phase spectra, and coherence are calculated using Equations (3.45) and (3.46) by the cross-spectra $P(f)$.

We define $A(f)^{-1}$ as $B(f) = (b_{jk}(f))$. When the variance-covariance matrix is the diagonal matrix $W = \mathrm{diag}(\sigma_1, \cdots, \sigma_\ell)$, the i-th power spectrum can be transformed as:

$$\begin{aligned} p_{ii}(f) &= \sum_{j=1}^{\ell} b_{ij}(f)\sigma_j^2 b_{ij}(f)^* \\ &\equiv \sum_{j=1}^{\ell} |b_{ij}(f)|^2 \sigma_j^2 \end{aligned} \tag{3.71}$$

This means that the i-th power spectrum can be decomposed into the ℓ noise effects represented as $|b_{ij}(f)|^2\sigma_j^2$, $(j = 1, \cdots, \ell)$. Therefore, by dividing Equation (3.71) by $p_{ii}(f)$, we obtain:

$$r_{ij}(f) = \frac{|b_{ij}(f)|^2\sigma_j^2}{p_{ij}(f)}. \tag{3.72}$$

$r_{ij}(f)$ represents the ratio of the variation originating from $v_n(j)$ to the variation of $y_n(i)$ at frequency f and is thus called the relative power contribution.

3.3.2.3 Example

Let's examine the description functions using synthetic data, which is generated from the VAR model with an order of 2. The sampling rate of the synthetic data is assumed to be 1 Hz. We define the model with coefficient matrices of

$$\begin{aligned} A_1 &= \begin{bmatrix} a_{1,11} & a_{1,12} \\ a_{1,21} & a_{1,22} \end{bmatrix} = \begin{bmatrix} 0.7 & 0.2 \\ -0.1 & 0.3 \end{bmatrix}, \\ A_2 &= \begin{bmatrix} a_{2,11} & a_{2,12} \\ a_{2,21} & a_{2,22} \end{bmatrix} = \begin{bmatrix} -0.3 & 0.1 \\ 0.1 & 0.2 \end{bmatrix}, \end{aligned} \tag{3.73}$$

and the variance-covariance matrix of

$$W = \begin{bmatrix} 0.3 & 0.0 \\ 0.0 & 0.2 \end{bmatrix}. \tag{3.74}$$

The simulated multivariate time series data are shown in Figure 3.6. While referring to the VAR coefficient matrix, let's briefly touch upon the characteristics of the time series generated by the model. Examining the autoregressive coefficients for $y_n(1)$ from Equation (3.73) ($a_{1,11}$, $a_{2,11}$), we observe that one lag in the past provides positive feedback, while two lags in the past provide negative feedback. As a result, a somewhat periodic signal is generated. In fact, $y_n(1)$ exhibits a somewhat periodic variation with a period of around 10 samples, which will also be confirmed as peaks in the power spectrum later. On the other hand, the autoregressive coefficients for $y_n(2)$ ($a_{1,22}$, $a_{2,22}$) are both positive, indicating that this variable tends to retain the memory of past values. However, in this model, since the values of these coefficients have values similar to the regression coefficients of $y_n(1)$ ($a_{1,21}$, $a_{2,21}$), the realization of $y_n(2)$ exhibits similar

variations to $y_n(1)$. It is possible to interpret some characteristics of the observations with autoregressive coefficients of the VAR model, but this is not always straightforward as we need to consider contributions from other series. We need to use the description functions below to facilitate an easier and more quantitative understanding of these features.

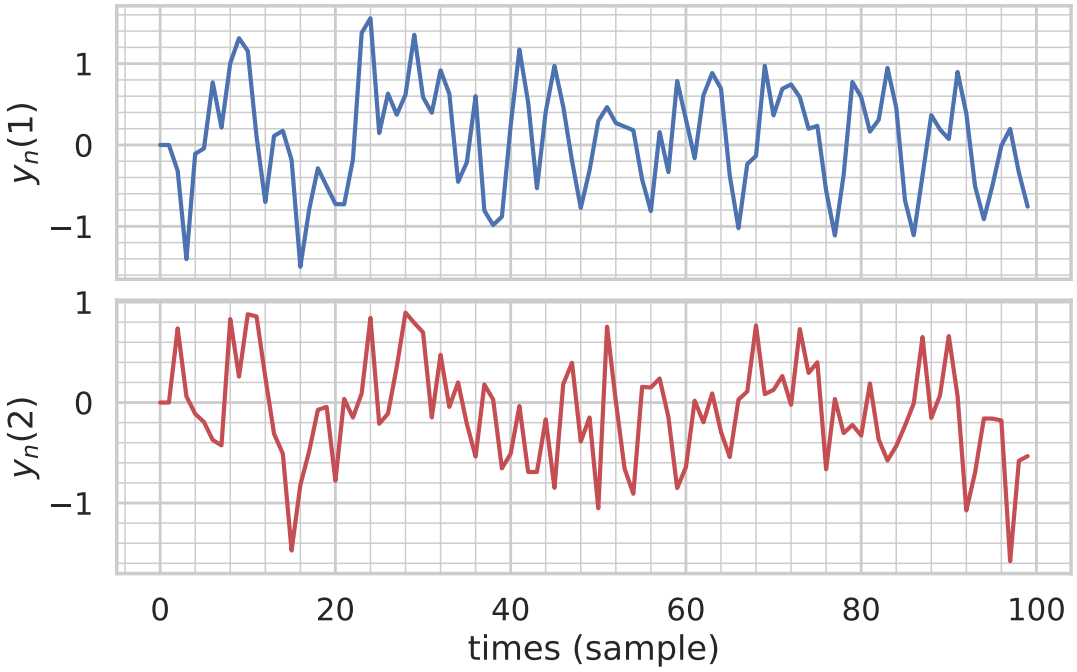


Figure 3.6: Generated sample curves from the VAR model.

The cross-correlation function is shown in Figure 3.7, and the amplitude spectrum, phase spectrum, coherence, and power decomposition are shown in Figures 3.8, 3.9 and 3.10. The functions are calculated based on the VAR coefficients. In the power spectrum, which is shown in the diagonal panels of Figure 3.8 and Figure 3.9, it is evident that $y_n(1)$ exhibits prominent power around 0.1 Hz, as previously mentioned.

Let's examine the cross-correlation functions. Compared to $\mathrm{Cor}(y_n(2), y_{n-k}(2))$, $\mathrm{Cor}(y_n(1), y_{n-k}(1))$ exhibits a more rapid decay and shows weak oscillations attributable to periodic signals. These oscillations are not observed in the autocorrelation of $y_n(2)$, reflecting a smaller influence of the periodic signal upon it. The cross-correlation function $\mathrm{Cor}(y_n(1), y_{n-k}(2))$ displays a peak in correlation at a lag of 2 samples, indicating the detection of a time lag signal. This can be attributed to the fact that both $a_{1,12}$ and $a_{2,12}$ in Equation (3.73) have positive values, which contribute to the presence of this time lag signal. On the other hand, $\mathrm{Cor}(y_n(2), y_{n-k}(1))$ shows a small peak created by this peak and periodic signals, but the correlation is close to zero.

The amplitude spectrum shows a decrease in amplitude as we move to higher frequencies. However, there is a local increase at around 0.1 Hz, corresponding to the increase in power of the periodic signal. In the phase spectrum, we observe significant phase shifts at higher frequencies, which is likely due to the time lag indicated by $\mathrm{Cor}(y_n(1), y_{n-k}(2))$ in Figure 3.7. The coherence also exhibits a pattern similar to that of the amplitude

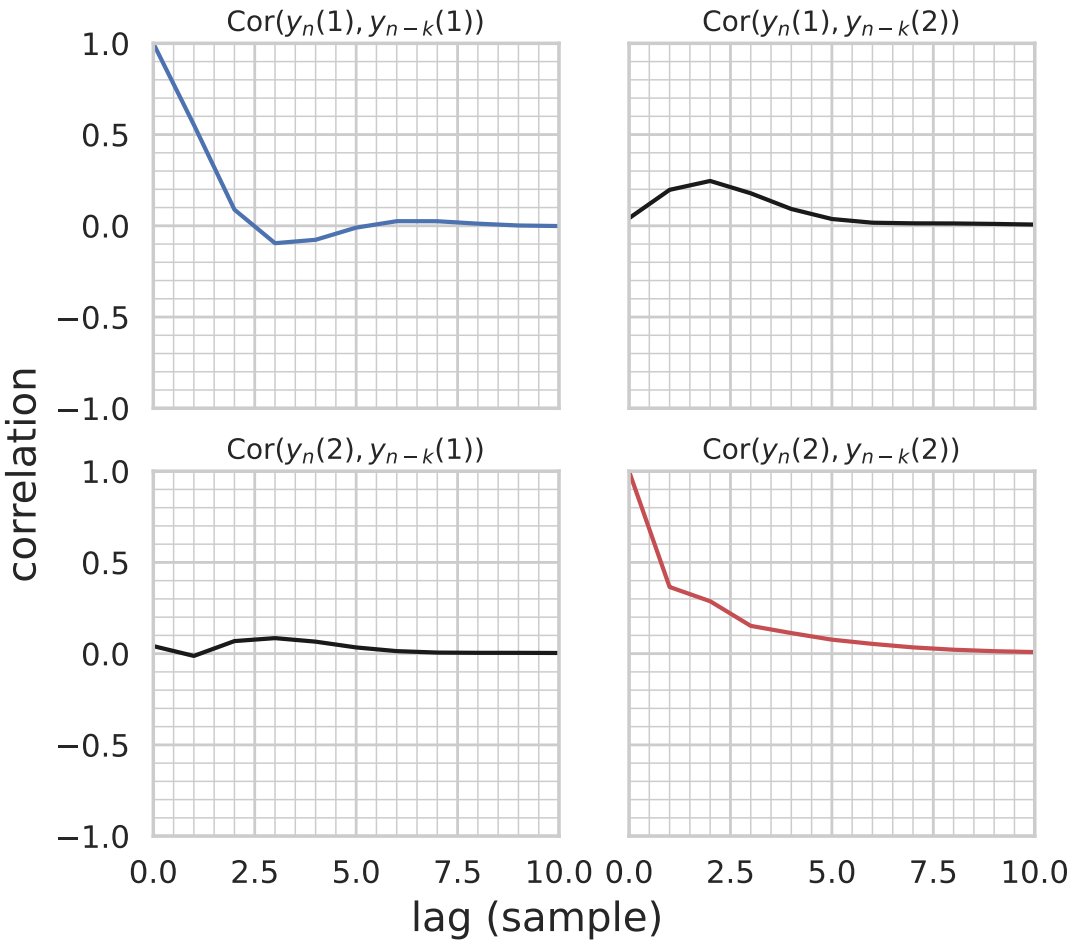


Figure 3.7: Cross-correlation function calculated from the VAR coefficient matrix. The diagonal panels are the autocorrelation functions.

spectrum, but the maximum value is around 0.2, indicating that the realizations do not show a similar behavior.

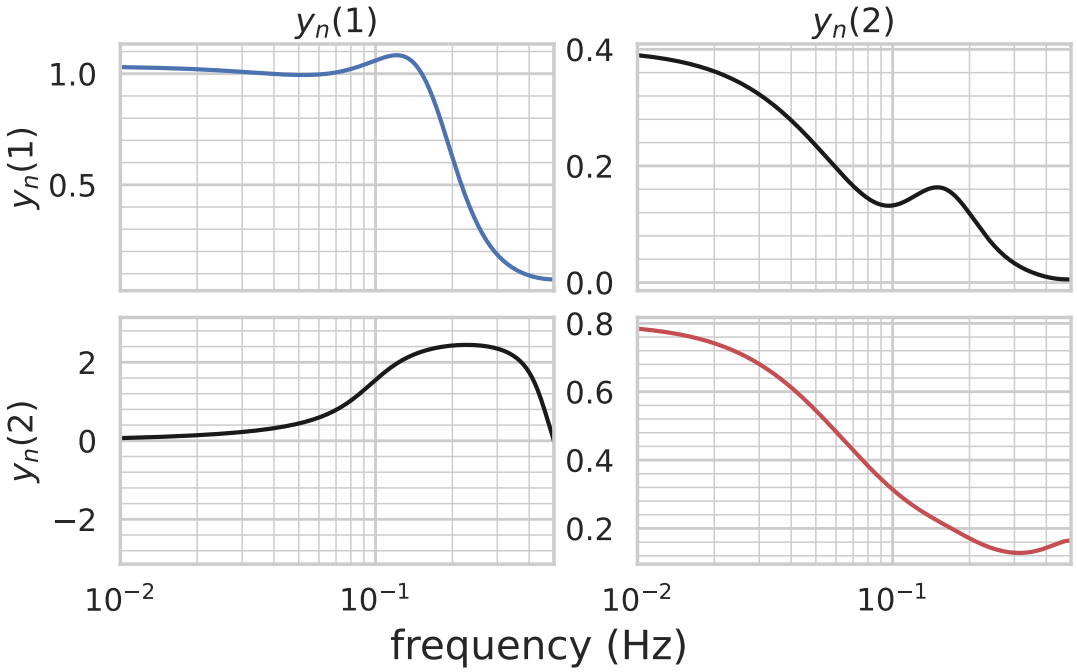


Figure 3.8: Amplitude and phase spectrum calculated from the VAR coefficient matrix. The power spectra are shown in the diagonal panels. The panel above the diagonal is the amplitude spectrum and the one below the diagonal is the phase spectrum.

Using these functions, we can examine the power contributions and relative power contributions (Equation (3.72)). For $y_n(1)$, we can see that most of the variations can be explained by $y_n(1)$, and $y_n(2)$ has some contribution, particularly at low frequencies. Similarly, for $y_n(2)$, most of the variations are attributed to $y_n(2)$, while $y_n(1)$ has a minor contribution. In the frequency band where $y_n(1)$ exhibits a periodic signal, $y_n(1)$ contributes to $y_n(2)$ although it is relatively small.

These functions reveal that $y_n(1)$ and $y_n(2)$ exhibit largely independent variations, the periodic signal originating from $y_n(1)$ has a weak correlation, and there is a weak time lag from $y_n(2)$ to $y_n(1)$. This example shows the potential of the VAR model for understanding the relationships among different components of a system.

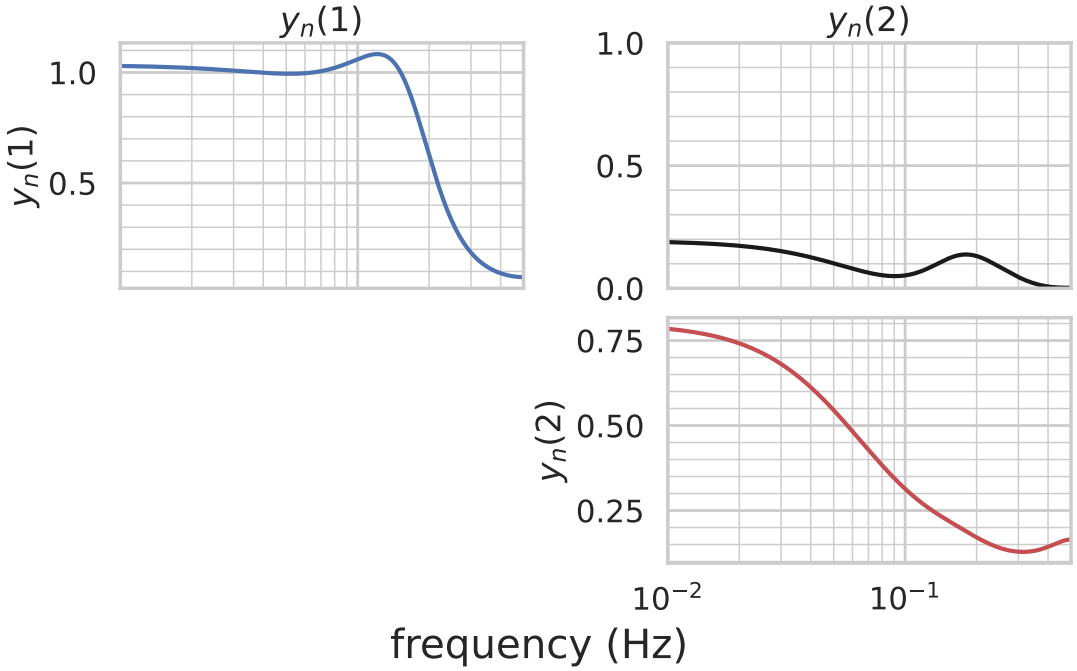


Figure 3.9: Coherence calculated from the VAR coefficient matrix. The power spectra are shown in the diagonal panels.

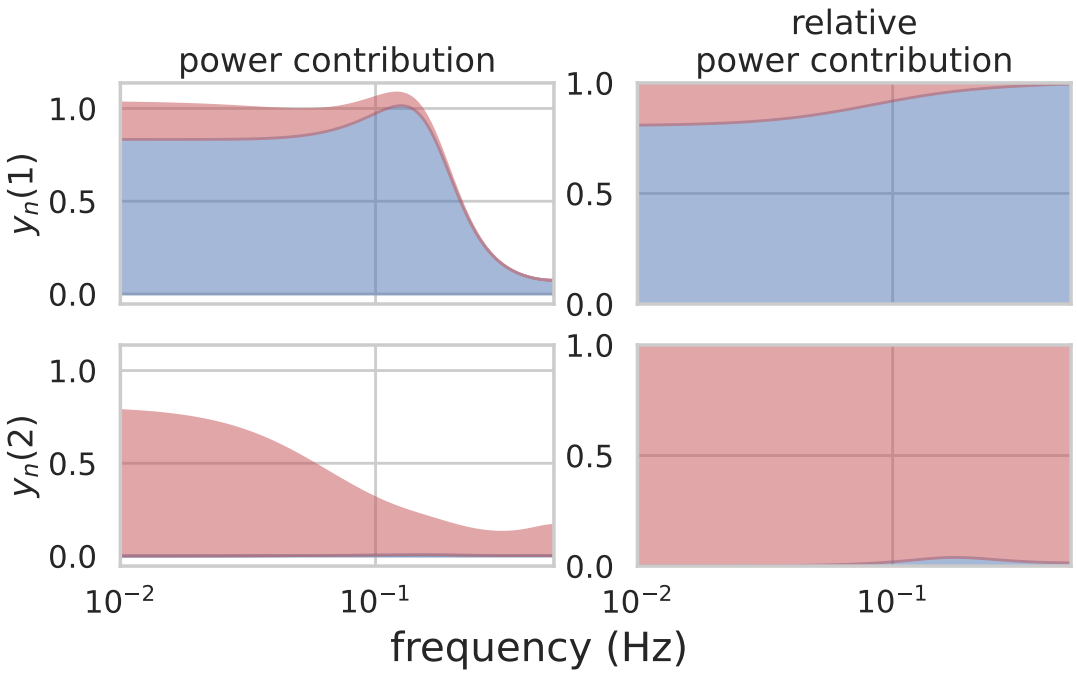


Figure 3.10: Power contribution and relative power contribution calculated from the VAR coefficient matrix. True decomposed power spectra are shown on the left panels. The blue and red areas indicate the contribution of $y_n(1)$ and $y_n(2)$, respectively. Their ratios are shown on the right panels

.

3.4 State-Space Model

The state-space model is a time series model that represents an observed data using two equations: the system equation and the observation equation. The system equation models the transition process of latent variables, while the observation equation models the observation process of these variables. The algorithm for likelihood estimation of the model differs depending on whether the equations are linear or nonlinear, and whether all probability distributions follow Gaussian or not. When the equations are linear and the probability distributions are Gaussian, it is referred to as the Linear Gaussian State-Space Model (LGSSM), which is most often used in time series modeling. This is because it allows efficient filtering, prediction, and likelihood calculation using the Kalman filter algorithm. In this chapter, we explain the concepts separately for linear and nonlinear state-space models.

3.4.1 Linear Gaussian State-Space Model

We start with the Linear Gaussian State-Space Model (LGSSM), where the equations are linear, and the variables follow Gaussian distributions. Let x_n be a k-dimensional vector, which is called the state (or latent) variable, and y_n represent a ℓ-dimensional vector at time n, which is called the observation variable. The LGSSM is expressed as follows:

$$\begin{aligned} x_n &= Fx_{n-1} + Gv_n \\ y_n &= Hx_n + w_n \end{aligned} \tag{3.75}$$

Here, v_n represents the system noise, which is an m-dimensional vector of the Gaussian white noise with a mean of 0 and a covariance matrix of Q_n. w_n is the observation noise, which is an ℓ-dimensional vector. The matrices F, G, and H have dimensions $k \times k$, $k \times m$, and $\ell \times k$, respectively. F and H are called a transition matrix and an observation matrix.

3.4.1.1 Kalman Filter

All variables in LGSSM become Gaussian distributions because the noise follows Gaussian and the equations are linear. Using the Kalman filter algorithm, it is possible to perform sequential state estimation and prediction, along with simultaneous likelihood estimation. The Kalman filter estimates the simultaneous joint distribution of all observed data and states by repeating filtering and prediction for each observation. The process of estimating the state is called filtering. This corresponds to estimating the conditional probability distribution $p(x_n|y_n)$. Prediction involves determining the state one time step ahead, x_{n+1}, given the observation y_n at time n, which is essentially estimating $p(x_{n+1}|y_n)$.

We give an overview of the algorithm for estimating the probability. Hereafter, the conditional expectation of a certain variable X is denoted as:

$$X_{n|j} \equiv \mathrm{E}[X_n|y_j]. \tag{3.76}$$

Let $x_{n|j}$ and $V_{n|j}$ represent the conditional mean and conditional covariance matrix of the state when the observation y_j is given. The one-step-ahead prediction of these variables can be calculated as follows:

$$x_{n|n-1} = F_n x_{n-1|n-1} \tag{3.77}$$
$$V_{n|n-1} = F_n V_{n|n-1} F_n^T + G_n Q_n G_n^T. \tag{3.78}$$

In the filtering step, the following calculations are performed to estimate the probability distribution of the state at time n when an observation is obtained:

$$K_n = V_{n|n-1} H_n^T (H_n V_{n|n-1} H_n^T + R_n)^{-1} \tag{3.79}$$
$$x_{n|n} = x_{n|n-1} + K_n(y_n - H_n x_{n|n-1}) \tag{3.80}$$
$$V_{n|n} = (I - K_n H_n) V_{n|n-1} \tag{3.81}$$

Here, K_n is called the Kalman gain, which serves as a coefficient to adjust the discrepancy between prediction and observation. It plays the role of weighting when correcting the prediction based on the observation.

Smoothing is the process of estimating the past states when all observations $Y_n = \{y_1, \ldots, y_N\}$ are obtained. Since the estimation is performed using all observations, the variability is reduced, and the errors are minimized. This is why it is called smoothing. In this process, the results of the Kalman filter are used, and the following calculations are performed:

$$A_n = V_{t|t} F_{t+1}^T V_{t+1|t}^{-1} \tag{3.82}$$
$$x_{n|N} = x_{n|n} + A_n(x_{n+1|n} - x_{n+1|n}) \tag{3.83}$$
$$V_{n|N} = V_{n|n} + A_n(V_{n+1|N} - V_{n+1|n}) A_n^T. \tag{3.84}$$

From these equations, it can be observed that smoothing involves estimating $x_{N|N}$ and $V_{N|N}$ using the Kalman filter for the final observation and then using these estimates to infer $x_{N-1|N}$ and $V_{N-1|N}$. This process is repeated in reverse chronological order from time N to 1, thus smoothing.

Let's see an example of filtered and smoothed states with synthetic data (Figure 3.11). The filtered state exhibits more fluctuations with a wide confidence interval than the smoothed state. This confirms that obtaining all observation data improves the accuracy of estimating past states.

In the Kalman filter algorithm, it is possible to perform a prediction even when observations are missing. This process can be considered as missing value interpolation when intermediate observations are absent, and as long-term forecasting when predictions are

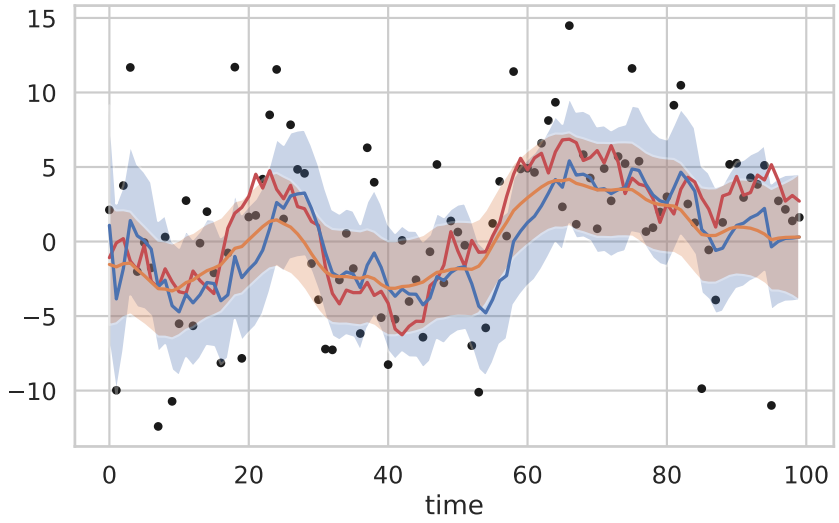


Figure 3.11: Example of synthetic data and its filtering and smoothing. Synthetic data are generated using a random walk with system noise and observation noise variances of 1 and 5, respectively. In the state-space representation of the random walk, we used $F = [1]$, $G = [1]$, and $H = [1]$ in Equation (3.75). The observed data are represented by the black points. The true state time series is denoted by the red curve. The 95% interval of the distributions for filtering and smoothing states are shown by the blue and orange areas, respectively.

iteratively made for future time points. Observations of missing data are quite common in astronomy. The state estimation for LGSSM with the Kalman filter is well suited due to its ability to handle missing values and perform long-term predictions in a natural way.

3.4.1.2 State-Space Form of AR model

Up until now, we have discussed state estimation in the general form of LGSSM. Now, let's consider a specific example of the state-space representation of an AR model. By extending this representation, we can obtain the state-space representation of a VAR model. In this research, we often assume that the system equation follows an AR or VAR model for its flexibility. The state and likelihood calculations are computed using the representation introduced here.

Consider the state-space representation of an AR model. By defining F, G, H, and x_n as follows:

$$F = \begin{bmatrix} a_1 & a_2 & \cdots & a_k \\ 1 & & & \\ & \ddots & & \\ & & 1 & 0 \end{bmatrix}, \; G = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \; H = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}, \; x_n = \begin{bmatrix} y_{n-1} \\ y_{n-2} \\ \vdots \\ y_{n-m-1} \end{bmatrix}. \tag{3.85}$$

We can confirm that they become equivalent equations to the original AR model. There are efficient methods to approximately estimate the maximum likelihood values of the AR

model, which require less computational effort compared to state-space models. Thus, the LGSSM of the AR model should be used when precise likelihood estimation is needed or tasks such as missing value interpolation and long-term forecasting are required. In the former cases, it is important to use appropriate parameters as the initial values for the Kalman filtering.

The state-space model representation of the VAR model can be obtained by replacing the coefficients a_i of the AR model with the matrices A_i as

$$F = \begin{bmatrix} A_1 & A_2 & \cdots & A_k \\ I & & & \\ & \ddots & & \\ & & I & 0 \end{bmatrix}, \; G = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \; H = \begin{bmatrix} 1 \cdots 1 \; 0 \cdots 0 \end{bmatrix} \; x_n = \begin{bmatrix} y_{n-1} \\ y_{n-2} \\ \vdots \\ y_{n-m-1} \end{bmatrix}, \tag{3.86}$$

where I is the identity matrix. It should be noted that y_n is the vector variable. An efficient method exists to approximately estimate the maximum likelihood for the VAR model, which can be used for deriving the initial values for Kalman filtering when needed, similar to the AR model.

3.4.1.3 Dynamic Factor Model

The dynamic factor model is a model that represents observations using fewer latent variables than the number of observed variables. It can be interpreted as a time series version of factor analysis. As we show later in this thesis, this modeling approach is highly relevant in astronomy. In astronomy, flux variation, or light curve, is measured at many different energy bands. They form a multivariate time series in the form of vectors. They are the superposition of the flux of different physical spectral components. If the number of observed energy bands is larger than the number of the spectral components, the dynamic factor model can be utilized. By modeling this way, the states correspond to the light curves of individual spectral components.

Similar to the factor analysis, the dynamic factor model suffers from the issue of rotational indeterminacy, which requires processing in a context somewhat closer to the factor analysis than the state-space models. In the factor analysis, the observation model in the state-space model is a model of how observations are changed by states over time. When the assumption is made that the probability variables are non-time-varying, it is expressed as a factor analysis model:

$$\begin{bmatrix} y(1) \\ y(2) \\ y(3) \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \begin{bmatrix} x(1) \\ x(2) \\ x(3) \end{bmatrix} + \begin{bmatrix} w(1) \\ w(2) \\ w(3) \end{bmatrix} \tag{3.87}$$

This is essentially the equation for factor analysis. The transformation by an arbitrary regular matrix corresponds to the rotation of the factor axes, and the same rotation

methods used in factor analysis can be applied here.

In the Gaussian linear state-space model, there is also flexibility in scaling. This arises from the assumption that both the observation and the system models are affected by Gaussian noise with scaling-free covariance. In practice, we need to impose constraints to ensure that the model remains identifiable.

Similar to the factor analysis, the dynamic factor model has rotational indeterminacy. By multiplying with a regular M from left to the system, we can perform the following transformation:

$$Mx_n = MFx_{n-1} + Mv_n \tag{3.88}$$
$$Mx_n = MFM^{-1}Hx_{n-1} + Mv_n \tag{3.89}$$
$$x'_n = M'x'_{n-1} + v'_n \tag{3.90}$$

where $x'_n = Mx_n$, $w'_n = Mw_n$. Also, by multiplying M by the observation equation, we obtain

$$y_n = HM^{-1}Mx_n + w_n \tag{3.91}$$
$$y_n = H'x'_n + w_n \tag{3.92}$$

where $A' = HM^{-1}$. These transformations show that the equivalent LGSSM can be defined using M as the original. In practice, in order to reduce the degrees of freedom due to rotational indeterminacy, one may assume, for example, a lower triangular matrix with diagonal elements of 1 for H during estimation. After estimation, interpretations can be made by introducing any arbitrary rotation matrices.

3.4.2 Other State-Space Models

We use the linear Gaussian models for the state-space model in this thesis, but for the sake of completeness and future outlook, we briefly touch upon more generalized state-space models.

Linear Non-Gaussian State-Space Models We have outlined state estimation using the Kalman filter algorithm, assuming that all probability distributions follow Gaussian distributions. However, it is possible to consider linear state-space models where the probability variables do not follow Gaussian distributions. In such cases, the model is referred to as a linear non-Gaussian state-space Model. Although the Kalman filter is no longer applicable to this model, an alternative algorithm called the "Extended Kalman filter" has been proposed, which estimates the approximated distributions. However, it is used in limited cases, including low-dimensional variations.

Non-Linear Non-Gaussian State-Space Models The non-linear non-Gaussian state-space model for the time series y_n is expressed as (Kitagawa and Gersch, 1996):

$$\begin{aligned} x_n &= F(x_{n-1}, v_n) \\ y_n &= H(x_{n-1}, w_n). \end{aligned} \tag{3.93}$$

Here, x_n and y_n represent the state and observed variables, respectively. v_n and w_n are white noise following density functions $q(v)$ and $r(w)$, respectively. This formulation encompasses the linear Gaussian state-space model, as defined in Equation (3.75), as a special case. In the linear Gaussian case, $F(x_{n-1}, v_n) = Fx_{n-1} + Gv_n$ and $H(x_{n-1}, w_n) = Hx_n + w_n$.

Generarized State-Space Model A more general form of the state-space model is known as the generalized state-space model, defined by the following equations (Kitagawa and Gersch, 1996):

$$\begin{aligned} x_n &\sim Q(\,\cdot\,|x_{n-1}) \\ y_n &\sim R(\,\cdot\,|x_n). \end{aligned} \tag{3.94}$$

In these equations, Q and R represent the conditional distribution functions for the states x_{n-1} and x_n, respectively. This form becomes particularly valuable when dealing with discrete states or when observed values follow a binomial or Poisson distribution.

As an illustration, let's examine a scenario where observed values are obtained as counts and need to be modeled using the Poisson distribution. One common model takes the following structure:

$$\begin{aligned} x_n &= Fx_{n-1} + Gv_n, \quad v_n \sim \mathcal{N}(0, Q) \\ \alpha_n &= Hx_n \\ y_n &\sim \exp\left(\alpha_n' y_n - b(\alpha) + c(y_n)\right), \end{aligned} \tag{3.95}$$

Here, F, G, and H denote arbitrary matrices, and Q is the covariance matrix. The functions $b(\cdot)$ and $c(\cdot)$ are arbitrary functions. The distribution of y_n follows the general form of the exponential family, encompassing the Poisson distribution and multinomial distribution (e.g. West and J. Harrison, 1989).

Chapter 4

X-ray Variability of BHBs

Contents

4.1 Overview of BHBs

4.1.1 Observational Features

Four major observational features distinguish BHBs from other celestial sources: their transient nature, the X-ray spectrum, short-term variability, and the hardness-intensity ratio. These characteristics of BHBs are summarized in Remillard and McClintock (2006). We briefly review each of them below.

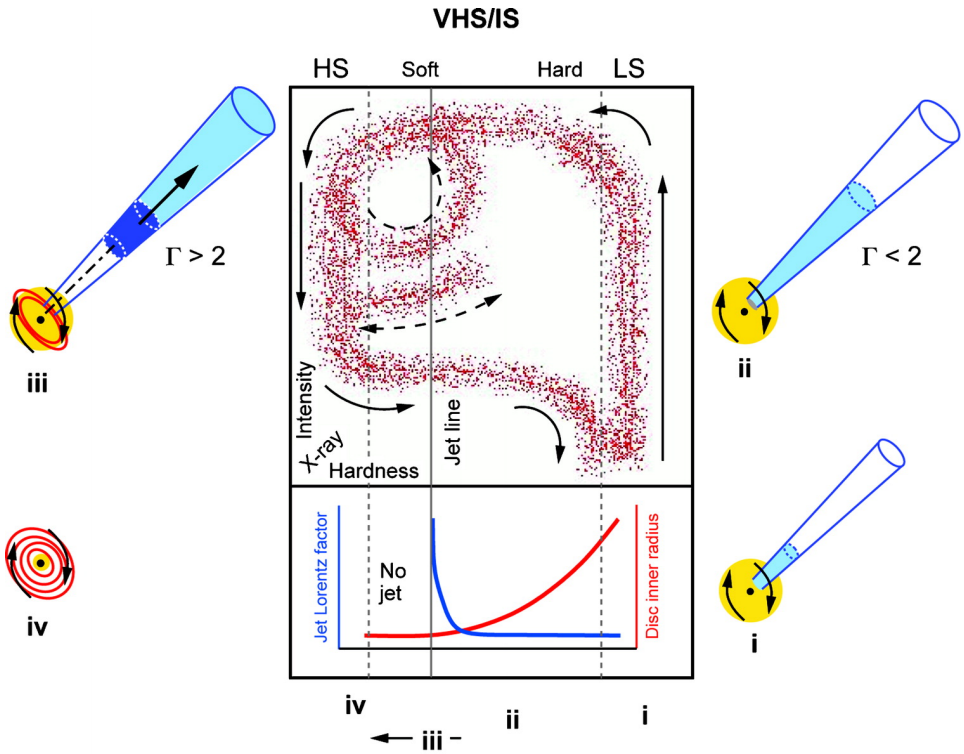
Transient nature BHBs are distinctively classified into two classes based on the mass of the companion star: high-mass BHBs when the companion star is larger than $\approx 3M_{\odot}$ and low-mass BHBs when it is smaller than $\approx 3M_{\odot}$.

In high-mass BHBs, there is a constant mass loss from the companion star via stellar winds, a part of which is captured by the gravitational potential of the black hole. This results in steady mass accretion. In low-mass BHBs, matter from the surface of the companion star filling the Roche lobe accretes through a Lagrangian point. The mass accretion rate changes and the state of the accretion disk changes. The state changes abruptly, and the X-ray luminosity increases suddenly. In this manner, low-mass BHBs are often discovered as a transient X-ray source. Most BHBs are of the low-mass type, and those of the high-mass BHBs are rare (e.g., Cygnus X-1, LMC X-1, and LMC X-3). The subject of our study, MAXI J1820+070 falls in the category of low-mass BHBs.

X-ray spectra The X-ray spectra of BHBs consist of two major components: a thermal component described by a multi-temperature blackbody and a non-thermal component represented by a power-law distribution. The non-thermal component usually extends to high energies, sometimes with and without a cutoff energy. The characteristic temperature of the thermal component is ∼1 keV and it does not typically extend to higher energies. Some BHBs also show the Fe Kα emission line in the 6–7 keV range. The relative strength of these components changes depending on the states of the BHBs.

Power spectrum Short-term flux variability is also a characteristic feature of BHBs. The variability exhibits a unique pattern in the power spectra. Many BHBs are known to have significant power in a specific frequency range known as the quasi-periodic oscillations (QPOs), typically falling between 0.01–450 Hz. These patterns are analyzed using a Lorentzian (Cauchy) distribution, with the coherence parameter defined as $Q = \nu/\mathrm{FWHM}$, where ν is the frequency and FWHM is the full width at the half maximum (Remillard and McClintock, 2006). Empirically, there is a known relationship between the coherence parameter and the state of the BHB. QPOs are classified into three types; Type A, B, and C, based on the presence of shifts and the leading energy band (van der Klis, 2006; Remillard and McClintock, 2006).

Hardness Intensity Diagram The Hardness Intensity Diagram (HID) is a plot that shows the flux intensity and the flux ratio of two bands called the hardness ratio (HR).



Remillard RA, McClintock JE. 2006.
Annu. Rev. Astron. Astrophys. 44:49–92

Figure 4.1: Schematics of hardness intensity diagram of BHBs (Figure taken from Remillard and McClintock, 2006)

This diagram is useful for tracking state changes in a system. For BHBs, the HID follows a distinctive cyclic pattern, as shown in Figure 4.1 (Remillard and McClintock, 2006). The relative positions of the hard and soft states are generally well-known in BHBs, with a higher HR indicating the hard state and a lower HR indicating the soft state. The typical evolution process in the HID transition for BHBs can be summarized as follows: immediately after an outburst, BHBs exhibit characteristics of the hard state. During this phase, the HR remains relatively constant and the intensity sharply increases, causing the HID to move almost vertically upward from the bottom right. After reaching the peak of the outburst, there is a gradual decrease in HR, resulting in a shift to the left. Once in the soft state, the intensity gradually decreases, leading to a nearly vertical downward shift from the top left. However, as the decay progresses, BHBs may return from the soft state to the hard state, resulting in an increase in HR and a rightward shift. The intensity continues to decrease, eventually returning to the quiescent state. Although there may be subtle differences depending on the energy bands, BHBs are generally known to undergo such cycles (Remillard and McClintock, 2006).

4.1.2 Definition of States

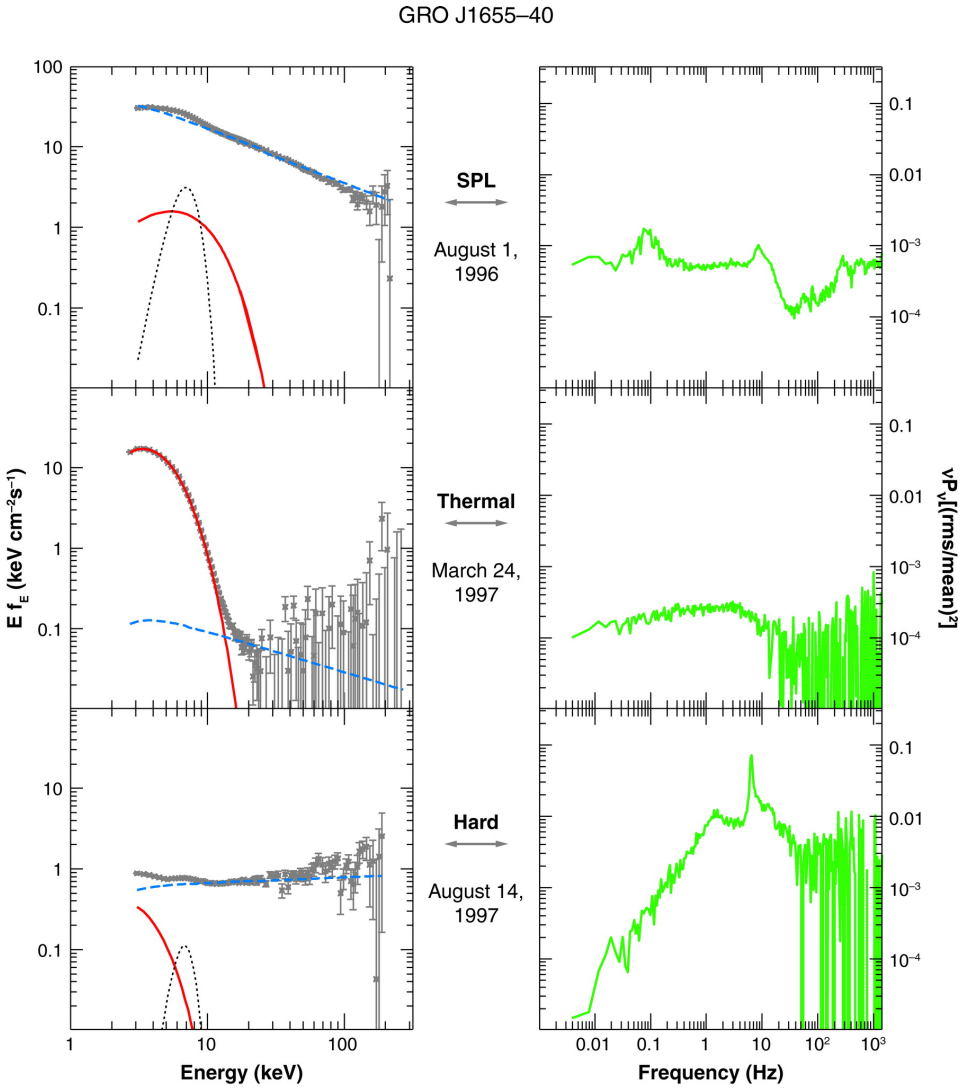
BHBs undergo abrupt changes in observational characteristics. A set of characteristics defines a state. Two major states are the low hard state and the high soft state. These

terms were coined based on the flux and spectral hardness, but some other features accompany them as described below. There are several additional states.

Soft State In the soft (thermal; Remillard and McClintock, 2006) state, the energy spectrum is characterized by a dominance of radiation in the soft X-ray band, as the name suggests. The spectrum is primarily composed of a thermal component from a multi-temperature blackbody from the accretion disk, with a presence of a weak non-thermal component. In Figure 4.2, the soft (thermal) state of low-mass BHB GRO J1655 − 40 is illustrated in the middle section. Up to 10 keV, the accretion disk component dominates, and beyond that, radiation from a weak non-thermal component becomes visible. The power spectrum in the soft state does not show particularly distinctive features, and QPOs may be either absent or very weak (see the middle left panel).

Hard State In the hard state, the energy spectrum stands out for its substantial flux extending to high energies, which is in sharp contrast to the soft state. The majority of the flux is in the form of a power-law distribution (with an index of ~ 1.7), and a weak disk component is introduced to explain the subdominant emission in the soft energy band. The power spectrum displays a frequency break, indicating a characteristic time scale of variations, and QPOs are frequently observed. In Figure 4.2, the energy spectrum and the power spectrum of GRO J1655 − 40 in the hard state are depicted in the lower section. The energy spectrum is modeled with a non-thermal component, a weak thermal component, and a Fe emission line. The power spectrum shows a break at 1 Hz, representing a typical time scale of 1 s, while a dominant QPO is evident around 5.0 Hz.

Other States In addition to the two states above, BHBs showcase several other states. These encompass a notably dim quiescence state, intermediate state during the transition between hard and soft states, the very high state featuring similar flux from both disk and non-thermal components, and the steep power-law state characterized by a significant increase in the power-law index of the energy spectrum. For more in-depth information on these states, refer to Remillard and McClintock (2006).



Remillard RA, McClintock JE. 2006.
Annu. Rev. Astron. Astrophys. 44:49–92

Figure 4.2: Energy spectra and power spectra of GRO J1655 − 40 in different states. (Figure taken from (Remillard and McClintock, 2006))

4.2 Time Lags

Among various observational tools, the time lags between two spectral components are keys to understanding the accretion geometry of BHBs. We summarize some of the findings from the past (§ 4.2.1), point out the inherent difficulties of time lag analysis in traditional light curve analysis (§ 4.2.2), and argue for the need to introduce the latent variable (§ 4.2.3).

4.2.1 Observed Lags

4.2.1.1 Hard Lag

The hard lag, observed when signals in the hard band lag behind those in the soft band, is a common feature in the hard state of most BHBs (Uttley et al., 2014). These signals have been studied using both cross-correlation functions and phase-lag spectra, with timescales typically ranging from subseconds to seconds. While the presence of hard lags was initially suggested by the asymmetry in cross-correlation functions between the hard and soft bands (e.g., Brinkman et al., 1974), evaluating them was challenging due to the autocorrelation function of a dominant component in both bands.

Subsequently, quantitative estimation became possible by analyzing frequency-specific phase-lag spectra, a method introduced by studies such as van der Klis et al. (1987) and Miyamoto et al. (1988). Consequently, phase-lag spectra became a primary tool for the time lag analysis. However, given multiple components contributing to the observed flux in an energy band, careful consideration is necessary to determine whether the values in the phase-lag spectrum accurately represent the lag amplitude of the desired signal.

As for the physical origins of the hard lag signal, it is too long for a light propagation time across a BHB system as the lag amplitudes range from submilliseconds to a few seconds. It is more likely associated with the time for matters to move within the accretion disk (Kotov et al., 2001; Arévalo and Uttley, 2006). The observed variations in the amplitude of the hard lag signal hint at some temporal changes in the structure of the accretion disk.

4.2.1.2 Soft Lag

The soft lag is a time lag signal where the soft band emission lags behind the hard band emission. The pioneering work of Uttley et al. (2011) first reported a soft lag in BHBs. They investigated delays in different energy bands using the phase-lag spectrum of cross-spectra, referred to as the phase-lag energy spectrum. The findings indicated that the soft band consistently lags behind other bands on a millisecond scale (Figure 4.3). Soft lags are also recognized as common characteristics in the hard states of BHBs

In the work by Uttley et al. (2011), they associated the soft lag signal with thermal reverberation due to Comptonization, a process that heats the disk. The lag amplitude was explained as the light-crossing time.

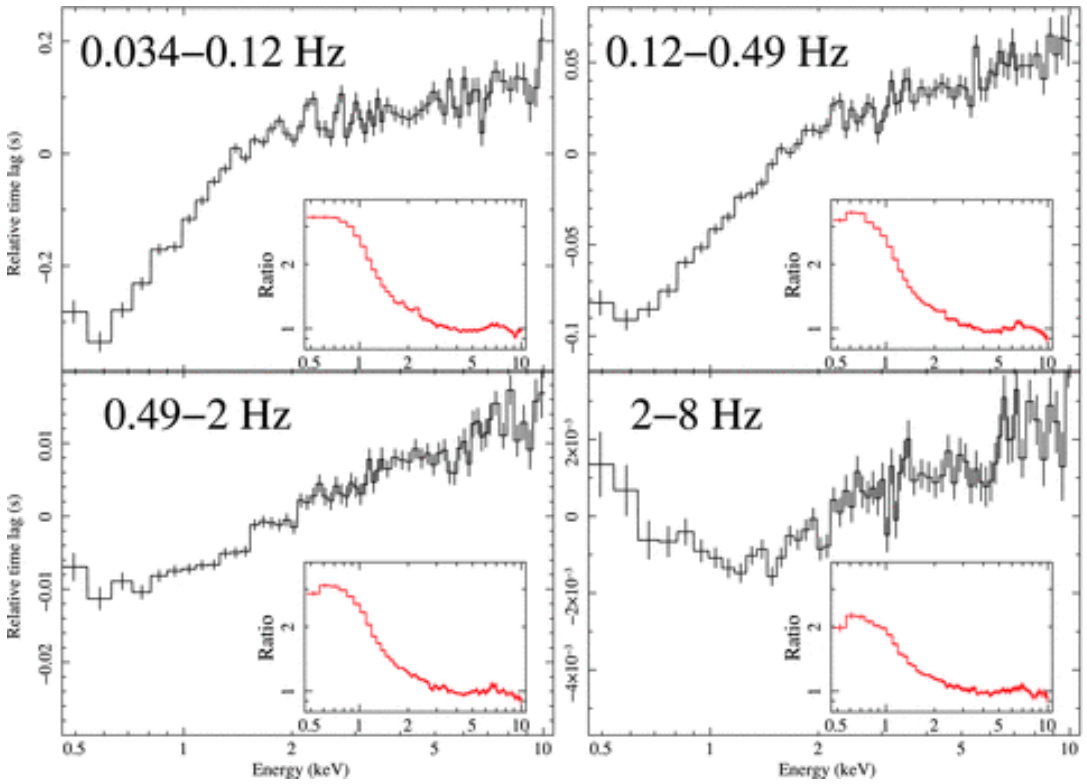


Figure 4.3: Phase-lag energy spectra of GX 339 − 4 with different frequency ranges. The insets are covariance spectra of the same frequency ranges. Figure taken from Uttley et al. (2011) .

4.2.1.3 Iron K Lag

The iron K lag in BHBs was first identified using data from the NICER observation of the hard state of MAXI J1820 + 070 (Kara et al., 2019). Figure 4.4 illustrates positive values around the energy of the iron K line, approximately 6.4 keV.

4.2.2 Spectral Dilution

4.2.2.1 Definition of Spectral Dilution

The time lag analysis is to derive the time lag in the time correlation between two light curves of different energy bands. Each band contains emission from the same spectral component to varying degrees. In the X-ray light curves of BHBs, it is often the case that a single spectral component dominates the emission in both energy bands. Figure 4.5 is an example of the multiband light curve of the BHB MAXI J1820 + 070. X-ray light curves of three energy bands (0.5–2.0, 2.0–5.0, and 5.0–10.0 keV) are shown. It can be seen that there is a common dominant signal in all bands. As a result, the cross-correlation functions for all combinations of these bands are dominated by the autocorrelation of the dominant signal (Figure 4.6).

As the cross-correlation function is heavily influenced by the dominant signal, delving into detailed time lag analyses is challenging. Consequently, the analysis of time lags in BHBs has predominantly shifted toward the use of cross spectra, specifically the phase

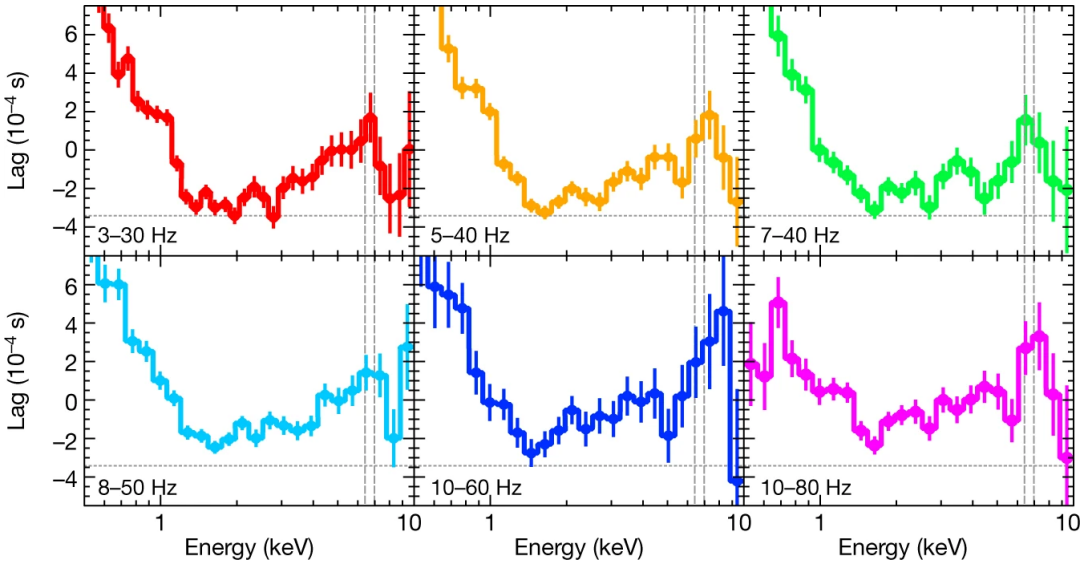


Figure 4.4: Phase-lag energy spectra of MAXI J1820 + 070 under various periods. Further details can be found in Kara et al. (2019). Figure taken from Kara et al. (2019). .

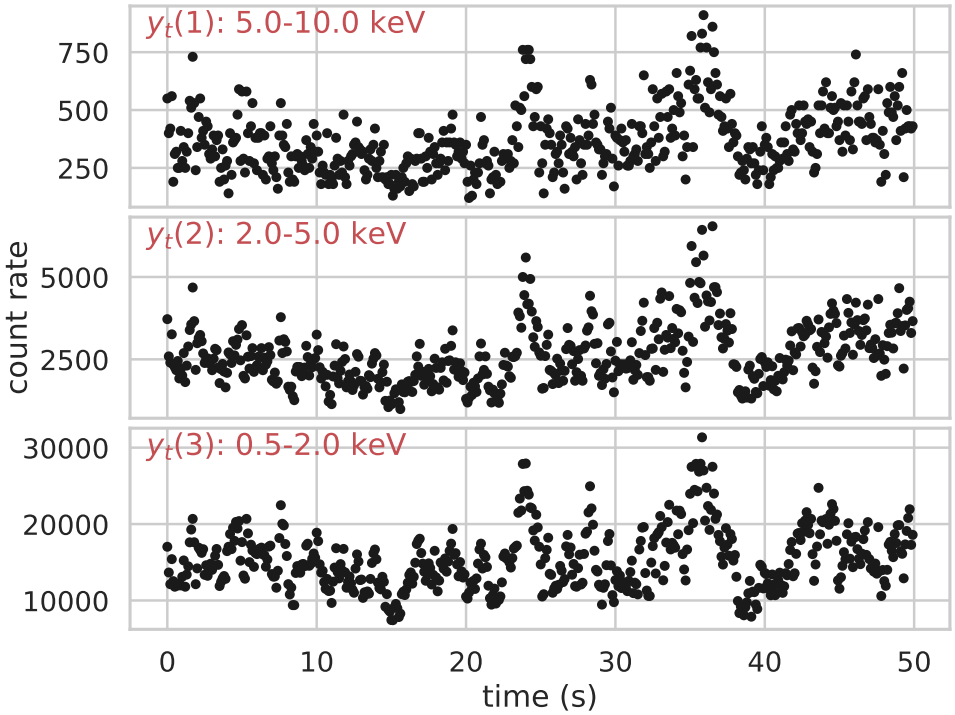


Figure 4.5: multiband light curve of BHB MAXI J1820 + 070. The energy bands are defined as 5.0–10.0 keV for $y_n(1)$, 2.0–5.0 for $y_n(2)$, and 0.5–2.0 keV for $y_n(3)$.

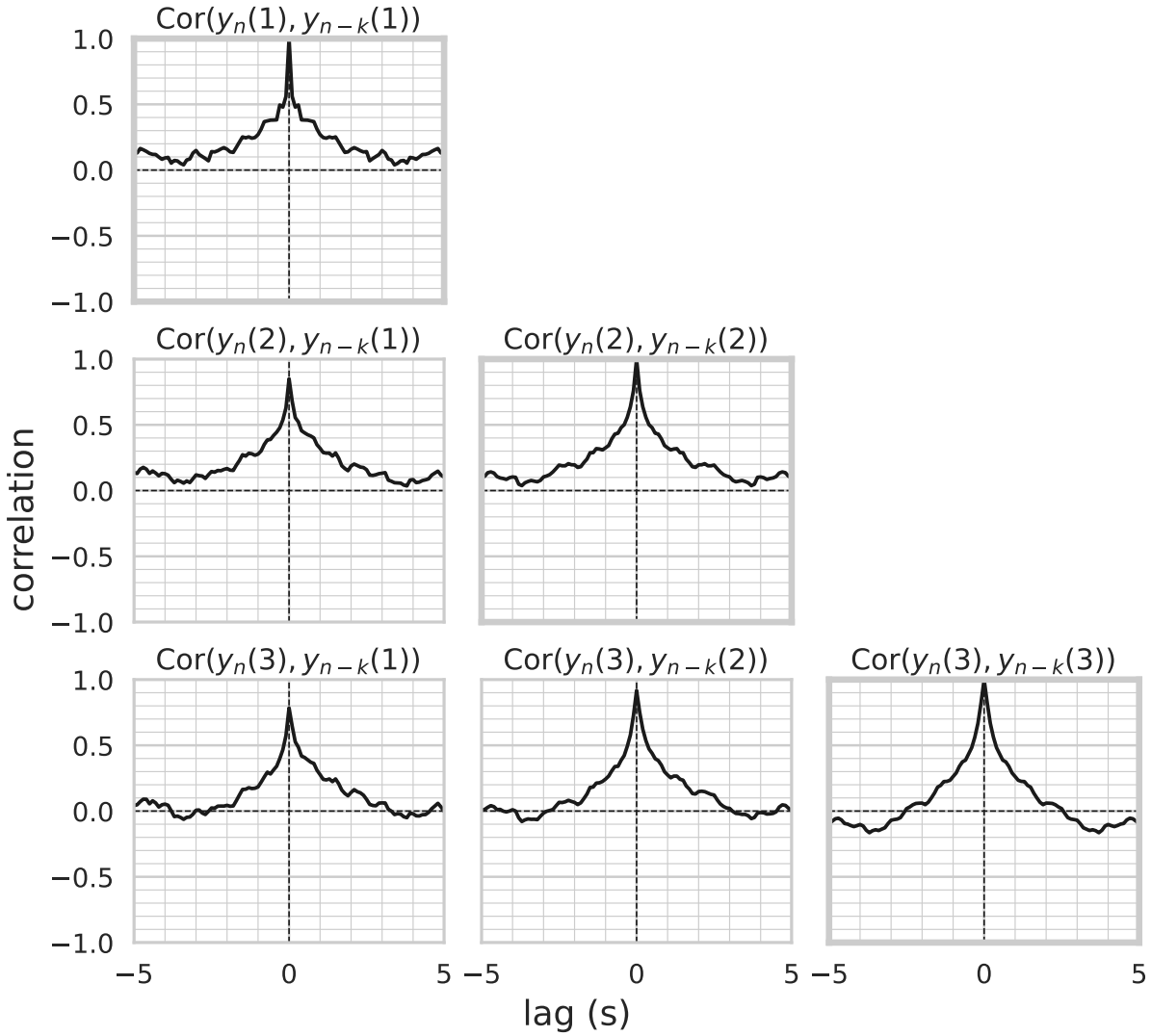


Figure 4.6: Cross correlation function for the multiband light curve of MAXI J1820 + 070.

spectra derived from them. The phase shift can be converted into the time shift or the lag amplitude, by multiplying the phase spectra by $(2\pi f)^{-1}$ for each frequency f. In simpler terms,

$$\tau_{jk}(f) = \frac{1}{2\pi f}\phi_{jk}(f) \tag{4.1}$$

Here, $\tau_{jk}(f)$ is the phase lag spectrum between the band j and k.

This metric suffers the spectral dilution. Consider two observed light curves, $y_n(1)$ and $y_n(2)$, in the energy band 1 and 2, which can be expressed as

$$\begin{aligned} y_n(1) &= x_n \\ y_n(2) &= c_1 x_n + c_2 x_{n-k} \end{aligned} \tag{4.2}$$

Here, we assume that the band 1 light curve is purely made of the spectral component x and the band 2 light curve includes both x and the time-lagged signal of x with the lag amplitude of k. This is the situation common in X-ray light curves of BHBs as in Figure 4.5. The phase lag spectrum is given by:

$$\tau_{12}(f) = \frac{1}{2\pi f}\arctan\left(\frac{(c_2/c_1)\sin 2\pi f k}{1+(c_2/c_1)\cos 2\pi f k}\right) \tag{4.3}$$

When the contribution of the second term of $y_n(2)$ is dominant ($c_2/c_1 \gg 1$), $\tau_{12}(f) = k$. When the contribution of the first term is dominant ($c_2/c_1 \ll 1$) in the band 2, $\tau_{12}(f) = 0$. In reality, c_2/c_1 is in between, which means that $\tau_{12}(f)$ is always smaller than k. At the $f \to 0$ limit,

$$\lim_{f\to 0}\tau_{12}(f) = \frac{c_2}{c_1+c_2}k \tag{4.4}$$

This is the spectral dilution. The derived time lag τ_{12} cannot be used as a measure of the actual time lag k (Poutanen, 2002; Uttley et al., 2014).

4.2.2.2 Workarounds

Some workarounds are used to address the spectral dilution. One is the zero crossing time and the other is the differential cross-correlation function.

Zero-Crossing Time Figure 4.7 illustrates the result of calculations with specific numbers in Equation (4.3) (Uttley et al., 2014). Here, the ratio is denoted as $R = c_2/c_1$. The left panel shows the relation between $\tau_{12}(f)$ and f for selected R. As expected, $\tau_{12}(f \to 0)$ decreases as R decreases. Also, $\tau_{12}(f)$ oscillates as a function of f. However, the frequency at which the lag first crosses the zero does not depend on the value of R and coincides with $1/2k$. This occurs because the frequency at which the numerator of Equation (4.3) first becomes zero (zero-crossing point) is given by

$$\begin{aligned} 2\pi f k &= \pi \\ f &= \frac{1}{2k} \end{aligned} \tag{4.5}$$

In the right panel of Figure 4.7, the time lag is represented by τ_0. It is confirmed that, regardless of the value of R, the zero-crossing point is at $1/(2\tau_0)$. This is used to avoid the spectral dilution in some studies (e.g., Mizumoto et al., 2019; De Marco et al., 2021).

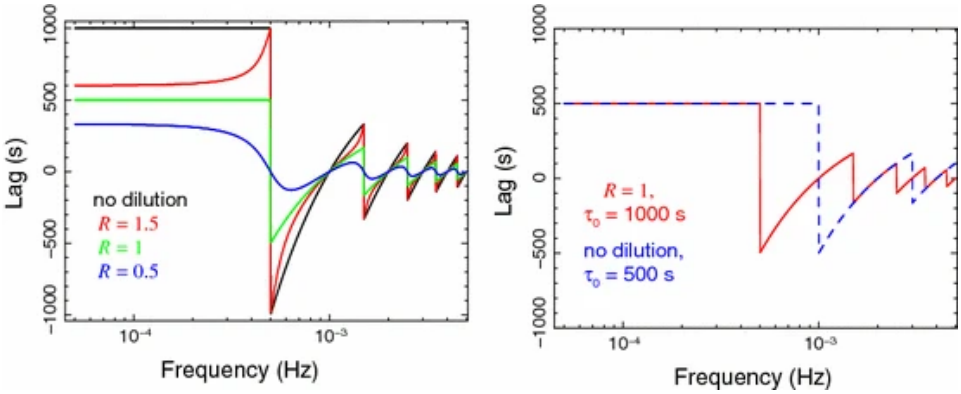


Figure 4.7: Theoretical phase-lag spectra with different situations. Left: the blue, red, and green curves represent spectra with $R \equiv c_2/c_1$ values of 0.5, 1.0, and 1.5, respectively, given an original time lag of 1000 s. Right: the red curve corresponds to $R = 1000$ s with a time lag of 1000s, and the blue curve corresponds to $R = 500$ s with a time lag of 500s. Despite their different original time lag amplitudes, the computed spectra are identical in the low-frequency range due to dilution. Figure taken from Uttley et al. (2014).

Differential Cross-Correlation Function Another approach is to stick to the cross-correlation function but use its asymmetry, which is noticeable in Figure 4.6. This is developed by Omama et al. (2023). The asymmetry in the cross-correlation function is likely a result of combining the autocorrelation function of the dominant signal plus the cross-correlation function of the time lag signal. Given the symmetry of the autocorrelation function, we can indirectly assess the cross-correlation function of the time lag signal by subtracting the negative lag from the positive lag correlation function, which is called the differential cross-correlation function. The method revealed the estimated time lag amplitude for the hard lag signal, ranging from sub-milliseconds to several seconds (Figure 4.8).

4.2.3 Need for Latent Variable

The time lag analysis so far replies on the correlation between X-ray light curves of two energy bands (§ 4.2). Each band has varying contributions of physical components. Therefore, the spectral dilution (§ 4.2.2) is an inherent problem that cannot be avoided whatever tools are used in the time and frequency domains. Some workarounds were proposed, but both of them reveal the actual time lag only as a subtle feature.

What we need to know is the variation of each spectral component, not the X-ray flux in an energy band. We cannot directly derive the variation of each spectral component

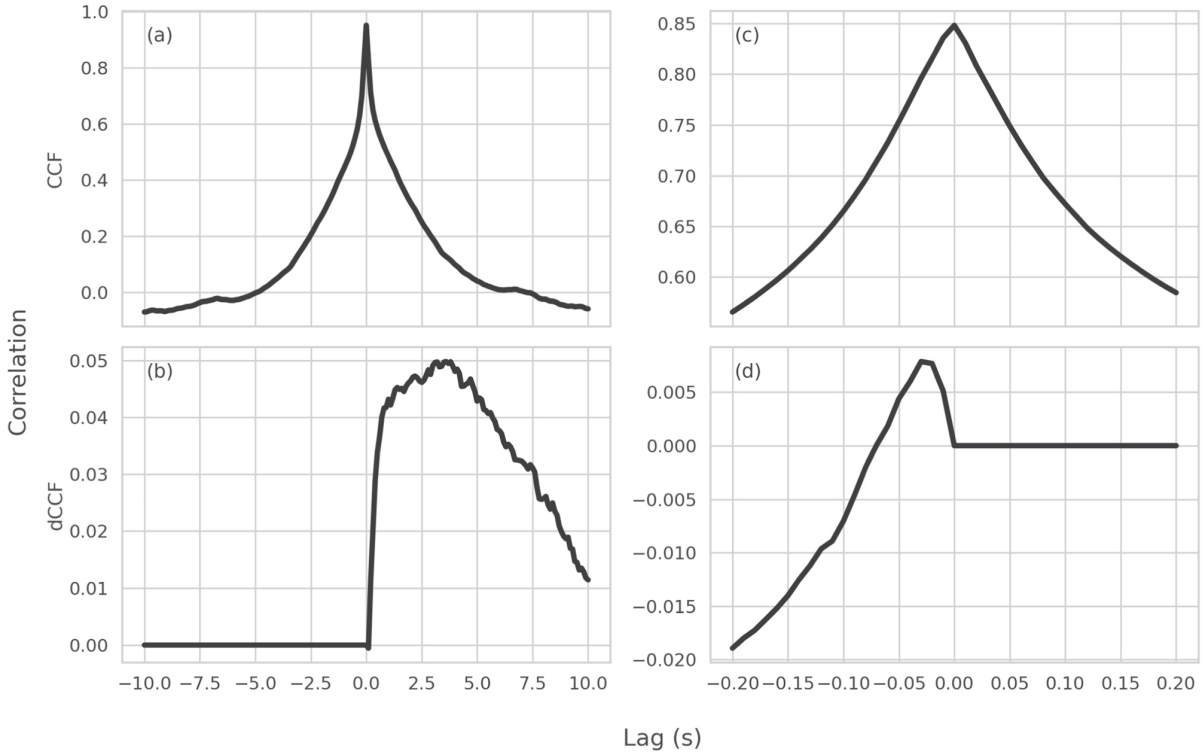


Figure 4.8: Cross-correlation function and differential cross-correlation function of MAXI J1820 + 070. Soft and hard bands are 0.5–1.0 keV and 1.0–10.0 keV. Figure taken from Omama et al. (2023).

from X-ray light curves. But what if we can estimate them indirectly, as in Figure 1.4? This is exactly the assumption of the state-space model. The state-space model, a statistical framework considering both observed and latent variables, offers a promising avenue to estimate the variability of the physical components from the observed light curves. In this study, we embark on a novel analysis using this approach.

Chapter 5

Observing Facility and Target

Contents

5.1 Observing Facility: NICER

5.1.1 Mission & Operation

The Neutron Star Interior Composition Explorer (NICER; Gendreau et al., 2016) is an X-ray observing instrument led by NASA Goddard Space Flight Center. It was launched on July 3, 2017, and was installed in the International Space Station (ISS). NICER's primary scientific goal is to elucidate the internal structure and physics of neutron stars by capturing their fast-timing behaviors. For this purpose, it realizes a high timing resolution and a large effective area, as well as a moderate spectral resolution.

Despite the specific mission goal, NICER's unique capabilities are useful for a much wider range of astrophysical applications that require simultaneous analysis of both energy spectra and timing features. In particular, time-resolved spectra uncompromised by the photon statistics and the detector dynamic range have been unavailable with preceding instruments with a similar concept, such as the Rossi X-ray Timing Explorer.

Figure 5.1 shows a schematic view of NICER. NICER's observing strategy is to track 3-6 targets during a ∼91 minute orbit of the ISS. The observation continues until it deviates from effective conditions, influenced by factors like the sun's position and the ISS's attitude. Once the conditions are met again, the operation resumes tracking new targets. This results in high-cadence observations of a target with a short duration per observation. NICER's unique aspect is its installation on the ISS, allowing for fast pointing and immediate observations of unexpected celestial events. The target command list is typically created twice a week to plan observations for dozens of celestial objects. When a transient source appears, the Target of Opportunity observation is activated, and new command sequence is swiftly updated and uploaded to NICER for immediate observations. If the staff is present, observations begin within 4 hours upon request (*NICER Mission Guide* 2023).

5.1.2 Instrument

The primary observing instrument onboard NICER is the X-ray Timing Instrument (XTI). To achieve the NICER's scientific objectives, XTI has the following features.

- Large effective area: 1900 cm^2 at 1.5 keV (Figure 5.2)
- Broad bandpass: $0.2 < E < 12.0$ keV
- Time-tagging resolution: < 300 ns
- Moderate spectral resolution: $6 < E/\Delta E < 80$ from 0.5 keV to 8 keV
- Restricted field of view: 30 $arcmin^2$

The XTI is made up of 56 sets of the X-ray Concentrator (XRC; Okajima et al., 2016) and the Silicon Drift Detector (SDD). Incoming X-rays are reflected by the XRC mirrors and collected by the SSD in the Focal Plane Module (FPM) located on XTI's backplane. Each FPM consists of an SDD, a pre-amplifier, and a housing. These FPMs are designed with a 2 mm aperture, allowing them to transmit X-rays from celestial

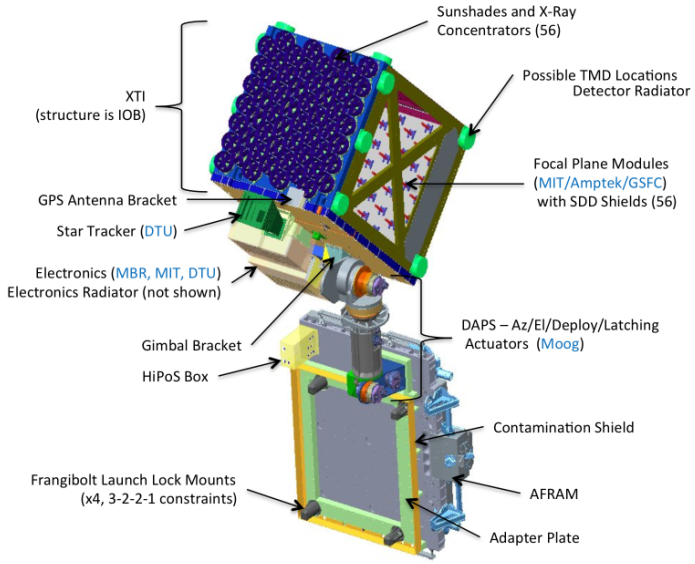


Figure 5.1: Schematic view of NICER (Okajima et al., 2016).

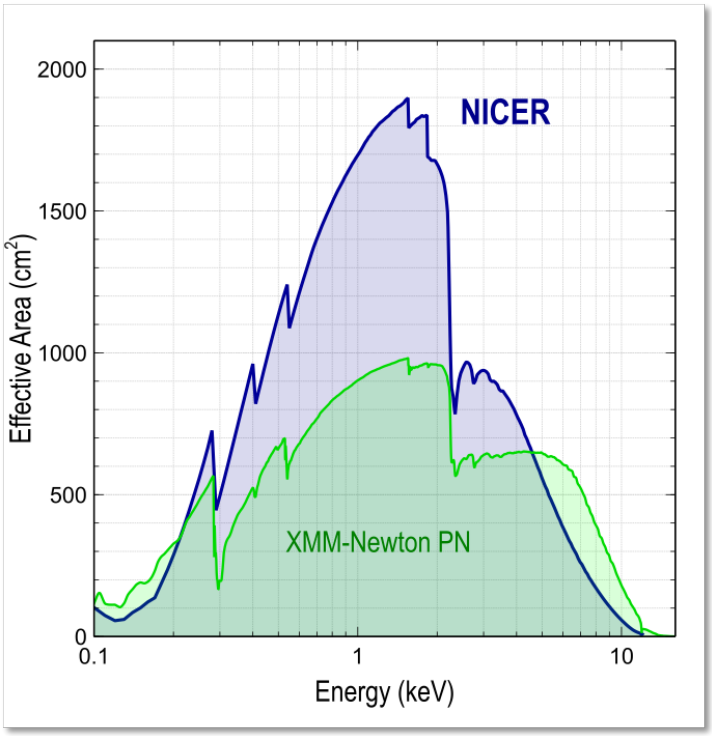


Figure 5.2: Effective area of NICER compared to the pn-type CCD camera onboard XMM-Newton*NICER Mission Overview* 2023

sources while reducing the influence of background radiation. This design also aims to improve timing performance.

In SDD (Prigozhin et al., 2016), the cathode electrodes are arranged in an onion-like shape, making the electric field in the radial direction. The electrons are generated in the X-ray photoelectric absorption with Si, which move along the radial gradient of the electric field and are collected by the anode electrode at the center. The amount of charge is proportional to the incoming energy, thus it works as an X-ray spectrometer. The detector also features a low noise by small capacitance and low leak current, which results in fast time of event discrimination, enabling high-time resolution.

The 56 FPMs are divided into seven groups, and each group containing eight FPMs. Electrical control and data readout for each group are managed by a single Measurement and Power Unit (MPU). Each FPM is assigned a unique Science Detector ID based on its sequential number within its group and the sequential number of the corresponding MPU (Figure 5.3). In data analysis, events are recorded and stored separately for each of these seven MPUs. Among them, four FPMs (11, 20, 22, and 60) are dysfunctional and are not used.

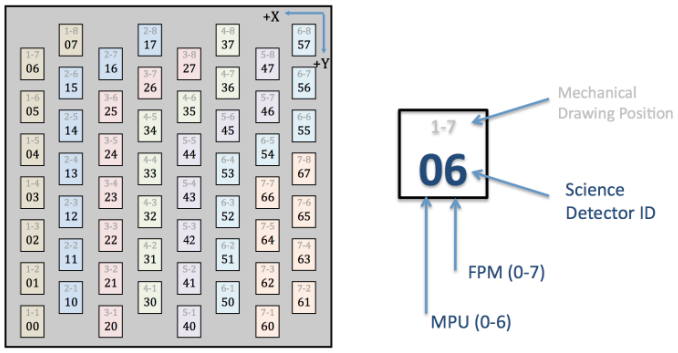


Figure 5.3: Detector layout of NICER. Figure from *NICER Mission Guide* (2023)

5.1.3 Data Processing

5.1.3.1 Onboard Processing

MPU is responsible for the onboard signal processing. For each detected X-ray as an analog electrical signal, its pulse is shaped with two different time constants, referred to as the "slow" and "fast" chains. The "slow" chain has a peak time of 465 ns and is optimized for energy measurement. On the other hand, the "fast" chain has a peak time of 85 ns and is optimized for timing measurements. When both fast and slow chains detect an event, the time stamp of the fast chain is used. However, the fast chain has a lower signal-to-noise ratio than the slow chain and cannot reliably detect lower energy events ($\leq$ 1.0 keV). In such cases, the time stamp of the slow chain is utilized (Prigozhin et al., 2016). This switching of time stamp usage enhances the overall timing accuracy of the X-ray Timing Instrument (XTI), ensuring excellent timing precision.

5.1.3.2 Ground Processing

After receiving the data from the ISS, the data are processed through the pipeline on the ground. Event data from the X-ray Timing Instrument (XTI) are typically processed through three main levels:

1. **Unfiltered File(UF)**: These are the raw data detected by XTI and are organized into seven groups, one for each MPU. The term `uf` is contained in the file name of this stage. Although these files contain events distributed over seven MPUs, they are not suitable for scientific analysis. Note that in the UF file, the pulse heights are stored in `PHA` and `PHA_FAST` of the `EVENTS` extension, but these values are not corrected for different energy gains for each MPU.

2. **Calibrated Unfiltered File (UFA)**: The events from the UF are merged, and an energy calibration is applied. The term `ufa` is contained in the file name of this stage. These files still contain particle and background events and are not used for scientific analysis. However, they can be use to set observatory parameters such as dead time and high background intervals.

3. **Cleaned Events (CL)**: In this stage, events relevant to the appropriate energy range for NICER analysis are extracted and processing steps, such as the removal of background and particle events, are performed. The term `cl` is contained in the file name of this stage.

For actual scientific analysis, it is recommended to use data processed with the NICER Level 2 (`NICERL2`) pipeline. `NICERL2` is a task within the NICER Data Analysis Software (`NICERDAS`) designed specifically for NICER data analysis. It applies the latest pipeline processes to the data. The key difference between archived CL files and `NICERL2`-processed data is that CL files undergo standard filtering processing defined at the start of the mission, whereas `NICERL2` processes the data with the latest pipeline algorithms and calibration database. Therefore, data processed with different versions of `NICERL2` may exhibit subtle differences.

5.2 Target: MAXI J1820+070

5.2.1 Observational Properties

MAXIJ1820+070 is a transient low-mass BHB discovered in March 2018 using MAXI(Kawamuro et al., 2018). Initially observed in the hard state, it stayed in this state for about 100 days before transitioning to the soft state. This long period in the hard state made MAXI J1820 + 070 an excellent candidate for studying BHBs in the hard state. Its outstanding properties, summarized in Table 5.1, made it an ideal source for BHB studies. At its brightest, it reached up to 4 Crab in the 2–6 keV energy range (Shidatsu et al., 2018). The estimated distance is 2.96 ± 0.33 kpc based on the radio parallax, consistent with the optical parallax ($3.46^{+2.18}_{-1.03}$ kpc; Gandhi et al., 2019). This proximity results in a low galactic absorption ($N_{\rm H} \sim 10^{-21}$ cm^{-2}; Uttley et al., 2018), which facilitates studies in the low X-ray energy band. The inclination, estimated at 66–81 based on the projected rotational velocity of the donor star, also constrains the donor-to-black hole mass ratio to 0.072 ± 0.012. These values lead to a black hole mass estimate of 5.73–8.34$M_\odot$ and a donor star mass estimate of 0.28–0.77$M_\odot$. Due to these favorable conditions, intensive observations were made using a suite of X-ray telescopes with MAXI (Matsuoka et al., 2009), NICER (Gendreau et al., 2012), Swift Niel Gehrels Observatory (Burrows et al., 2005), NuSTAR (F. A. Harrison et al., 2013), as well as those in other wavelengths.

Table 5.1: Observed properties of MAXI J1820+070.

Parameter	Value	Unit	Method	Reference
Distance	2.96 ± 0.33	kpc	Radio parallax	Atri et al., 2020
Galactic absorption	1.5×10^{21}	cm^{-2}	X-ray spectra	Uttley et al., 2018
Mass (Black Hole)	5.73–8.34	$M_\odot$	Doppler measurement of donor	Torres et al., 2020
Mass (Donor)	0.28–0.77	$M_\odot$	Doppler measurement of donor	Torres et al., 2020
Inclination	66–81	degree	Doppler measurement of donor	Torres et al., 2020

5.2.2 State Changes

Figure 5.4 shows the X-ray light curve acquired by MAXI in the soft ($S(t)$; 2–6 keV) and hard ($H(t)$; 6–20 keV) bands, along with the spectral hardness defined as $H(t)/S(t)$ (Shidatsu et al., 2019). Figure 5.5 illustrates the HID of the same duration. As explained in § 4.1, it shows the characteristic state transition pattern of BHBs, moving circularly counterclockwise from the bottom right. Based on this observed behavior, three fundamental states were identified: low/hard state (LS), high/soft state (HS), and intermediate state (IM) (Done et al., 2007), as indicated in Figure 5.4.

Figure 5.6 illustrates the energy spectra of representative data in the soft and hard states. In the soft state, the dominant disk component peaks at 2 keV, while it diminishes in the hard state, where the power-law component takes precedence. These patterns follow typical features of the soft and hard states of BHBs (§ 4.1.2). The periodograms

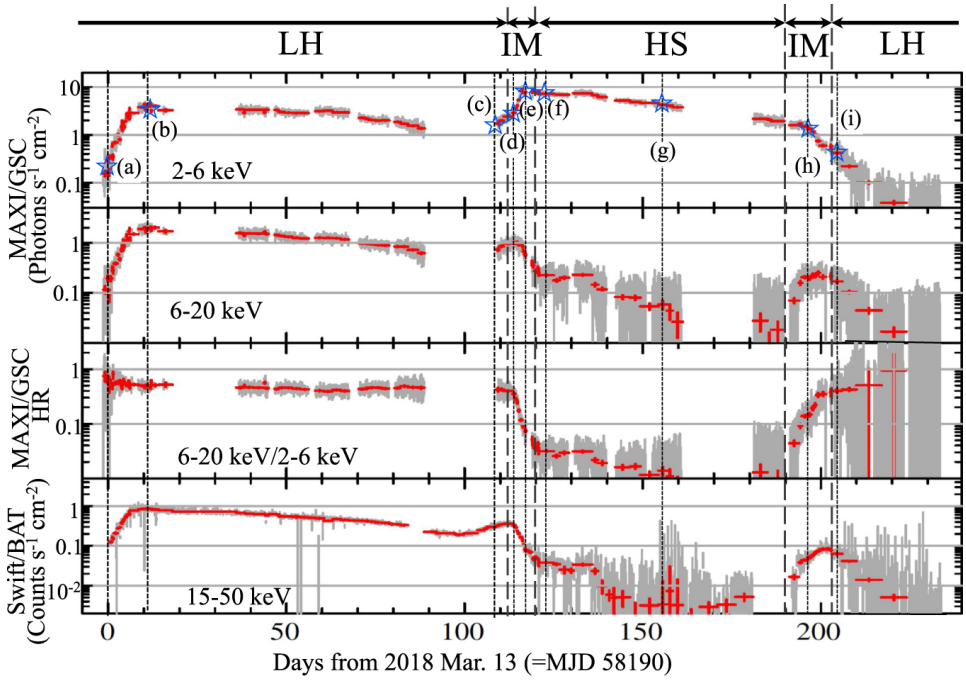


Figure 5.4: Light curve and hardness ratio of MAXI J1820 + 070 obtained by MAXI. Figure is taken from (Shidatsu et al., 2019).

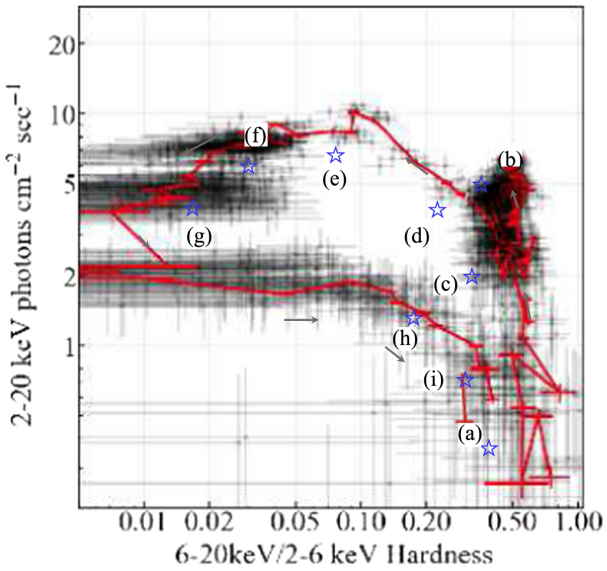


Figure 5.5: HID of MAXI J1820 + 070 observed by MAXI. Figure is taken from (Shidatsu et al., 2019).

further confirm that the defined states align with the state definition (Figure 5.7). Specifically, in the soft state, there are no distinctive powers across all frequency ranges, while in the hard state, noticeable power breaks and locally significant QPOs are observed.

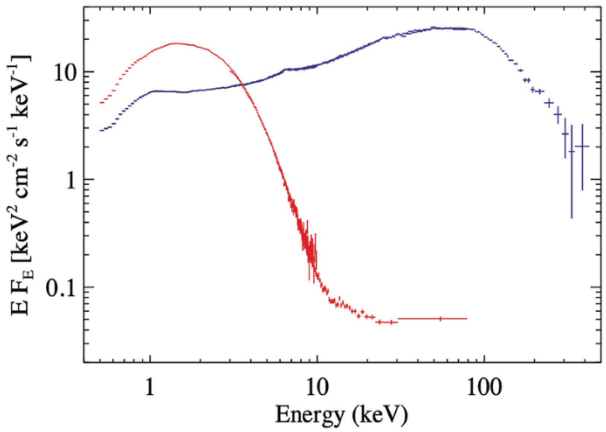


Figure 5.6: Energy spectra of MAXI J1820 + 070 in the soft and hard states. The red and blue curves correspond to the energy spectra of the soft band and hard bands. Figure taken from (Kalemci et al., 2022)

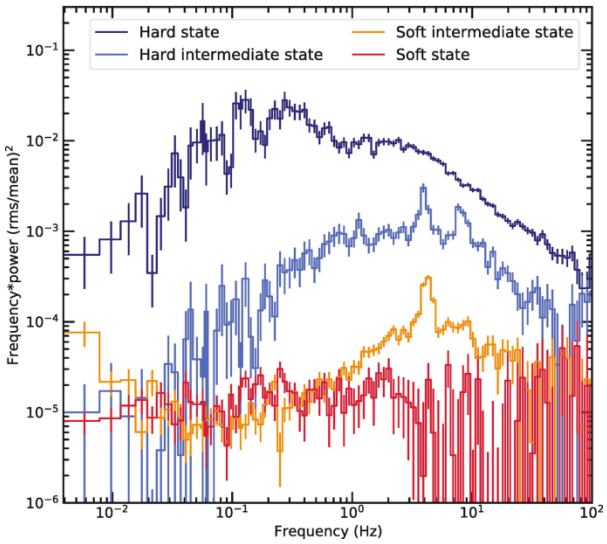


Figure 5.7: Periodograms of MAXI J1820 + 070 in different states. Figure is taken from (Kalemci et al., 2022)

5.2.3 Geometry in the Hard State

The geometry of the accretion disk and accretion flow of BHBs is in intense debate. The most important parameter is the inner radius of the accretion disk, which depends on the gravitational field of the black hole governed by the general relativity. In the soft state, it is well established that the inner edge of the accretion disk extends very close to the innermost stable orbit of the black hole. This is also the case for MAXI J1802+070 (Fabian et al., 2020).

In the hard state, there are two main competing ideas. In one model, the inner edge of the accretion disk remains close to the black hole as the soft state. The lamppost configuration is proposed. This assumes that the source of the X-ray emission is located just above the rotational axis of the black hole. The X-ray emission from the lamppost is reprocessed on the surface of the accretion disk, which produces time lags and heavily distorted Fe K emission lines. This theory is employed to explain some of the observed features of MAXI J1802+070 (e.g., Kara et al., 2019, Buisson et al., 2019).

The other idea is that the accretion disk is truncated, and the inner radius is far away from the black hole. From the inner radius of the truncated disk, the matter accretes in the form of the accretion flow, which is much more tenuous (Esin et al., 1997). This theory is alternatively employed to explain the same observed features of MAXI J1802+070 (e.g., Zdziarski et al., 2021; De Marco et al., 2021; Axelsson and Veledina, 2021; Kawamura et al., 2022; Omama et al., 2023).

For estimating the inner disk radius, the time lag analysis is often used. Two main features are studied— the soft lag and the Fe K lag. In the soft lag, the soft band emission is delayed in time in comparison to the emission in the hard band. In the Fe K lag, the Fe K band emission is delayed from the others. These lags are considered to reflect the physical distance between the corona and the accretion disk (Uttley et al., 2014; Kalemci et al., 2022). Although the phase spectrum of the cross-spectrum is commonly used for the estimation, this method suffers the spectral dilution issue when naively applied (see § 4.2.2).

Using the NICER observation data of MAXI J1820 + 070, Kara et al. (2019) found that the Fe K emission lags behind other bands by a time scale of sub-milliseconds based on the cross-spectral phase spectrum amplitudes (§ 6.1.3.2) and translated it to be a distance of $\sim 10 r_g$, where r_g is the gravitational radius. Using this and the shape of the Fe K line emission, they argued for the lamppost geometry with the accretion disk extending close to the black hole with an inner radius of a few r_g (Figure 5.8).

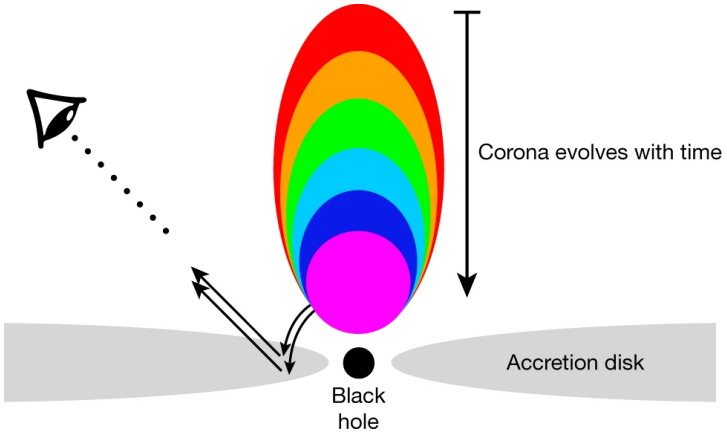


Figure 5.8: Schematics of the lamppost model. Figure is take from Kara et al., 2019)

In contrast, De Marco et al. (2021) used the cross-spectral timing analysis of the

same data set. They estimated the time lag amplitude of the soft lag using the zero-crossing frequency. They found that the lag amplitude is an order of magnitude larger than the one estimated by Kara et al. (2019), supporting the truncated disk model. They also observed a decrease in the amplitude of the soft lag signal and interpreted it as evidence that the inner disk radius is reduced, indicating the development of the disk. Evidence for a truncated disk is strengthened by results of frequency-resolved spectroscopy (Axelsson and Veledina, 2021). In this method, the energy spectrum is compared to sliced energy spectra, called frequency-resolved energy spectra, at various frequencies. The frequency-resolved energy spectra align well with the truncated disk model, rather than the lamppost model.

Different conclusions were drawn from the same data set. The data analysis of the previous studies was all based on the conventional light curve analysis in our definition. A new approach to data analysis is necessary.

5.2.4 Data Definition

Figure 5.9 shows the X-ray light curve in the soft (0.5–1.0 keV) and hard (1.0–10.0 keV) bands ($S(t)$ and $H(t)$, respectively) and the spectral hardness is defined as $\{H(t) - S(t)\} / \{H(t) + S(t)\}$ using NICER. The light curve during the first hard state can be divided into four distinctive phases following De Marco et al. (2021): the rise, plateau, bright decline, and hard-soft transition phases in chronological order. After the rapid brightening at the beginning, the flux change was slow with a noticeable break between MJD 58240 and 58260. The source started a transition to a new state in the last part after MJD 58280.

In this study, we selected the data taken at an epoch immediately after the outburst on March 21, 2018, which is indicated by the red line in Figure 5.9. This epoch is within a typical low hard state with extreme statistics in counts. This dataset was investigated intensively in many papers (e.g., Kara et al., 2019, Buisson et al., 2019, De Marco et al., 2021, Zdziarski et al., 2021), thus we can compare results. The NICER observation sequence number is 1200120106. We now call this dataset "O106" hereafter. We summarize the observation properties of MAXI J1820 + 070 used for this study in Table 5.2.

Table 5.2: NICER Observation of MAXI J1820 + 070.

ObsID	Date	Time (UT)	Exposure (s)
1200120106	2018-03-21	09:15:20	5437.54

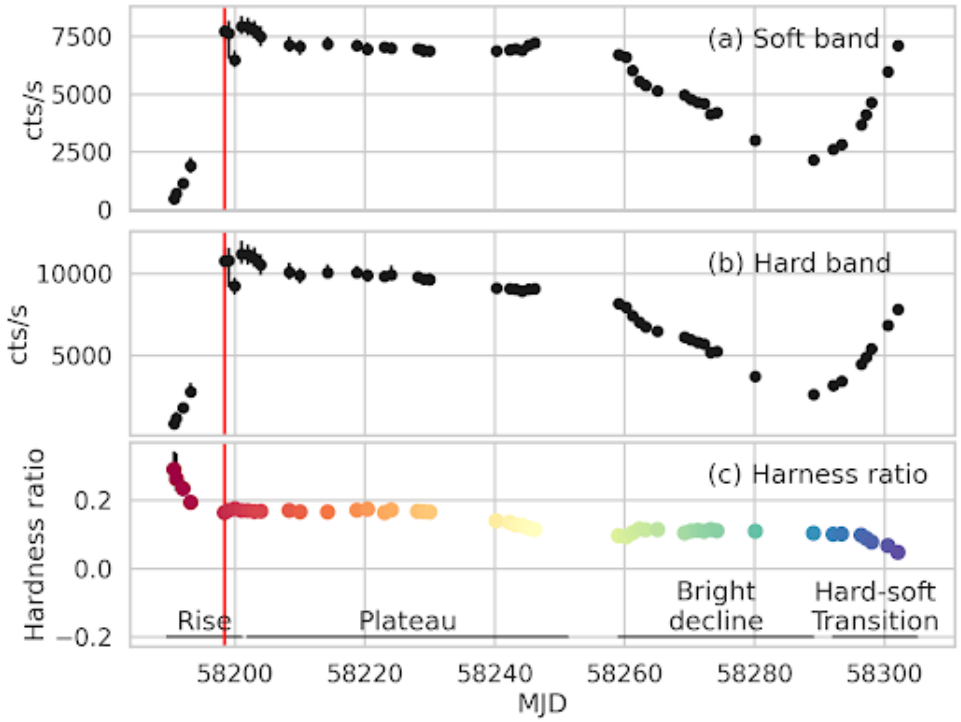


Figure 5.9: Entire light curve in the hard state (taken from Omama et al., 2023). The soft band (a) and hard band (b) are defined as 0.5–1 keV. The red vertical dashed line is the date when O106 data was obtained.

Chapter 6

Analysis and Results

Contents

6.1 Time Series Data

6.1.1 Generation

We start with the X-ray event list of the O106 data set. Each event has the energy and arrival time of individual X-ray photons and background, which is almost negligible for this bright source. From the data, we generated multiple time series data (light curves) in different energy bands. We need to determine the energy bands, time bin, and data length for generating the time series.

Energy Band Using the entire duration of the O106 data, we constructed the time-averaged energy spectrum in Figure 6.1. We fit the spectrum with a phenomenological model commonly used for BHBs:

```
phabs * (powerlaw + diskbb + gauss + gauss + gauss)
```

Here, `phabs` represents the interstellar photoelectric absorption, `powerlaw` represents a power-law component for the Comptonized emission, `diskbb` represents the multi-temperature disk blackbody emission from the accretion disk, the first `gauss` represents a broad excess emission around 0.1 keV, and the remaining two `gauss` components represent the broad and narrow components of the Fe Kα line emission. We used the energy range of 0.7–10 keV and obtained a reasonable fitting result (Table 6.1).

Table 6.1: Reasonable fitting result of the energy spectrum fitting. The asterisk after the value indicates a frozen parameter. The colors in the energy spectrum (Figure 6.1) are denoted next to the component names.

Component	Parameter	Unit	Value
`phabs`	nH	10^{22}	0.15*
`powerlaw` (blue)	PhoIndex		1.63
	norm		3.77
`diskbb` (red)	Tin	keV	0.33
	norm		$2.06 \times 10^{+4}$
`gaussian` (green)	LineE	keV	1.02×10^{-7}
	Sigma	keV	0.47
	norm		18.0
`gaussian` (yellow)	LineE	keV	6.61
	Sigma	keV	0.56
	norm		1.76×10^{-2}
`gaussian` (yellow)	LineE	keV	6.39
	Sigma	keV	5.09×10^{-2}
	norm		1.58×10^{-3}
reduced χ^2			22.2

Some structures are recognized in the residuals, but this is good enough as the goal is not to devise a physically-motivated spectral model but to determine the energy band for light curves. The Comptonized component dominates the entire energy band, but the

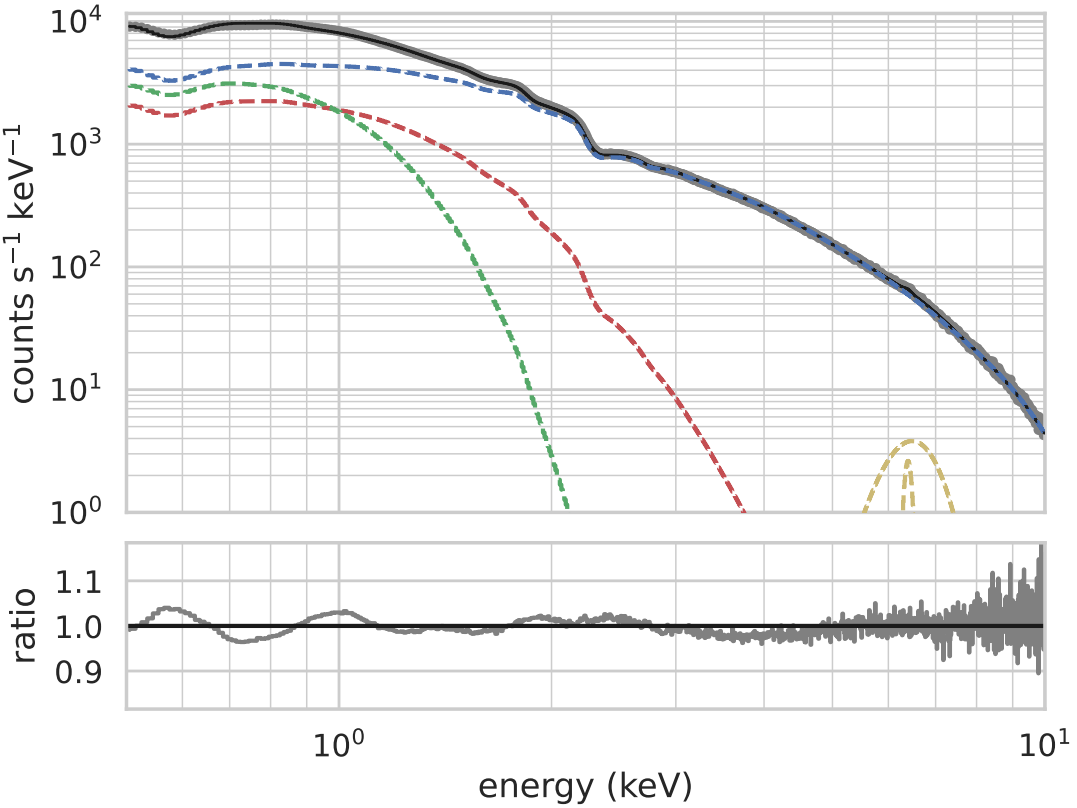


Figure 6.1: Energy spectrum of O106 and the best-fit model (upper panel) and residuals (lower panel). The different components, including Comptonization (`powerlaw`), disk black body (`diskbb`), soft excess (`gauss`), and broad and narrow Fe lines (`gauss` and `gauss`), are represented by the blue, red, green, and yellow curves, respectively. The residuals are computed by taking the ratio of the observation to the model.

other two broad components (disk blackbody and soft excess) contribute differently in different energy bands. The disk blackbody component contributes up to ∼5 keV, while the soft excess component is only up to ∼2 keV. We thus define three energy bands: 10.0–5.0, 5.0–2.0, and 2.0–0.5 keV.

Time Bin Next, we consider the appropriate bin size of the light curve. We want the noise to behave close to Gaussian for the convenience of modeling. We set the criterion to have an average count of ∼ 30 per bin and chose a bin size of 0.1 s. BHBs exhibit variations over a wide range of time scales with noise. The bin size limits our ability to investigate the variation above the Nyquist frequency of 5 Hz, but this is much wider than other data sets thanks to the extreme brightness of the source and the large effective area of NICER.

Data Length We chose 50 s for the data length, thus the total number of bins is 500. We can access the frequency down to 1/50=0.02 Hz. When determining the data length of the multiband light curve to be analyzed, it is necessary to consider the assumptions

Table 6.2: Band definition used for the AR and VAR model.

Band	Name	Lower edge (keV)	Higher edge (keV)	p-value	AR order
1	hard	5.0	10.0	0.002	6
2	medium	2.0	5.0	0.005	6
3	soft	0.5	2.0	0.000	7

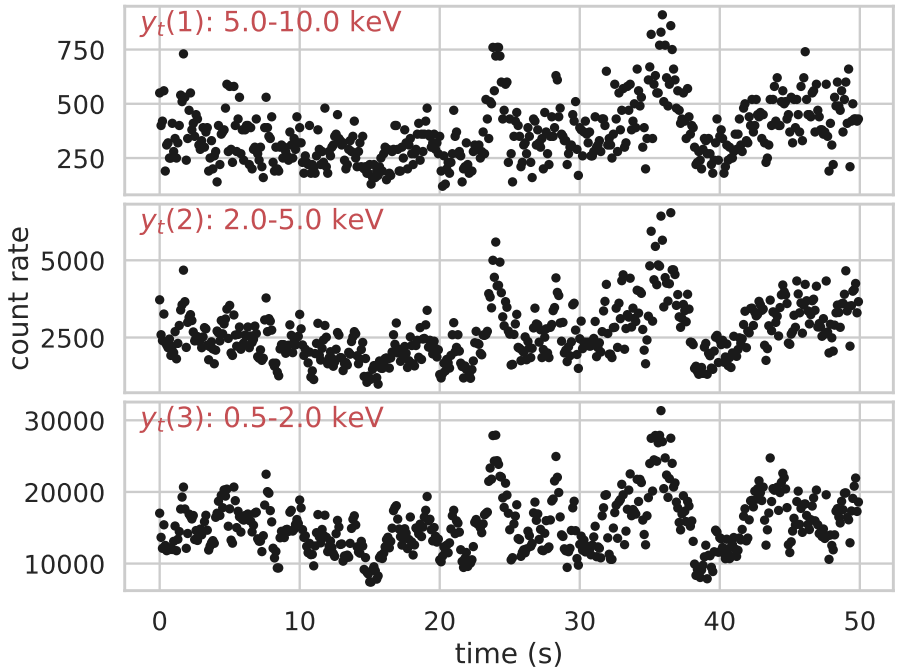


Figure 6.2: multiband light curve of O106.

made for the system as well as the computing time. We use 50 s, which we believe is short enough to ensure that the system remains stationary, which is the premise of the modeling described below.

We now obtained the three light curve $y = [y_n(1)\ y_n(2)\ y_n(3)]^T$ to represent the bands 10.0–5.0, 5.0–2.0, and 2.0–0.5 keV, respectively (Table 6.2). Figure 6.2 shows the light curves for these three bands. The variability in the three bands is correlated, but some variations (like the one observed at 45 s) are distinctive in a particular band.

6.1.2 Inspection and Pre-Processing

Stationary Check The approach to time series modeling varies significantly depending on whether the time series is stationary or non-stationary. Here, we test the stationarity of the generated multiband light curves using the ADF test (§ 3.3.1). We utilize `statsmodels.tsa.stattools.adfulle`[1] for the test (Seabold and Perktold, 2010).

For each of the light curves in the three bands, we derived the optimum AR order that minimizes the AIC. The AR model was constructed and the residuals were tested with the ADF test. Table 6.2 tabulates the AR order and the p-values of the ADF test.

[1] https://www.statsmodels.org/dev/generated/statsmodels.tsa.stattools.adfuller.html

A significance level of 5% is considered as the rejection point. As all bands have p-values below the threshold. We thus conclude that the null hypothesis that the time series is a unit root AR model is rejected for the three bands.

Normalization The AR and VAR models assume a zero mean for time series. Additionally, as shown in Figure 6.2, the count rates vary widely among different energy bands, which makes the modeling biased. We thus normalizes the individual light curves so that the mean is 0 and the variance is 1. We hereafter use the normalized light curves.

6.1.3 Description Functions

We characterize the given time series using the sample cross-correlation functions and sample cross-spectrum before modeling the time series.

6.1.3.1 Sample Cross-Corelation Function

We explore the correlations within each band and among the three bands using the sample cross-correlation function (Figure 6.3) using Equation (3.40) . For computing the sample cross-correlation function, we use `scipy.signal.correlate`[2] and `scipy.signal.correlation_lag`[3] functions (Virtanen et al., 2020). As often seen in other BHBs, a strong correlation is found at the time lag = 0 made by the most dominant emission component. As the lag increases, the correlation decreases. The shape is asymmetrical, though, with the positive lags being more enhanced than the negative lags, indicating the presence of a hard lag signal. The correlation of this hard lag signal is particularly evident by comparing $\mathrm{Cor}(y_n(1), y_n(3))$ and $\mathrm{Cor}(y_n(3), y_n(1))$.

6.1.3.2 Sample Cross-Spectrum

We next calculate the sample cross-spectrum from the light curve. As explained in Chapter 3, the cross-spectrum essentially behaves like a power spectrum. Using the `scipy.signal.csd`[4] function, where the Welch's method is used, we compute the sample cross-spectrum for the observations of O106. These observations, comprising 500 data points, are segmented into 100 points, and we calculate the mean of nine segments with a 50-point shift.

The periodogram exhibits some characteristics commonly seen in BHBs (Figure 6.4). The power spectra of BHBs typically have a curvature (Remillard and McClintock, 2006) with the slope of the powers gradually decreasing with increasing energy. This makes a difference in different energy bands.

The panels below the diagonal line in Figure 6.5 illustrate the sample phase spectra. These spectra are mostly close to zero across frequencies, indicating a dominant signal

[2] https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.correlate.html
[3] https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.correlation_lags.html
[4] https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.csd.html

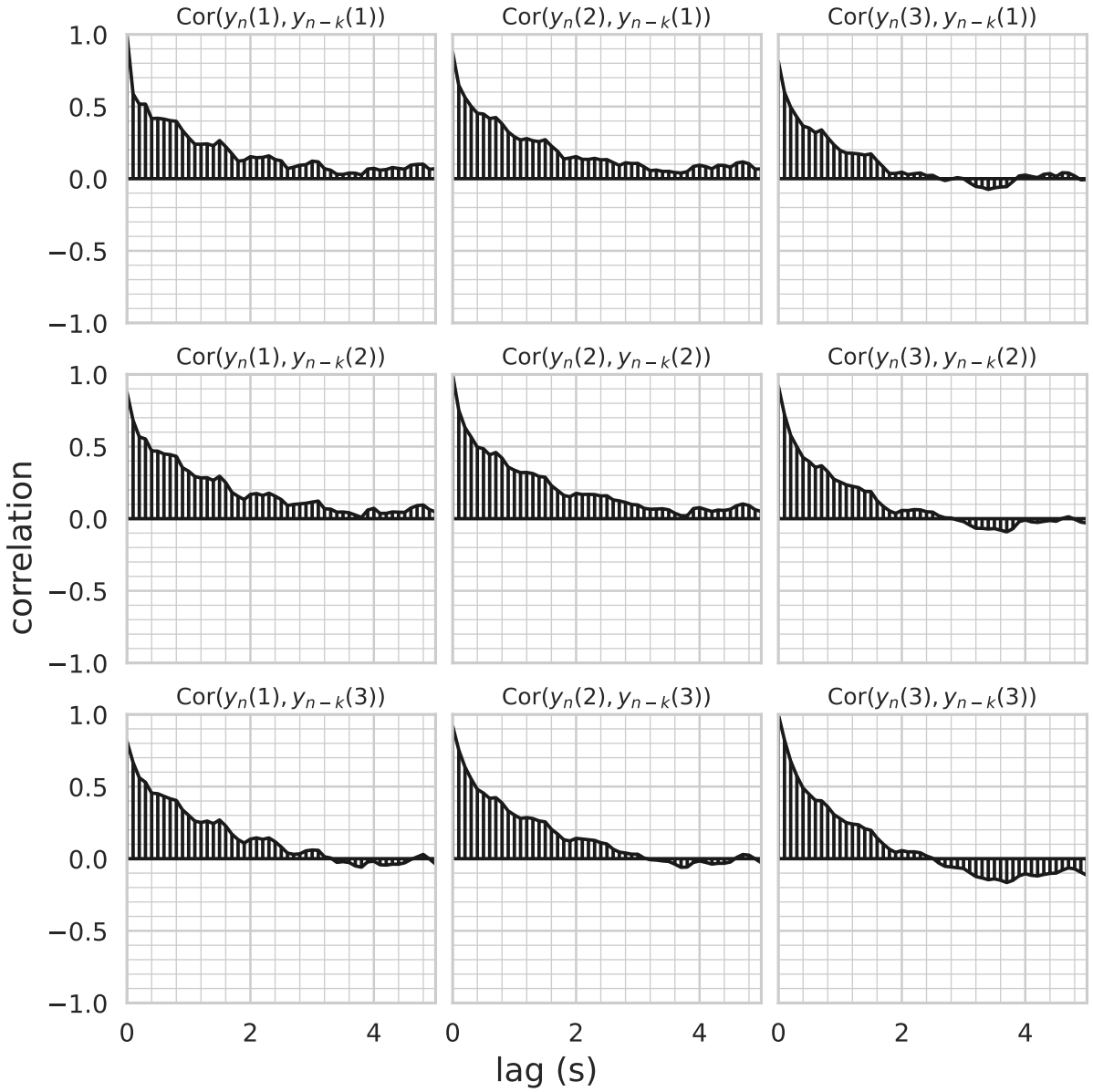


Figure 6.3: Sample cross-correlation functions of O106. The correlations between the two out of three light curve data are displayed. The diagonal panels are autocorrelation functions. A positive correlation indicates that the y-axis band is delayed compared to the x-axis band. Note that the panels below the diagonal line are the time-inverted version of those above the diagonal line, and thus are omitted.

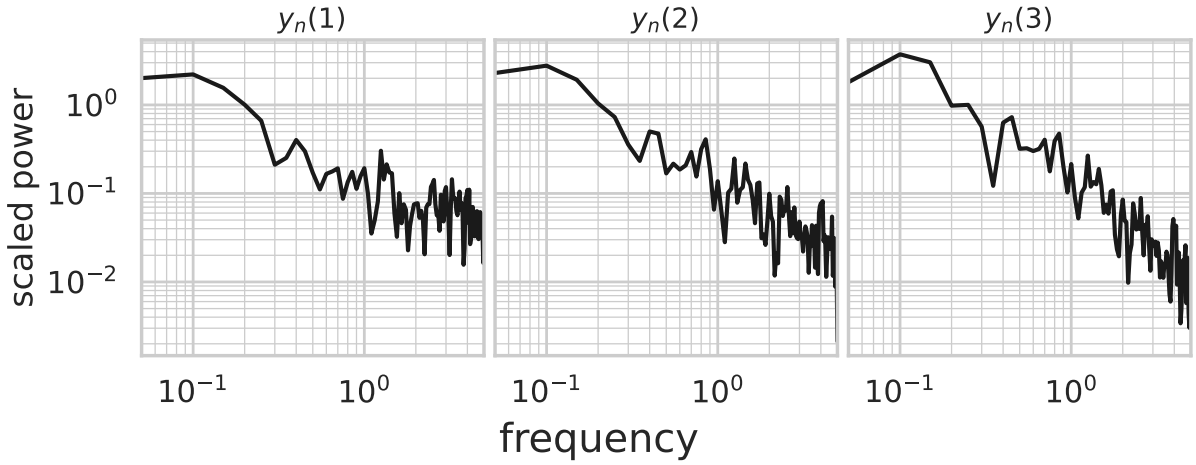


Figure 6.4: Periodogram of MAXI J1820 + 070.

with zero phase shifts. The spectra above ∼1 Hz are too noisy to interpret. Around 0.1 Hz, there is a slight positive deviation, likely due to the hard lag signal identified in Figure 6.3. The panels above the diagonal line in Figure 6.5 show the sample amplitude spectra. They all share a similar shape without notable differences.

The panels below the diagonal line in Figure 6.6 display the sample coherency computed using Equation (3.46). Notably, the coherency between $y_n(1)$ and $y_n(2)$ is higher at low frequencies compared to the other two. Although high-frequency coherencies across all combinations exhibit noise, the decreasing trends appear to be common features.

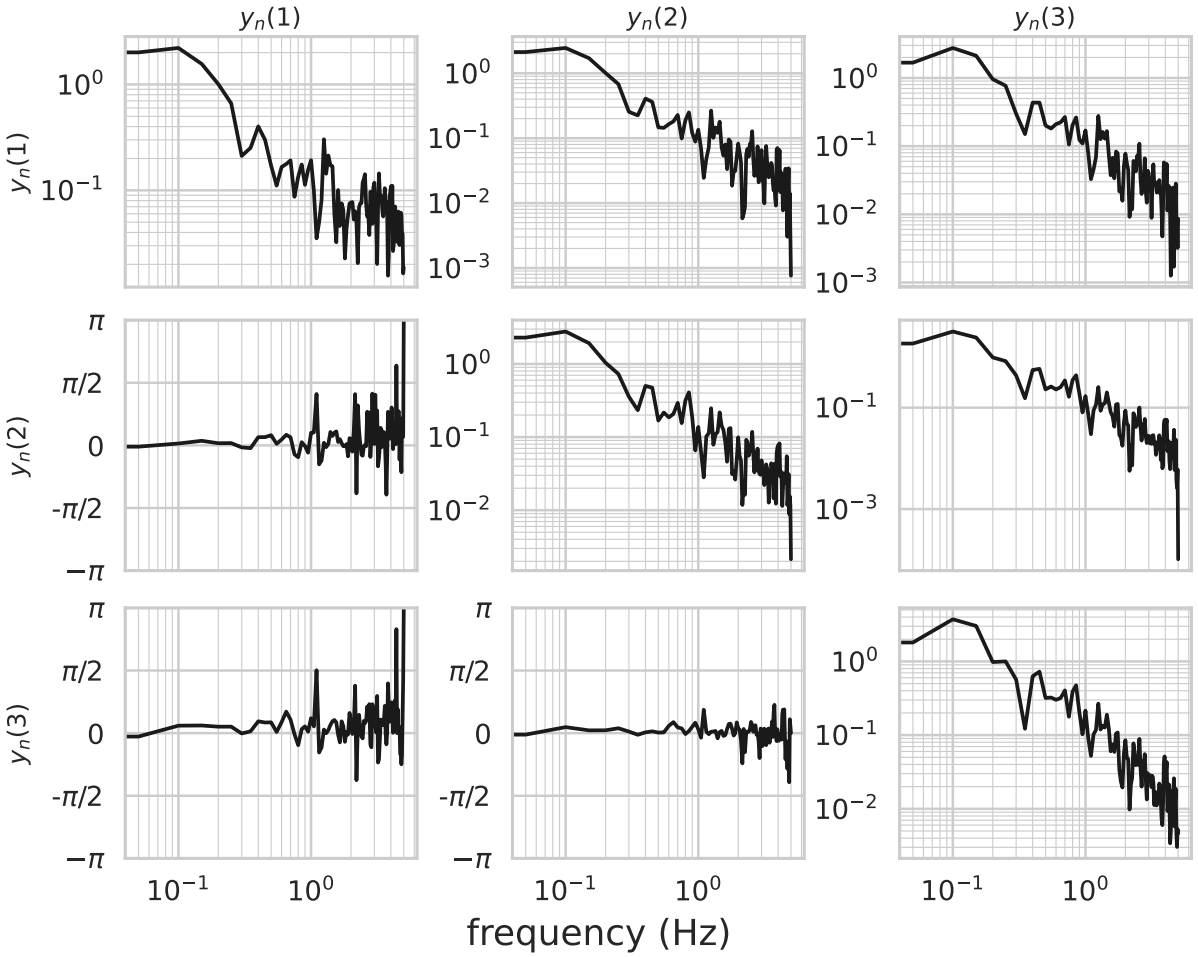


Figure 6.5: Amplitude and phase spectra of MAXI J1820 + 070.

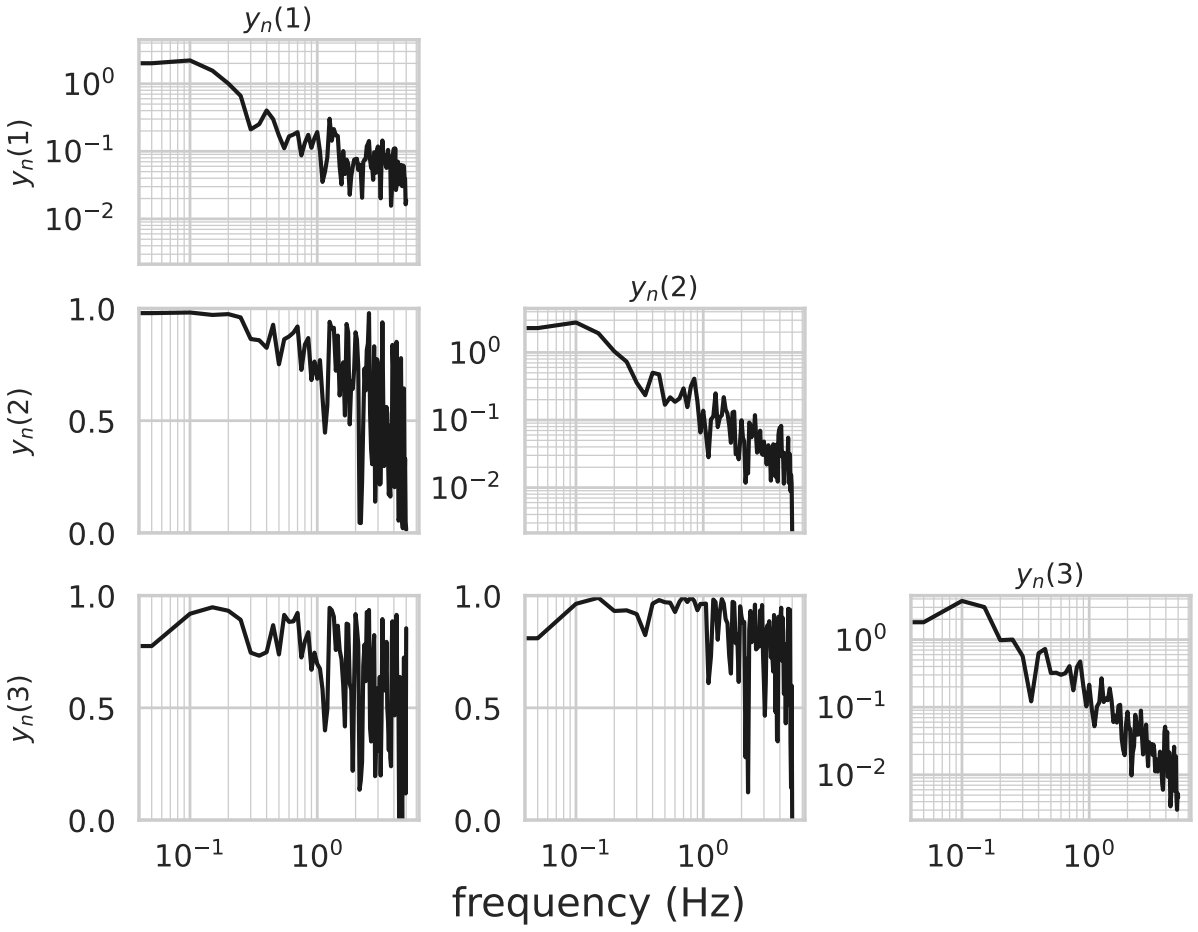


Figure 6.6: Sample coherencies of O106

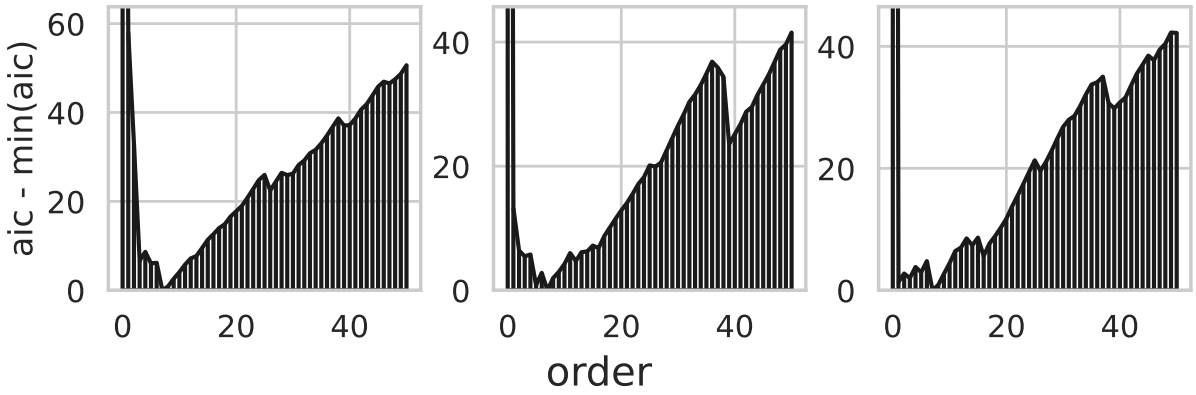


Figure 6.7: AIC as a function of AR orders for $y_n(1)$ (left), $y_n(2)$ (middle), and $y_n(3)$ (right). The minimum AIC is subtracted from the others to emphasize the differences.

6.2 Time Series Modeling I: Classical Approach

6.2.1 AR Model

6.2.1.1 Model Construction

As the classical approach, we start with the AR modeling of the light curve defined in § 6.1. Given that the AR model is univariate, this model ignores the correlation between different bands.

We first define the AR order based on the AIC. As shown in Figure 6.7, the optimum order for the light curve $y_n(1)$, $y_n(2)$, and $y_n(3)$ is 7, 7, and 8, respectively. For the optimum order, we derived the AR coefficients based on the least squares method using the `statsmodels.tsa.ar_model.AutoReg`[5] module.

Figure 6.8 presents the light curves of the observation, the AR model, and the residuals. The success of the AR modeling can be judged by examining that the residual follows the normal distribution, indicating that all useful information is incorporated into the modeling. We use the Q–Q plot here (§ 3.1.5).

In the Q–Q plot (Figure 6.9), the quantiles of the standard normal distribution and the residual time series are compared. The points are distributed along the diagonal, in particular around 0 line. However, the data away from 0 exhibits some departure. This shows the limitation of the AR modeling particularly in the low and high count rates.

6.2.1.2 Description Functions

Based on the constructed AR model, we derive the description functions of the univariate models.

Autocorrelation Function and Partial Autocorrelation Function The autocorrelation and partial autocorrelation functions are shown in Figure 6.10. In all bands, the

[5] https://www.statsmodels.org/dev/generated/statsmodels.tsa.ar_model.AutoReg.html

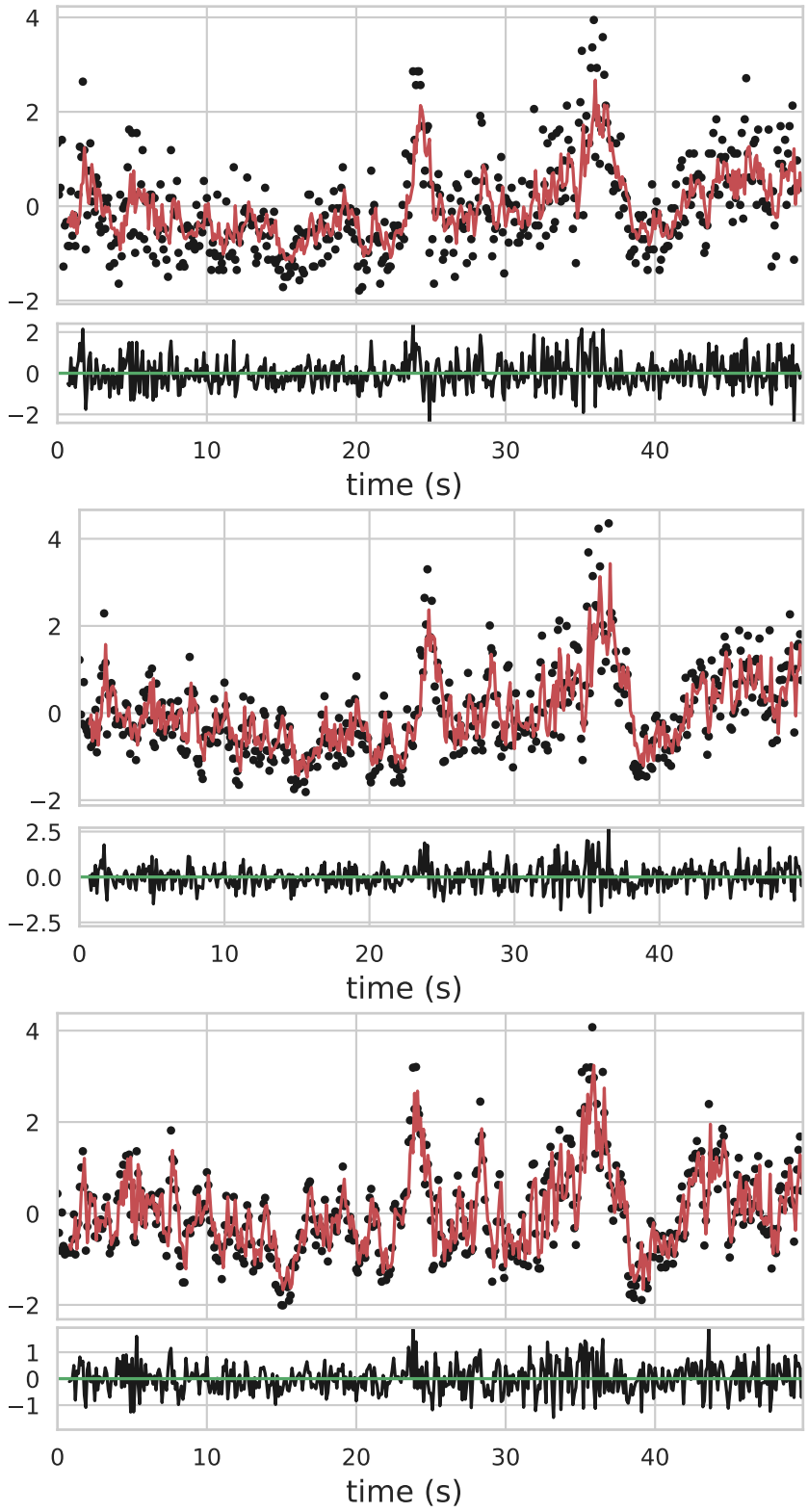


Figure 6.8: Observed light curves (black circles) and the AR model (red curves) and the residuals in the soft (0.5–2.0 keV; upper), middle (2.0–5.0 keV; middle), and hard (5.0–10.0 keV; lower) bands. Residuals are shown below each light curve.

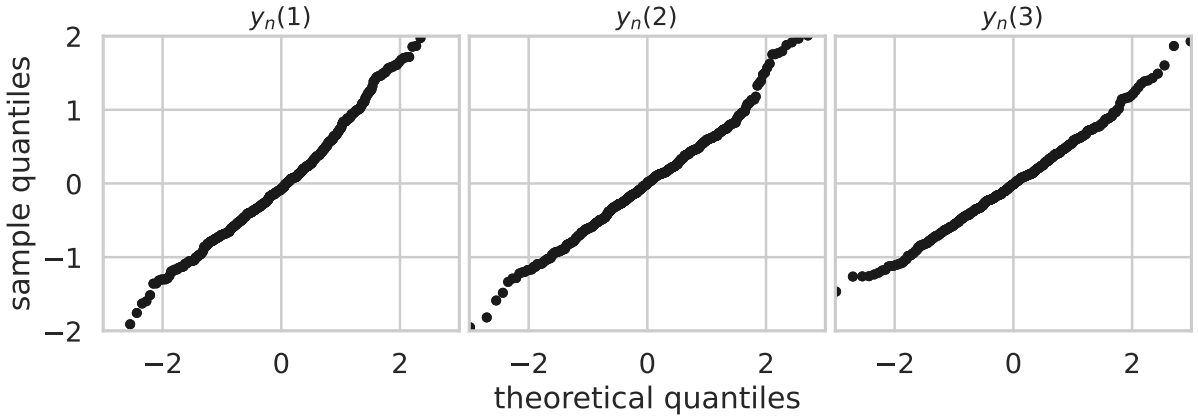


Figure 6.9: Q-Q plot of the residuals from the observation and estimated values of the AR model.

autocorrelation function decays rapidly and monotonically. The partial autocorrelation function exhibits a significant correlation around 0.1–0.2 s in all bands, but then rapidly drops beyond the point.

Power Spectrum We calculated the power spectrum in Figure 6.11. For all three bands, the power decreases as a function of increasing frequency. Up to ∼1.0 Hz, the slopes of all bands are similar. Beyond that point, the hard band ($y_n(1)$) shows higher power than the other bands. This suggests that variation of shorter timescales contributes more to the hard band.

Impulse Response Function The impulse response function provides insight into the noise impulse that is added at some point. We compute this function using the AR coefficients (Figure 6.12). The impulse response functions for all energy bands exhibit an exponential decay. The exponential decay timescales are shorter for the higher energy, implying that physical phenomena at higher energies settle faster than those at lower energies.

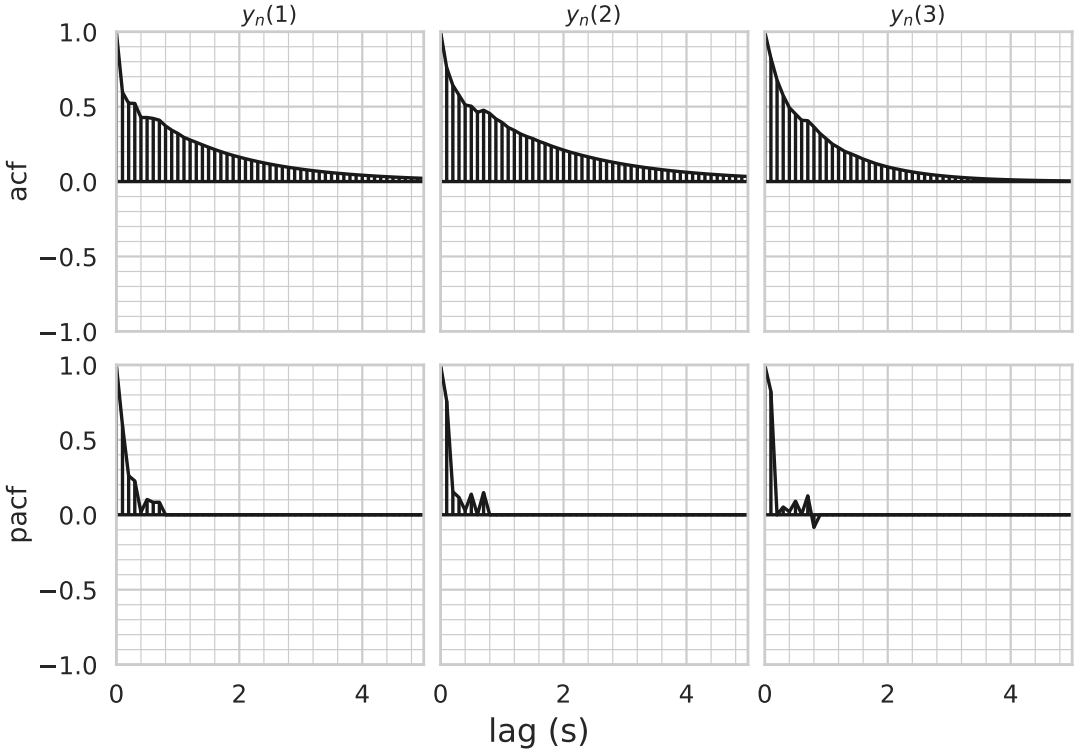


Figure 6.10: Autocorrelation (top) and partial autocorrelation (bottom) functions computed from the AR model.

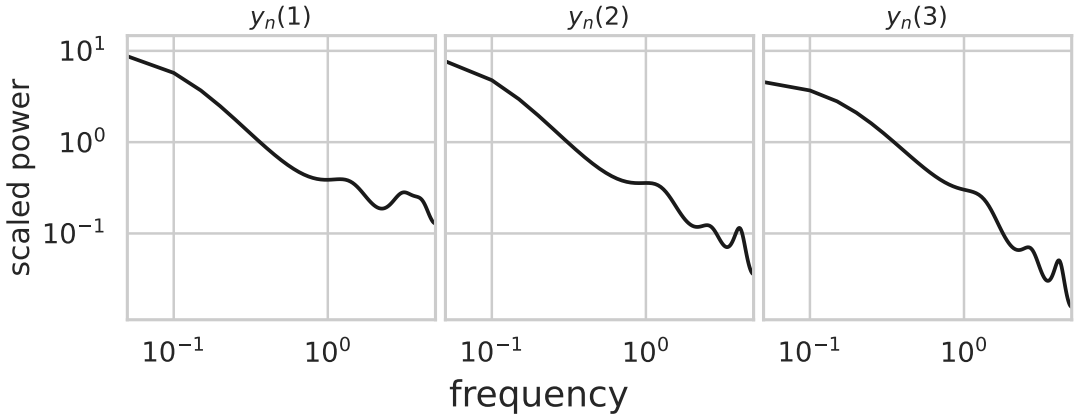


Figure 6.11: Power spectra computed from the AR coefficients.

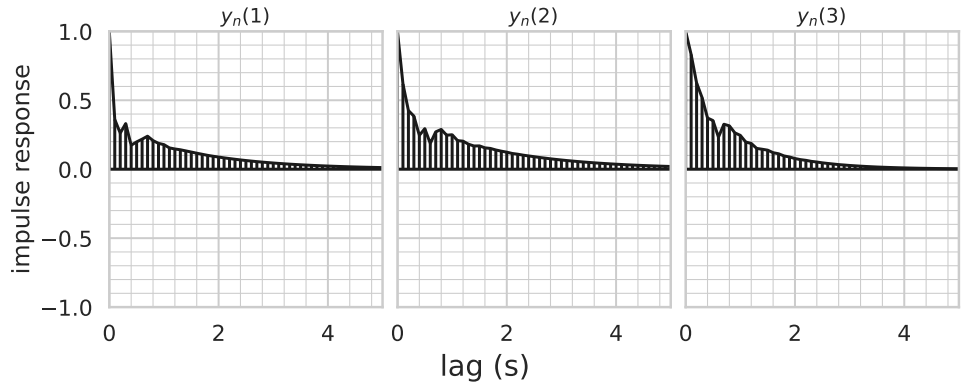


Figure 6.12: Impulse response functions estimated from the AR coefficients.

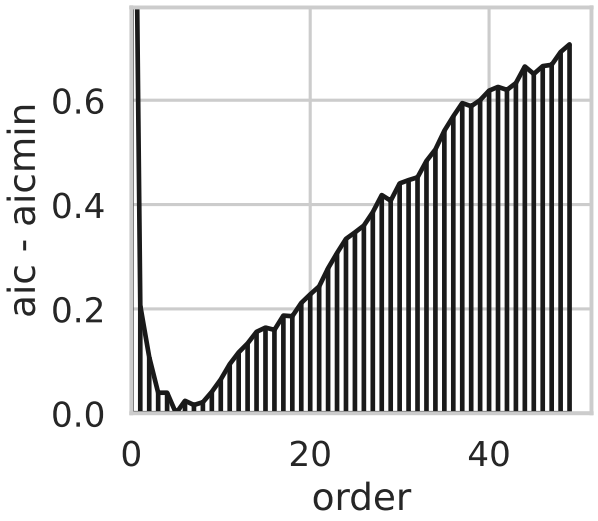


Figure 6.13: Order of the VAR model versus AIC. The lowest AICs are subtracted from the other ones to emphasize the differences.

6.2.2 VAR Model

6.2.2.1 Model Construction

Up to this point, we used the AR model to model the light curve of O106. The strong correlation observed among different energy bands in the data requires a multivariate time series model. To address this, we use the VAR model, which allows us to include correlation among different energy bands.

We also use the AIC to determine the appropriate order of the VAR model as well as the AR model (Figure 6.13). The AIC exhibits a monotonically decreasing trend until the 5th order, after which they start to increase. The behavior is smoother than the AR model (Figure 6.7), suggesting that the application of the multivariate time series model is more appropriate for the multiband light curve analysis of BHBs. Based on the AIC, we selected the 5th-order VAR model.

The VAR model (Figure 6.14) yielded a closer fit to the observed data for all energy bands in comparison to the AR model (Figure 6.8). As for the AR model, we used the Q–Q plot to assess the quality of the model by inspecting if the residuals follow a normal distribution (Figure 6.15). Some systematic deviations from the diagonal line are noticeable at both ends, but it is improved in comparison to the AR model (Figure 6.9).

6.2.2.2 Description Functions

Based on the constructed VAR model, we derive the description functions of the multivariate model.

Cross-Correlation Function We constructed the VAR cross-correlation function calculated from the estimated VAR coefficients (Figure 6.16). In comparison to the sample cross-correlation (Figure 6.3), the trend is similar but is more smooth, which is the benefit of the model-based VAR cross-correlation function.

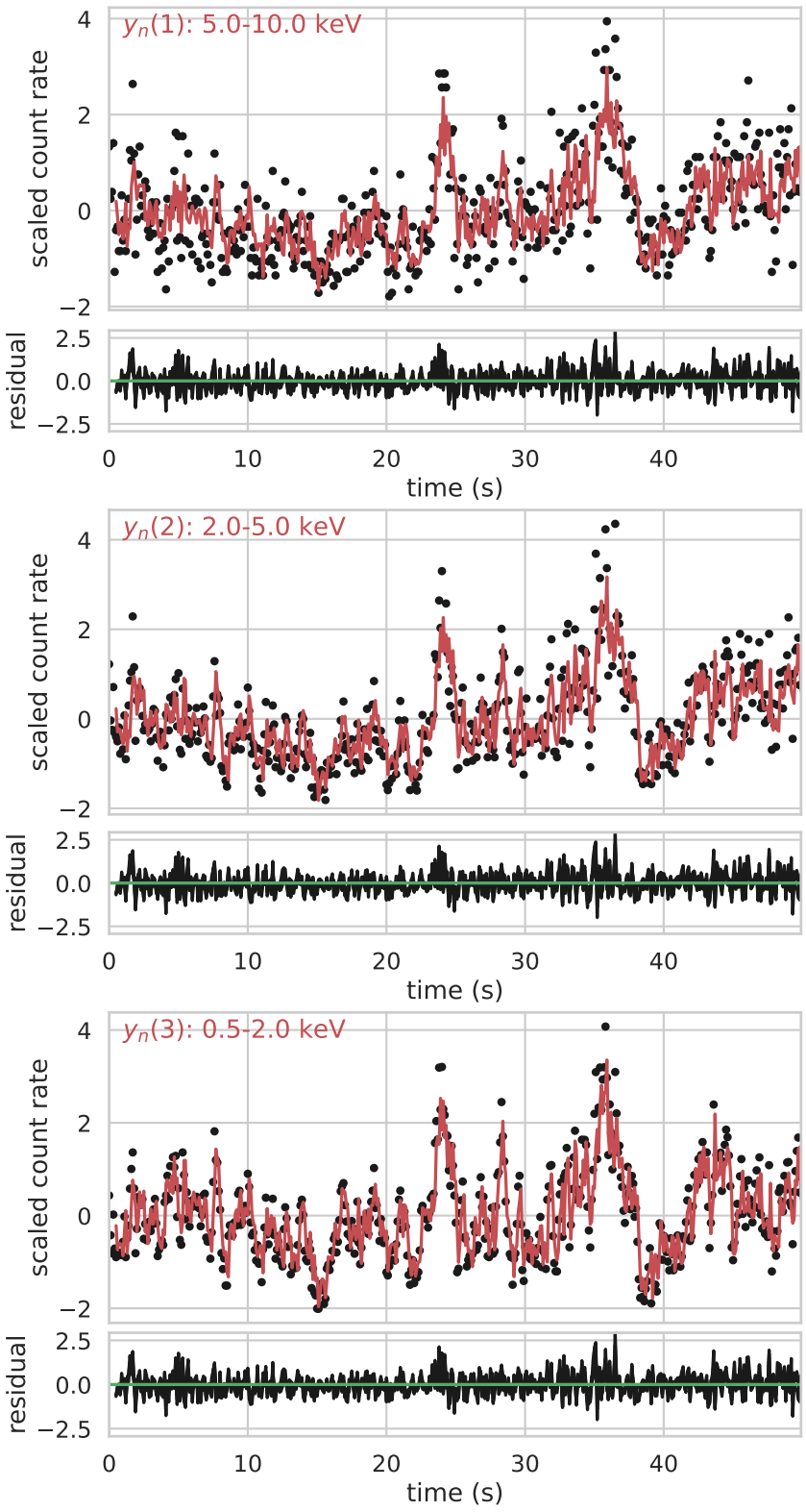


Figure 6.14: Observed values and VAR model curve with residuals. The order is the same as Figure 6.8.

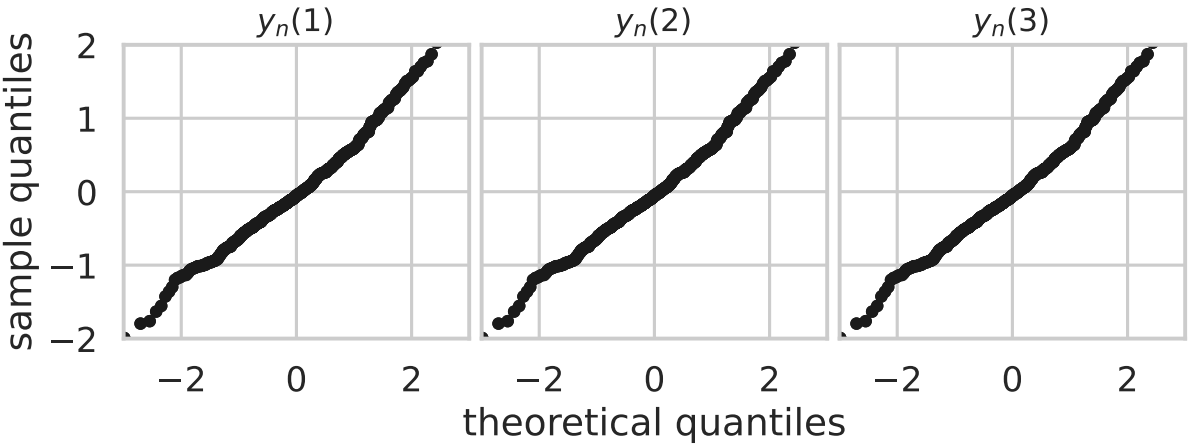


Figure 6.15: QQ-Plot of the residuals from the observation and estimated values of the VAR model.

Cross-Spectrum We computed the VAR cross-spectrum in Figure 6.17 using the VAR model, which shows the power spectra in the panels in the diagonal line, the amplitude spectra in the upper right panels, and the phase spectra in the lower left panels.

In the power spectra, all energy bands show that constant power is up to ~ 0.1 Hz, followed by a gradual decline. Excess powers at 2.0 Hz and 4.0 Hz are discernible, suggesting some periodic signals of these frequencies.

Regarding the amplitude spectra, a similar profile to that of the power spectra was obtained. Regarding the phase spectrum, which mostly arises from a common dominant component, it is almost zero in all frequencies for all combinations. Yet, intriguingly, the phase spectra between $y_n(1)$ and $y_n(2)$, as well as the one between $y_n(1)$ and $y_n(3)$, reveal positive signal around 4.0 Hz, indicating the presence of a phase shift. The positive shifts suggest a delay on the high-energy side, implying the propagation of the periodic signal from the low-energy side to the high-energy side. Importantly, this shift is absent in the phase spectrum between $y_n(2)$ and $y_n(3)$, implying that the propagation occurs from the component present in all bands to the one in $y_n(2)$ and $y_n(3)$.

Using Equation (3.46), coherencies were computed (Figure 6.18). While the coherencies between $y_n(1)$ and $y_n(2)$, as well as $y_n(1)$ and $y_n(3)$, follow a similar trend, the latter consistently shows smaller values. As we move toward higher frequencies, the coherency tends to decrease with local increases around 2.0 Hz and 4.0 Hz at which periodic signal was observed in the power spectrum. On the other hand, the coherency between $y_n(2)$ and $y_n(3)$ shows a different trend.

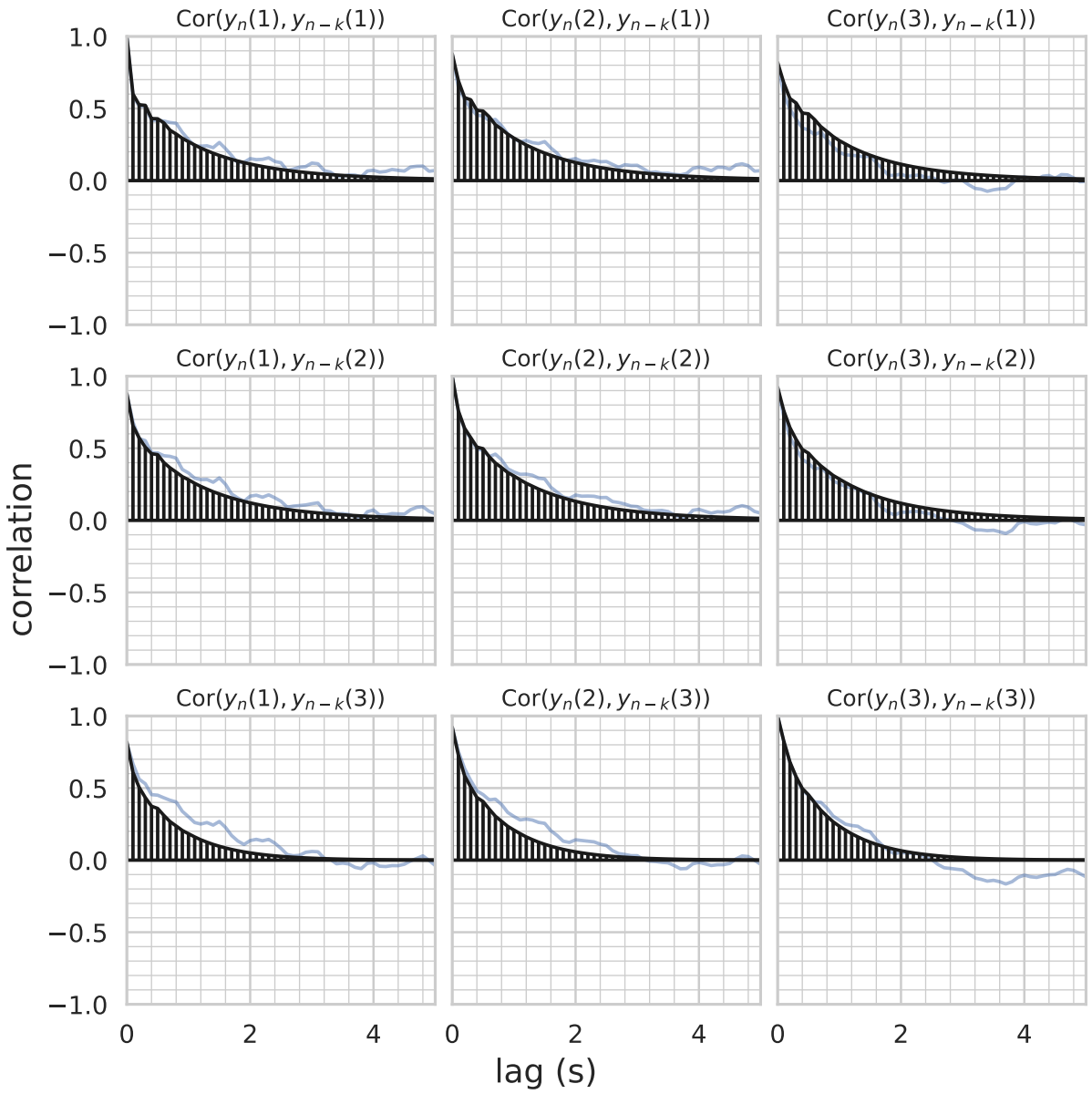


Figure 6.16: VAR cross-correlation functions. The diagonal panels are the autocorrelation functions for the energy bands. The blue curves are the sample cross-correlation functions shown in Figure 6.3.

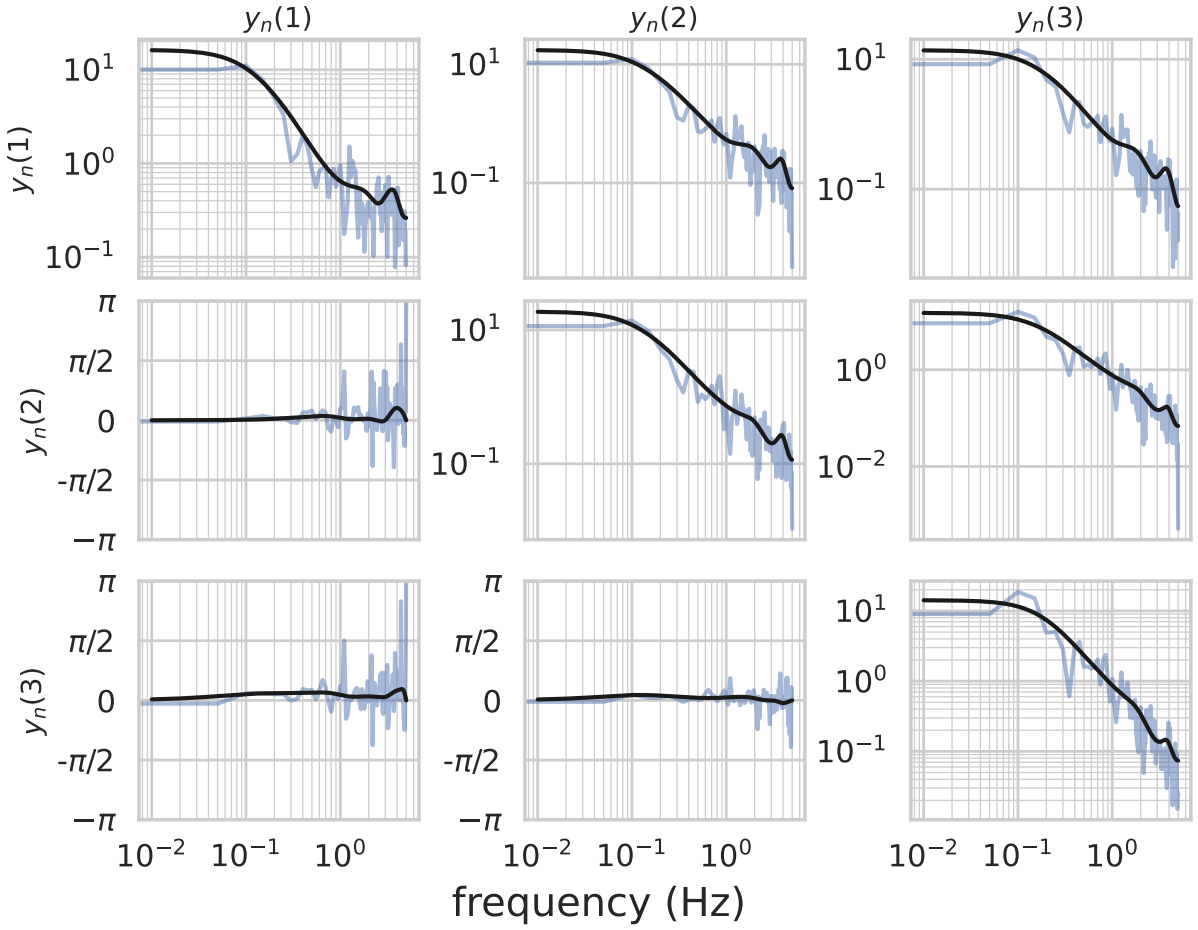


Figure 6.17: VAR amplitude and phase spectra. The diagonal panels are the VAR power spectra. The panels above the diagonal line are the VAR amplitude spectra and the ones below are the VAR phase spectra. In the VAR phase spectra, the positive phase is defined as the delay of the band on the x-axis. The blue curves are counterparts computed from the sample cross-spectra.

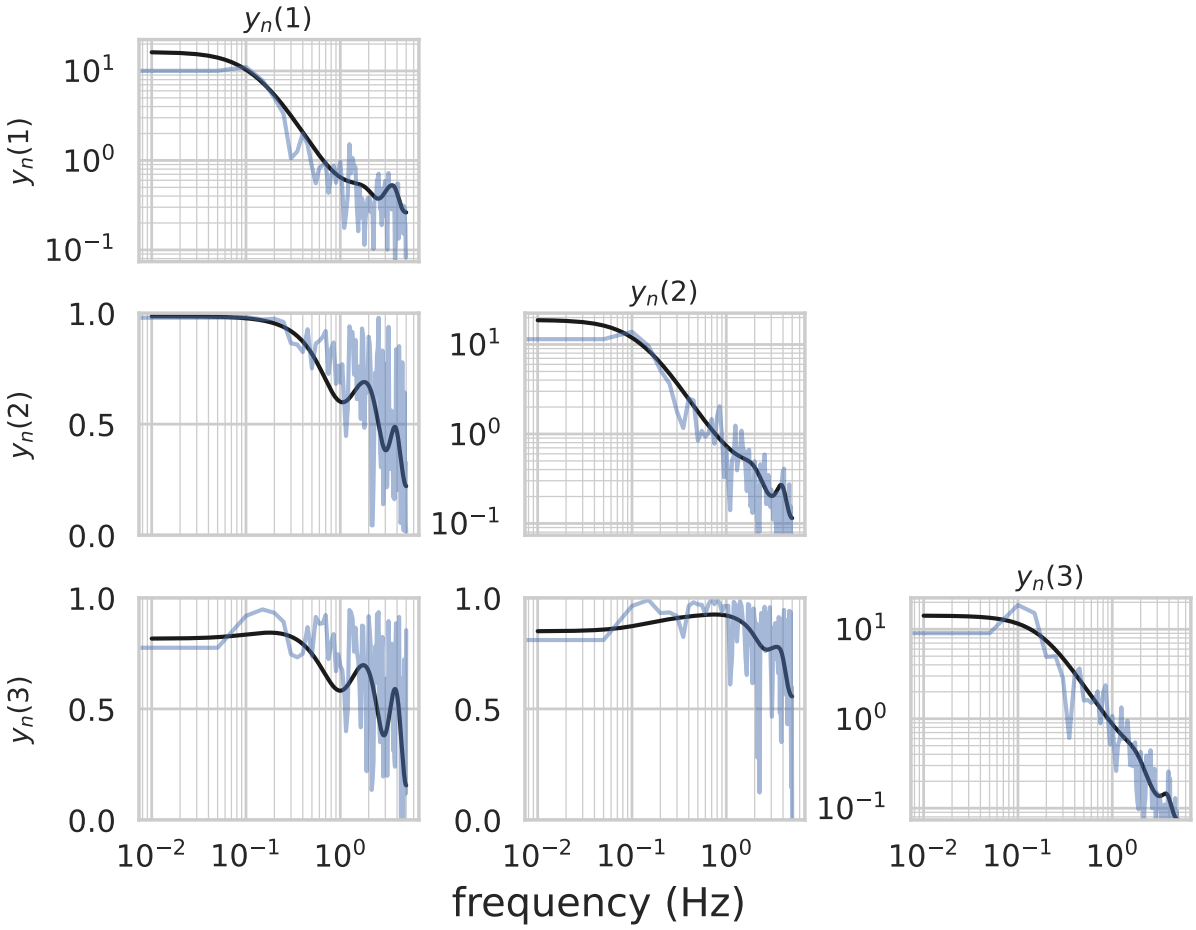


Figure 6.18: Coherencies computed from the VAR coefficients. The diagonal panels are power spectra. The blue curves are the periodograms and coherencies.

6.3 Time Series Modeling II: State-Space Model

6.3.1 Definition of Variables

The largest distinction between classical modeling (§ 6.2) and a state-space modeling is that the observation and state (or latent) variables are separated in the latter. We thus need to define the state variables as well as the observation variables. Amongst the state-space modeling, we use the linear Gaussian state-space model (LGSSM). This requires two things. One is that the noise should behave Gaussian, thus we need to set the minimum number of counts per bin for generating the time series data. The other is that the observation variables should be expressed as a linear combination of the state variables. Therefore, they need to be in the same unit; i.e., counts s^{-1}. If we want to use physical values of BHBs, such as the black hole mass and the inner radius of the accretion disk, as state variables, we need to use a "non-linear" state-space model, which is out of the scope of this section. We also consider the number of variables; it is often considered that a robust model is obtained when the number of the observation variables is larger than the number of the state variables. Considering these points, we define the observation and state variables below.

Observation Variables As in the classical modeling (§ 6.2), the observation variables are the light curves of different energy bands. The generated time series (§ 6.1.1) meets the requirement of $\gtrsim$30 counts bin^{-1} to ensure Gaussian approximation of the noise. In addition, we want to satisfy that the number of the observation variables is larger than the state variables. Fortunately, the soft-band light curve is richer in photon statistics than the hard-band light curve, which drives the bin size. Therefore, we decided to divide the soft band further into three; i.e., 0.5–1.0, 1.0–1.5, and 1.5–2.0 keV. A total of five bands are thus defined. The multiband light curves are shown in Figure 6.19.

As we redefined the energy bands and generated new time series data, we repeated the inspection and pre-processing (§ 6.1.2). We performed a stationarity test using the ADF test. The p-values and the order of the AR model used for the ADF test, which minimizes the AIC, are presented in Table 6.3. Although the number of bands is increased from 3 to 5, the p-values remain below 5 %, resulting in the rejection of the null hypothesis that the time series follows a unit root AR model for all bands. We also pre-process the data for normalization to have a mean of 0 and variance of 1 as in the classical approach.

State Variables As we need to have the same physical units for the observation and state variables, we use the count rate of the physical spectral component (Comptonization, disk blackbody, and soft excess) bolometric emission as the state variables. This appears a reasonable choice as observed variables can be expressed by a linear combination of the state variables.

The number of state variables is a key to a successful application of the state-space model. We used three state variables for two reasons. One reason is from the spectral

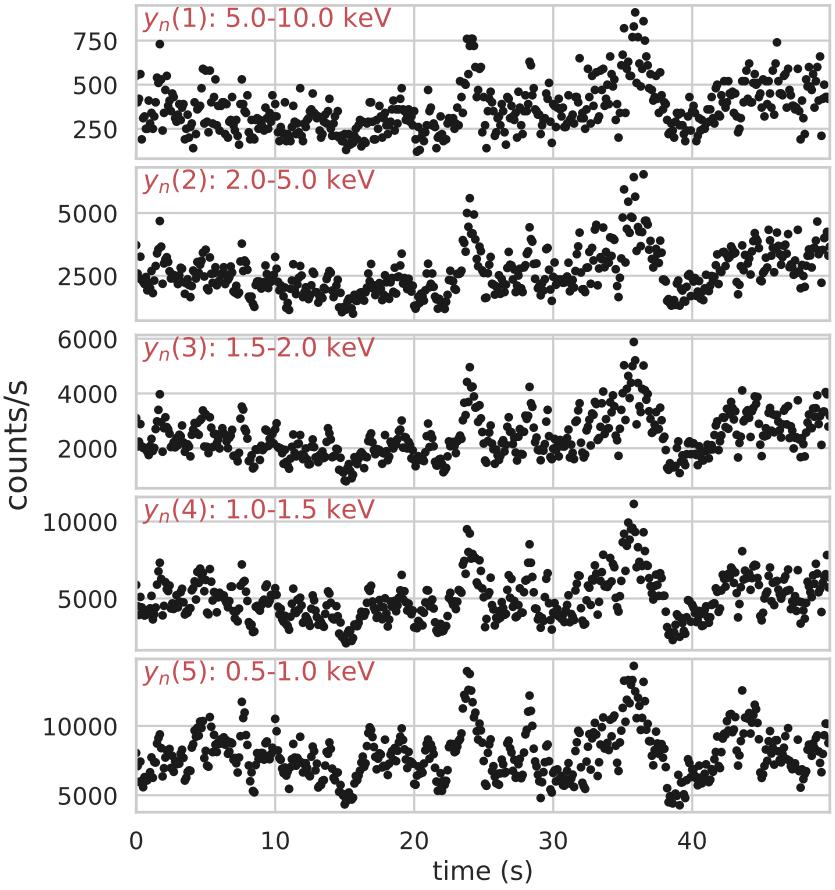


Figure 6.19: Multivand light curves used for the linear state-space modeling.

Table 6.3: Band definition used for LGSSM and details for the stationarity test.

Band	Lower edge (keV)	Higher edge (keV)	p-value	AR order
1	5.0	10.0	0.002	6
2	2.0	5.0	0.005	6
3	1.5	2.0	0.001	8
4	1.0	1.5	0.001	6
5	0.5	1.0	0.000	7

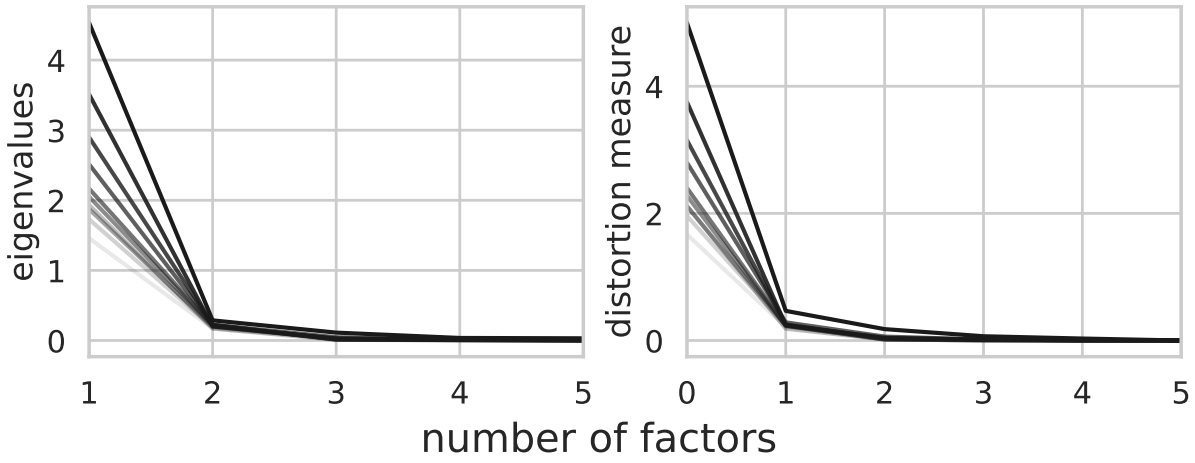


Figure 6.20: Eigenvalue spectrum (left) and distortion measure (right). The different lag results (up to 10) are shown with different levels of transparency. More transparent curves indicate larger lag values.

fitting result, which requires three broad components (Figure 6.1). The other is from the principal component analysis (PCA) of the observation variables, investigating the number of eigenvalues, or independent components, to explain the time series data. Bishop (2006) discusses the approach for determining the number of state variables using probabilistic PCA within the Bayesian framework. Based on this idea, Pena and Box (1987) presented how we apply the method to the time series using the cross-covariance function (Equation (3.33)) instead of the cross-covariance matrix.

Following their method, we computed the cross-covariance function of the five observation variables with a lag of up to 10. We derived the eigenvalues and eigenvectors. Figure 6.20 shows the eigenvalue spectrum (left), in which the eigenvalues are sorted in descending order, and the distortion measure (right), in which the sum of the truncated eigenvalues (sum of leftover eigenvalues when the larger ones are discarded) are shown. In both panels, from the fourth lag onward, the metrics are nearly zero, indicating that the number of significant components to explain the observation variable is 3. This is in agreement with the first reasoning.

6.3.2 Model Specification

Observation Model Next, we relate the observation variables and the state variables with a linear combination, which is called the observation model. We represent the state variables as a vector $x_n = [x_n(1)\ x_n(2)\ x_n(3)]^T$ and the observation variables as a vector $y_n = [y_n(1)\ y_n(2)\ y_n(3)\ y_n(4)\ y_n(5)]^T$, and the relation between them as

$$y_n = H_1 x_n + w_n, \tag{6.1}$$

Table 6.4: Contribution of the physical components to the energy bands.

band	Energy range (keV)	$x_n(1)$	$x_n(2)$	$x_n(3)$
$y_n(1)$	10.0–5.0	✓	-	-
$y_n(2)$	5.0–2.0	✓	✓	-
$y_n(3)$	2.0–1.5	✓	✓	✓
$y_n(4)$	1.5–1.0	✓	✓	✓
$y_n(5)$	1.0–0.5	✓	✓	✓

where

$$H_1 = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \\ h_{41} & h_{42} & h_{43} \\ h_{51} & h_{52} & h_{53} \end{bmatrix}. \tag{6.2}$$

and

$$w_n \sim \mathcal{N}(0, L_R L_R^T), \quad L_R = \begin{bmatrix} r_6 & 0 & 0 & 0 & 0 \\ r_{11} & r_{12} & 0 & 0 & 0 \\ r_{15} & r_{14} & r_{13} & 0 & 0 \\ r_{10} & r_9 & r_8 & r_7 & 0 \\ r_5 & r_4 & r_3 & r_2 & r_1 \end{bmatrix}. \tag{6.3}$$

The order of subscripts of L_R is in a clockwise spiral for programmatical convenience (See `tensorflow_probability.math.fill_triangular` function[6]). Here, h_{ij} means the contributions from $x(i)$ to $y(j)$. In other words, the observation matrix represents the spectral mixture of the physical components in each energy band.

The observation matrix H_1 can be rotated arbitrarily, thus the solution cannot be obtained uniquely (§ 3.4.1.3). We thus need to restrict the problem to some extent to break the degeneracy. We utilize the fact that the soft excess and the disk black body component contribute only to the softer bands (Figure 6.21). The contributions of the physical components are summarized in Table 6.4. We set $h_{ij} = 0$ if $x_n(i)$ does not contribute to $y_n(j)$. The observation matrix H_1 is now revised as follows:

$$H_1 = \begin{bmatrix} h_4 & 0 & 0 \\ h_6 & h_5 & 0 \\ h_3 & h_2 & h_1 \\ h_7 & h_8 & h_9 \\ h_{10} & h_{11} & h_{12} \end{bmatrix} \tag{6.4}$$

The order of subscripts is in a clockwise spiral.

[6] https://www.tensorflow.org/probability/api_docs/python/tfp/math/fill_triangular.html

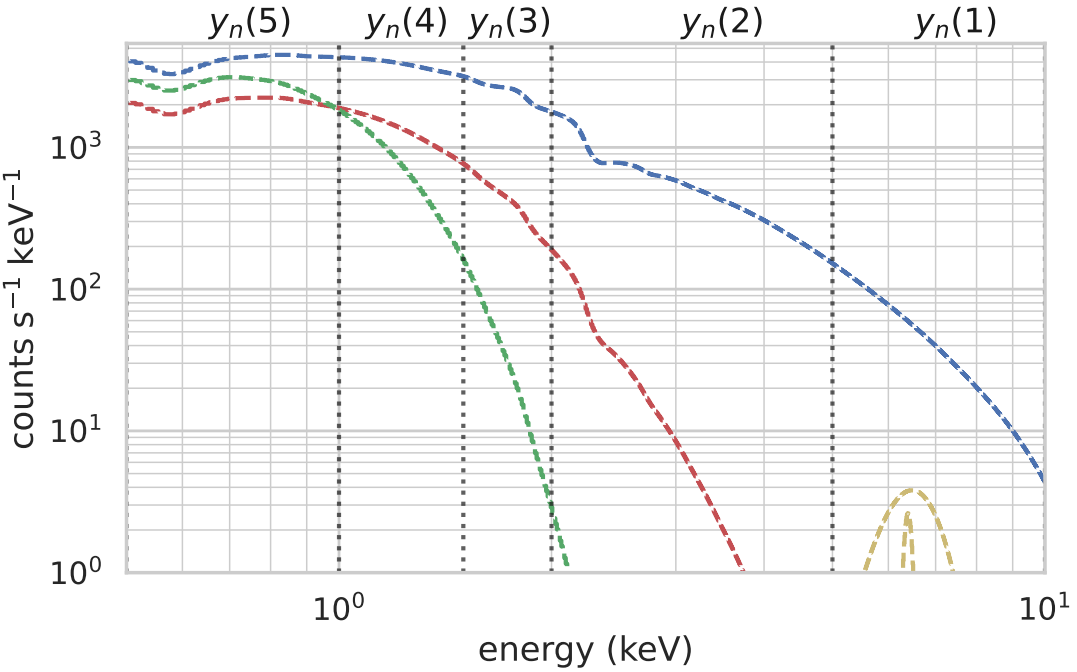


Figure 6.21: Energy spectrum of O106 and the best-fit model with the energy band edges. The same as Figure 6.21, but only the spectrum models are shown since the purpose is to emphasize the energy band edges denoted by the black dotted lines.

System Model Then, we construct a model to describe how the state variables change. We use the VAR model as in the classical approach (§ 6.2.2), so that we can investigate the relations among different physical (Comptonization, disk blackbody, and soft excess) components. For the choice of the VAR order, we used the AIC in the classical approach (§ 6.2.2). In the LGSSM, however, the parameter increases as the VAR order increases. They are correlated with each other, thus the penalty term proportional to the number of the parameters, as in the AIC, does not work well. As the oversimplified model like the first-order VAR model is not suited for capturing the local features of the power spectra, we adopt the second-order VAR model here.

The system model is now expressed as

$$x_n = A_1 x_{n-1} + A_2 x_{n-2} + v_n, \tag{6.5}$$

where

$$A_i = \begin{bmatrix} a_{i,11} & a_{i,12} & a_{i,13} \\ a_{i,21} & a_{i,22} & a_{i,23} \\ a_{i,31} & a_{i,32} & a_{i,33} \end{bmatrix}, \tag{6.6}$$

and

$$v_n \sim \mathcal{N}(0, Q), \quad Q = I. \tag{6.7}$$

State-Space Model To estimate parameters using the state-space model, we need to transform the given observation and system models into the state-space form. Looking

at the state-space form of the VAR model in Equation (3.86), we can express both the observation and system models as

$$\begin{aligned} y_n &= \begin{bmatrix} H_1 & O \end{bmatrix} \begin{bmatrix} x_n \\ x_{n-1} \end{bmatrix} + w_n \\ \begin{bmatrix} x_n \\ x_{n-1} \end{bmatrix} &= \begin{bmatrix} A_1 & A_2 \\ I & O \end{bmatrix} \begin{bmatrix} x_{n-1} \\ x_{n-2} \end{bmatrix} + \begin{bmatrix} v_n \\ O \end{bmatrix}, \end{aligned} \tag{6.8}$$

When all the elements of the observation and system models are written down, the observation model becomes as follows:

$$y_n = \begin{bmatrix} h_4 & 0 & 0 & 0 & 0 & 0 \\ h_6 & h_5 & 0 & 0 & 0 & 0 \\ h_3 & h_2 & h_1 & 0 & 0 & 0 \\ h_7 & h_8 & h_9 & 0 & 0 & 0 \\ h_{10} & h_{11} & h_{12} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_n(1) \\ x_n(2) \\ x_n(3) \\ x_{n-1}(1) \\ x_{n-1}(2) \\ x_{n-1}(3) \end{bmatrix} + \begin{bmatrix} w_n(1) \\ w_n(2) \\ w_n(3) \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{6.9}$$

The system model becomes as follows:

$$\begin{bmatrix} x_n(1) \\ x_n(2) \\ x_n(3) \\ x_{n-1}(1) \\ x_{n-1}(2) \\ x_{n-1}(3) \end{bmatrix} = \begin{bmatrix} a_{1,11} & a_{1,12} & a_{1,13} & a_{2,11} & a_{2,12} & a_{2,13} \\ a_{1,21} & a_{1,22} & a_{1,23} & a_{2,21} & a_{2,22} & a_{2,23} \\ a_{1,31} & a_{1,32} & a_{1,33} & a_{2,31} & a_{2,32} & a_{2,33} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{n-1}(1) \\ x_{n-1}(2) \\ x_{n-1}(3) \\ x_{n-2}(1) \\ x_{n-2}(2) \\ x_{n-2}(3) \end{bmatrix} + \begin{bmatrix} v_n(1) \\ v_n(2) \\ v_n(3) \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{6.10}$$

The total parameters are listed in Table 6.5.

6.3.3 Model Inference

As described in § 3.1, we take the Bayesian inference approach here. We define the prior distributions and the sampling of the inference below. We denote the dimension of the state variable as $p = 3$, that of the observation variables as $q = 5$, and the order of the VAR model as $m = 2$.

6.3.3.1 Prior Distributions

To obtain the posterior distributions, it is necessary to specify the prior distributions in the Bayesian inference. We discuss how we choose the prior distributions (Table 6.5) considering the characteristics of each parameter.

VAR Matrix The VAR coefficient matrices are denoted as A_i, where i ranges from 1 to m, in Equation (6.6). Each order of the VAR coefficient matrix consists of $p \times p$ elements, denoted as $a_{i,jk}$, with $j, k = 1, 2, \cdots, p$. The total number of elements in

Table 6.5: Parameters, their names, and prior distributions. $\mathcal{N}(\cdot)$, HalfCauchy$(\cdot)$, and LKJ$(\cdot)$ represent normal, half-Cauchy, and LKJ distributions, respectively.

Symbol	Parameter Name	Prior Distribution
$a_{i,jk}$	VAR coefficients	$\mathcal{N}(0, 1)$
h_i	Observation matrix elements	HalfCauchy$(0, 5)$
σ_i	Standard deviation of observation noise	HalfCauchy$(0, 5)$
r_i	Covariance matrix elements of observation noise	LKJ(3)

the VAR coefficient matrices is $m \times p \times p = 18$. These elements need to be chosen to satisfy the stationary condition of the VAR model, ensuring that all eigenvalues of the characteristic equation lie outside the unit circle (§ 3.3.2).

Ideally, a prior distribution meeting this condition would be set. In practice, however, calculating such a prior distribution is challenging and computationally demanding. In this study, we simply use a normal distribution as the prior distribution. If the stationary condition is not met, the likelihood should be very small as the estimated state values exhibit explosive increases or decreases, which contradicts the observed time series that are stationary. Therefore, parameters violating the stationary condition are expected to be rejected during the MCMC sampling process.

Alternatively, the Minnesota prior distribution could be adopted, which is often used for the prior distribution of VAR coefficient matrices (Lütkepohl, 2005). While this prior does not strictly enforce the stationary condition, it assumes sparsity in the coefficients, meaning that many coefficients will be close to zero. This assumption is generally reasonable, and having a sparse coefficient matrix has the advantage of making the interpretation of the estimated values easier. We did not take this approach for simplicity.

Observation Matrix The observation matrix is denoted as H_1 in Equation (6.4). As the matrix is restricted (§ 3.1), it has $pq - p(p-1)/2 = 12$ parameters. We can assume that the state variables (physical components) contribute only additively to the observation variables (multiband light curves), we should assume that the parameters are positive. We thus choose the half-Cauchy distribution as the prior distribution for these parameters with scale parameters of 5, which is a common weak prior recommended by Stan Development Team (2024) [7].

Observation Noise (standard deviation and covariance matrix) We employ the Half-Cauchy distribution for the prior distribution of the standard deviation of observation noise. Additionally, for the prior distribution of the covariance matrix of observation noise, we adopt the Lewandowski-Kurowicka-Joe (LKJ) distribution (Lewandowski et al., 2009), which is commonly used as the prior distribution of the correlation matrices in recent Bayesian modeling. It can be combined with a diagonal matrix to generate the covariance matrix and used for the prior distribution of the covariance matrix. For

[7] https://github.com/stan-dev/stan/wiki/Prior-Choice-Recommendations

example, using a correlation matrix

$$R = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \tag{6.11}$$

and a diagonal matrix

$$\sigma = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix}, \tag{6.12}$$

we can compute the positive definite matrix as follows:

$$\Sigma = \sigma R \sigma = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix} \tag{6.13}$$

The conjugate prior distribution for a covariance matrix is the inverse Wishart distribution, but the LKJ distribution is computationally stable and efficient for sampling. The probabilistic programming language `Stan` recommends using this distribution for the prior distribution of a covariance matrix (Stan Development Team, 2024). The parameter of the LKJ distribution governs the degree of correlation, with one representing a non-informative prior and a large number indicating a strong prior.

Since we have no prior information for the standard deviations of the observation noise, we assign the half-Cauchy distribution as the prior distribution for them, setting the scale parameters to 5. For the LKJ distribution, we assume independence of the observation noise term, achieved by ensuring the correlation matrix has zero values except for non-diagonal elements. Consequently, we set the LKJ distribution parameter to 3.

6.3.3.2 Sampling

We use MCMC sampling for the Bayesian parameter estimation. It proposes candidate parameters based on the information from the previous iteration and only accepts them if certain conditions are met. We use the NUTS, a variant of the Hamiltonian Monte Carlo method, for the sampling (§ 3.1.3). The estimation conditions include a burn-in of 1000 iterations and 5000 iterations for sampling.

Sampling values from the normal and half-Cauchy distributions are trivial (Table 6.5). For the sampling of the LKJ distribution, we use `tensorflow_probability.distibutions.CholeckyLKJ` [8], which samples from the Cholesky decomposed LKJ distributions. Representing the yield lower triangular matrix as

$$L_P \equiv \begin{bmatrix} \rho_6 & 0 & 0 & 0 & 0 \\ \rho_{11} & \rho_{12} & 0 & 0 & 0 \\ \rho_{15} & \rho_{14} & \rho_{13} & 0 & 0 \\ \rho_{10} & \rho_9 & \rho_8 & \rho_7 & 0 \\ \rho_5 & \rho_4 & \rho_3 & \rho_2 & \rho_1 \end{bmatrix}, \tag{6.14}$$

[8] https://www.tensorflow.org/probability/api_docs/python/tfp/distributions/CholeskyLKJ.html

the correlation matrix is

$$
\begin{aligned}
P &= L_P L_P^T \\
&= \begin{bmatrix} \rho_6 & 0 & 0 & 0 & 0 \\ \rho_{11} & \rho_{12} & 0 & 0 & 0 \\ \rho_{15} & \rho_{14} & \rho_{13} & 0 & 0 \\ \rho_{10} & \rho_9 & \rho_8 & \rho_7 & 0 \\ \rho_5 & \rho_4 & \rho_3 & \rho_2 & \rho_1 \end{bmatrix} \begin{bmatrix} \rho_6 & \rho_{11} & \rho_{15} & \rho_{10} & \rho_5 \\ 0 & \rho_{12} & \rho_{14} & \rho_9 & \rho_4 \\ 0 & 0 & \rho_{13} & \rho_8 & \rho_3 \\ 0 & 0 & 0 & \rho_7 & \rho_2 \\ 0 & 0 & 0 & 0 & \rho_1 \end{bmatrix} \\
&= \begin{bmatrix} 1 & \rho'_{11} & \rho'_{15} & \rho'_{10} & \rho'_5 \\ \rho'_{11} & 1 & \rho'_{14} & \rho'_9 & \rho'_4 \\ \rho'_{15} & \rho'_{14} & 1 & \rho'_8 & \rho'_3 \\ \rho'_{10} & \rho'_9 & \rho'_8 & 1 & \rho'_2 \\ \rho'_5 & \rho'_4 & \rho'_3 & \rho'_2 & 1 \end{bmatrix}.
\end{aligned} \tag{6.15}
$$

By multiplying L_P and a variance diagonal matrix, the triangular matrix of the covariance matrix can be calculated as

$$
\begin{aligned}
L_R &= \begin{bmatrix} \sigma_{r1} & 0 & 0 & 0 & 0 \\ 0 & \sigma_{r2} & 0 & 0 & 0 \\ 0 & 0 & \sigma_{r3} & 0 & 0 \\ 0 & 0 & 0 & \sigma_{r4} & 0 \\ 0 & 0 & 0 & 0 & \sigma_{r5} \end{bmatrix} \begin{bmatrix} \rho_6 & 0 & 0 & 0 & 0 \\ \rho_{11} & \rho_{12} & 0 & 0 & 0 \\ \rho_{15} & \rho_{14} & \rho_{13} & 0 & 0 \\ \rho_{10} & \rho_9 & \rho_8 & \rho_7 & 0 \\ \rho_5 & \rho_4 & \rho_3 & \rho_2 & \rho_1 \end{bmatrix} \\
&= \begin{bmatrix} \sigma_{r1}\rho_6 & 0 & 0 & 0 & 0 \\ \sigma_{r2}\rho_{11} & \sigma_{r2}\rho_{12} & 0 & 0 & 0 \\ \sigma_{r3}\rho_{15} & \sigma_{r3}\rho_{14} & \sigma_{r3}\rho_{13} & 0 & 0 \\ \sigma_{r4}\rho_{10} & \sigma_{r4}\rho_9 & \sigma_{r4}\rho_8 & \sigma_{r4}\rho_7 & 0 \\ \sigma_{r5}\rho_5 & \sigma_{r5}\rho_4 & \sigma_{r5}\rho_3 & \sigma_{r5}\rho_2 & \sigma_{r5}\rho_1 \end{bmatrix} \\
&\equiv \begin{bmatrix} r_6 & 0 & 0 & 0 & 0 \\ r_{11} & r_{12} & 0 & 0 & 0 \\ r_{15} & r_{14} & r_{13} & 0 & 0 \\ r_{10} & r_9 & r_8 & r_7 & 0 \\ r_5 & r_4 & r_3 & r_2 & r_1 \end{bmatrix}.
\end{aligned} \tag{6.16}
$$

We use $r_1, \ldots, r_{15}$ as the samples for the covariance matrix of the observation noise.

6.3.4 Results and Inspection

6.3.4.1 Results

Figure 6.22 shows the posterior distribution of the Bayesian inference. For each sample, using the Kalman filter smoothing, we can construct the light curves of the observation variables and compare them to the observed data (Figure 6.23). The smoothed observation variables describe the data very well in all bands. In addition, we can also construct the light curves of the state variables, or the physical components (Figure 6.24). This

demonstrates the advantage of using the latent variable in the state-space model; no such light curve can be obtained through classical approaches.

Several interesting observations are made. In the smoothed state variable light curves (Figure 6.24), each state variable, or the physical component, exhibits distinct variations in each component. For instance, $x_n(1)$ (Comptonization) and $x_n(2)$ (disk blackbody) show mostly high-frequency fluctuations, whereas $x_n(3)$ (soft excess) also shows low-frequency fluctuations. The most notable feature is found at 46 s. This low-frequency variation is only found in $x_n(3)$. This is consistent with the variation in the observed data (Figure 6.24), in which the low-frequency variation is increasingly seen toward lower energy bands.

Conversely, this indicates that we can do spectral decomposition based on the time series analysis. Because the system model assumes a VAR model with a zero mean, we can derive the time-averaged energy spectrum from the observation matrix. The reproduced count rates, denoted as c_{ij}, are given by

$$\begin{bmatrix} c_{1j} \\ c_{2j} \\ c_{3j} \\ c_{4j} \\ c_{5j} \end{bmatrix} = \begin{bmatrix} \sigma_1 & 0 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_3 & 0 & 0 \\ 0 & 0 & 0 & \sigma_4 & 0 \\ 0 & 0 & 0 & 0 & \sigma_5 \end{bmatrix} \begin{bmatrix} h_{1j} \\ h_{2j} \\ h_{3j} \\ h_{4j} \\ h_{5j} \end{bmatrix} + \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \end{bmatrix}. \tag{6.17}$$

Here, $j = 1, 2, 3$, and μ_i and σ_i represent the mean and standard deviation of the observed multiband light curve, respectively. The resultant time-averaged energy spectrum (Figure 6.25) closely resembles the energy spectrum model obtained through energy spectral fitting (Figure 6.1). It is important to note that this spectral decomposition is achieved solely from the time series modeling only by assuming the number of state variables of 3 and the observation matrix form in Equation (6.4).

6.3.4.2 Inspection

Statioarity Check We used the normal distribution as the prior distribution of the VAR matrix elements. This ignores the stationary conditions to be met, but we expected that the MCMC process would reject the non-stationary samples. Indeed, in the posterior distribution (Figure 6.22), most of the coefficients are sampled within the range of -1 to $+1$, which implies the model stationarity is likely satisfied as a result.

Gaussian Check We used the Q–Q plot in Figure 6.26 to assess whether the differences between the observation variables and the observed values adhere to a normal distribution. A particular improvement was made from the classical approach (Figures 6.9, 6.15) at low count rates by separating the observation and state variables. The deviation at high count rates still remains in the LGSSM. This may be a signature that is difficult to model with linear models.

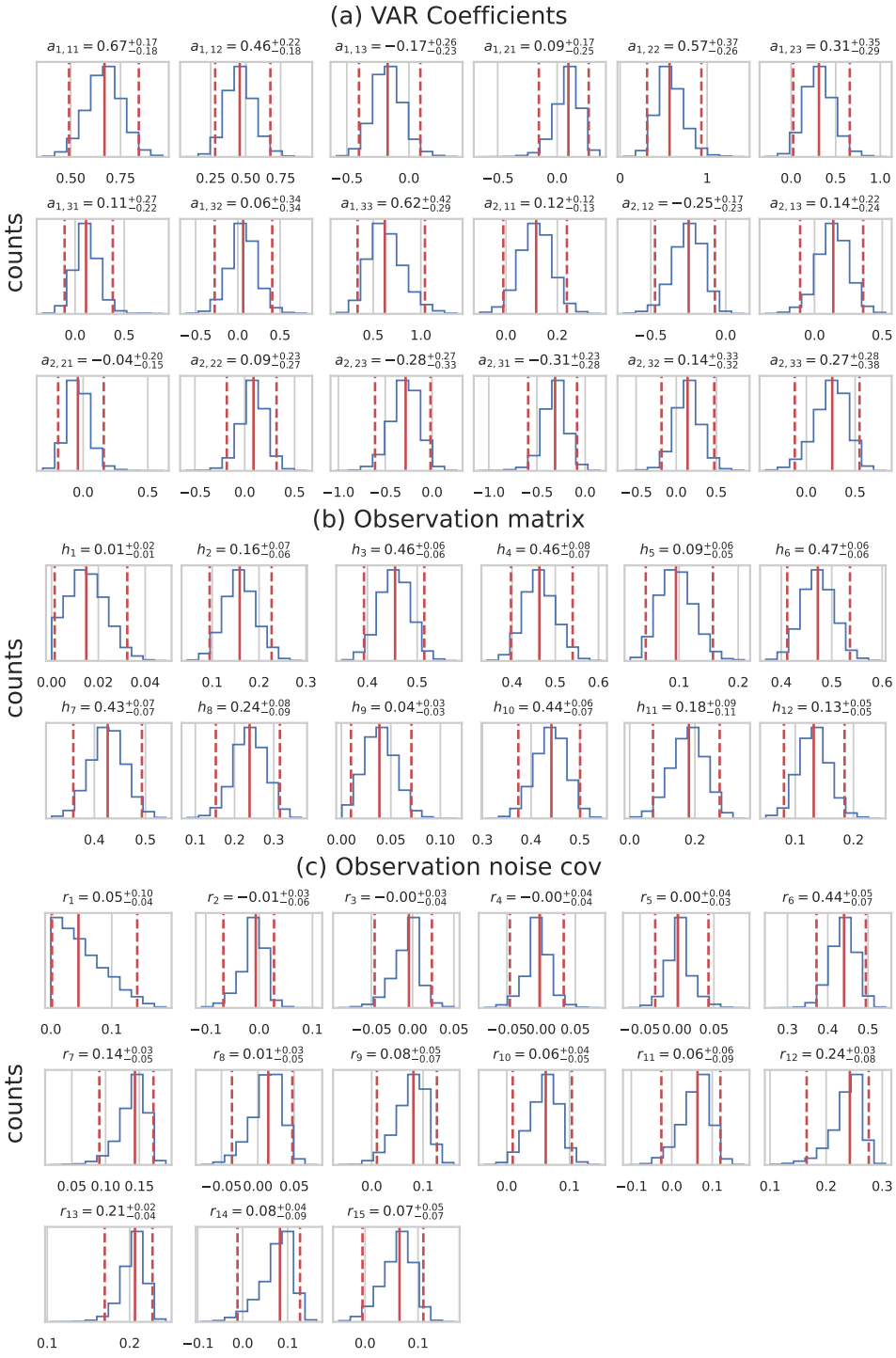


Figure 6.22: Posterior distributions estimated by the MCMC. Figure (a), (b), and (c) are the VAR coefficients, the observation matrix, and the elements of the variance-covariance matrix, respectively. The order of the panels follows, Equations (6.6), (6.4), and (6.3), respectively. The numbers above each panel are the median of a distribution and 2.5 % and 97.5 % quantiles.

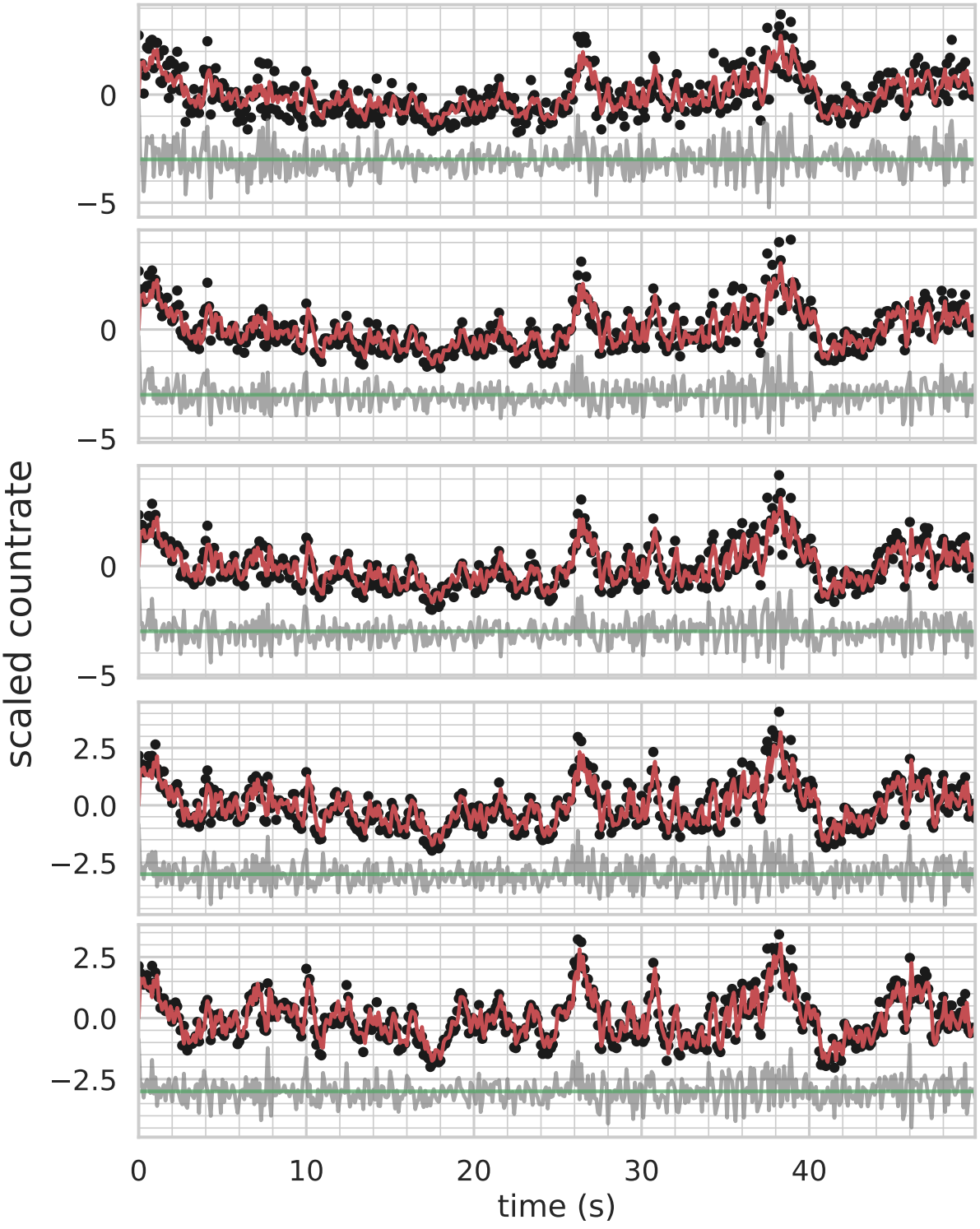


Figure 6.23: Estimated light curve of the observation variables (red curves) and the observed data (black points) at the top, and the residuals at the bottom shifted downward by −3 for clarity. The ranges of the observation variables are shown in the 2.5 % and 97.5 % quantiles, which are not visible due to the narrow width compared to the variation of the curve.

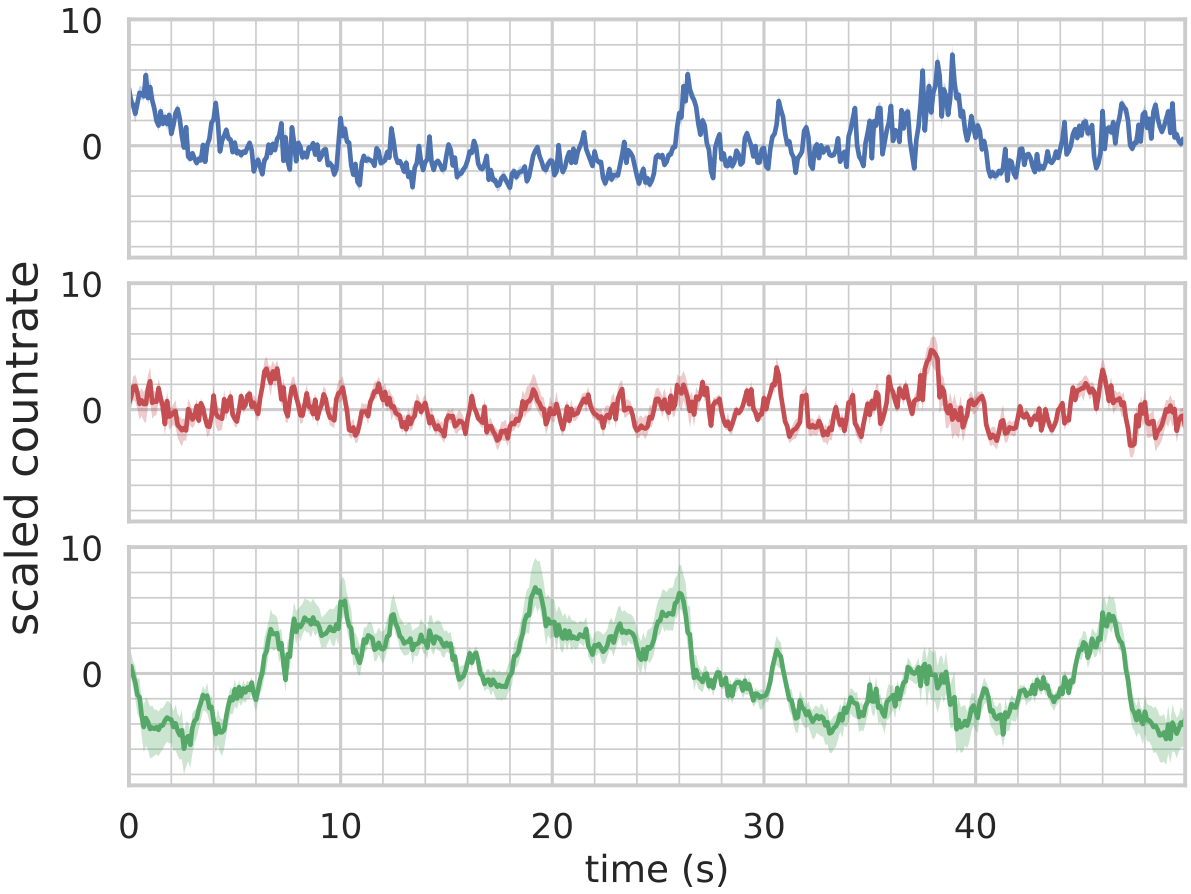


Figure 6.24: Estimated light curve of the state variables, or the physical components. The Comptonization, disk blackbody, and soft excess components are illustrated by the blue, red, and green curves, respectively. The range of the state variables is shown with the 2.5 % and 97.5 % quantiles.

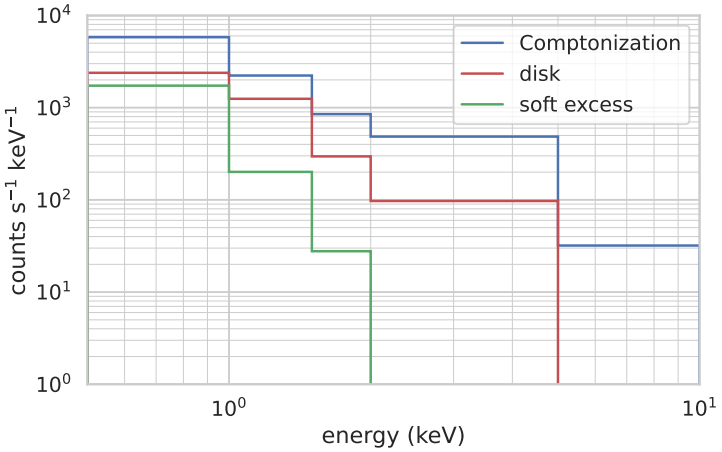


Figure 6.25: Reproduced energy spectrum from the observation matrix.

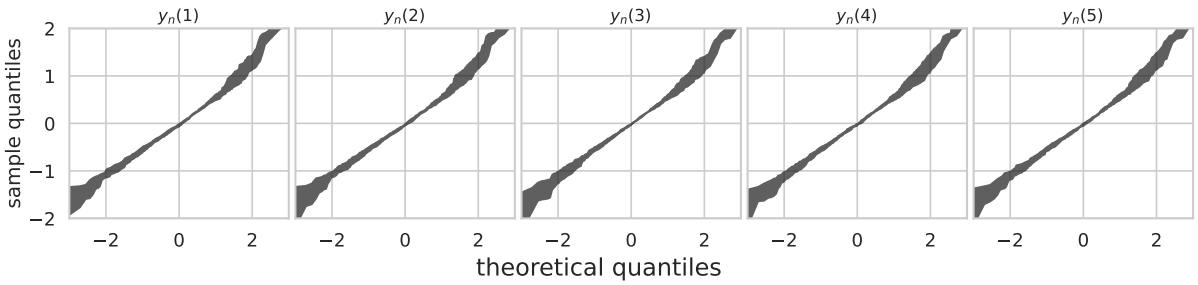


Figure 6.26: Q–Q plot of residuals between the observation variables and the observed data. We calculated the plot for each MCMC sample, highlighting the width between the 2.5 % and 97.5 % quantiles.

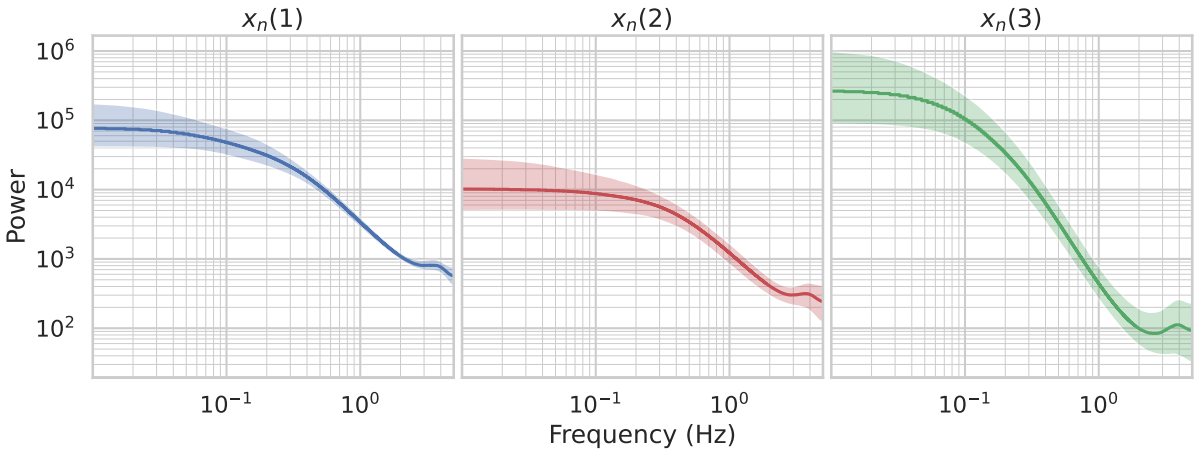


Figure 6.27: VAR power spectra of the latent components. The VAR power spectra of the $x_n(1)$, $x_n(2)$, and $x_n(3)$ are represented by the blue, red, and green curves, respectively.

6.3.5 Description Functions

Finally, we explore the multivariate description functions. The description functions presented here are not sample description functions (§ 6.1) nor the description functions of the observation variables (§ 6.2), but the description functions of the state variables; i.e., relations among the physical components free from the spectral mixture.

6.3.5.1 Cross-Spectrum

For cross-spectra of the variability for the energy spectrum component, it is more reasonable to use the scale of real data when comparing contribution ratios. This can be achieved by using Equation (6.17).

Figure 6.27 shows the power spectra of energy spectrum components converted to the scale of the softest band (0.5-1.0 keV), where all components contribute. The power spectra of all components show a similar feature overall; flat power in frequencies below a break, a sloped power above the break, and some excess power at high frequencies. The breaking frequencies are different among different components; $x_n(1)$, $x_n(2)$, and $x_n(3)$ exhibit breaking frequencies around 0.05, 0.1, and 0.05 Hz, respectively. Therefore, the contribution of $x_n(3)$ is larger at low frequencies. This is consistent with the observed data that low-frequency variations are more visible in the low-energy range (Figure 6.23).

For the excess power at the high frequencies, all physical components exhibit signals around 4.0 Hz. It is improbable that each component independently generates the periodic signal at the same frequency. Therefore, we consider that the periodic variation in one component propagates to the other components.

The periodic signal observed in the VAR power spectra of energy bands (Figure 6.17) is not identifiable in the VAR power spectra of latent components. This could be attributed to the VAR power spectra of energy bands picking up characteristics of noise. In Bayesian estimation, calculating samples from the posterior distribution may result in such probabilistic signals being buried within the width of the probability distribution.

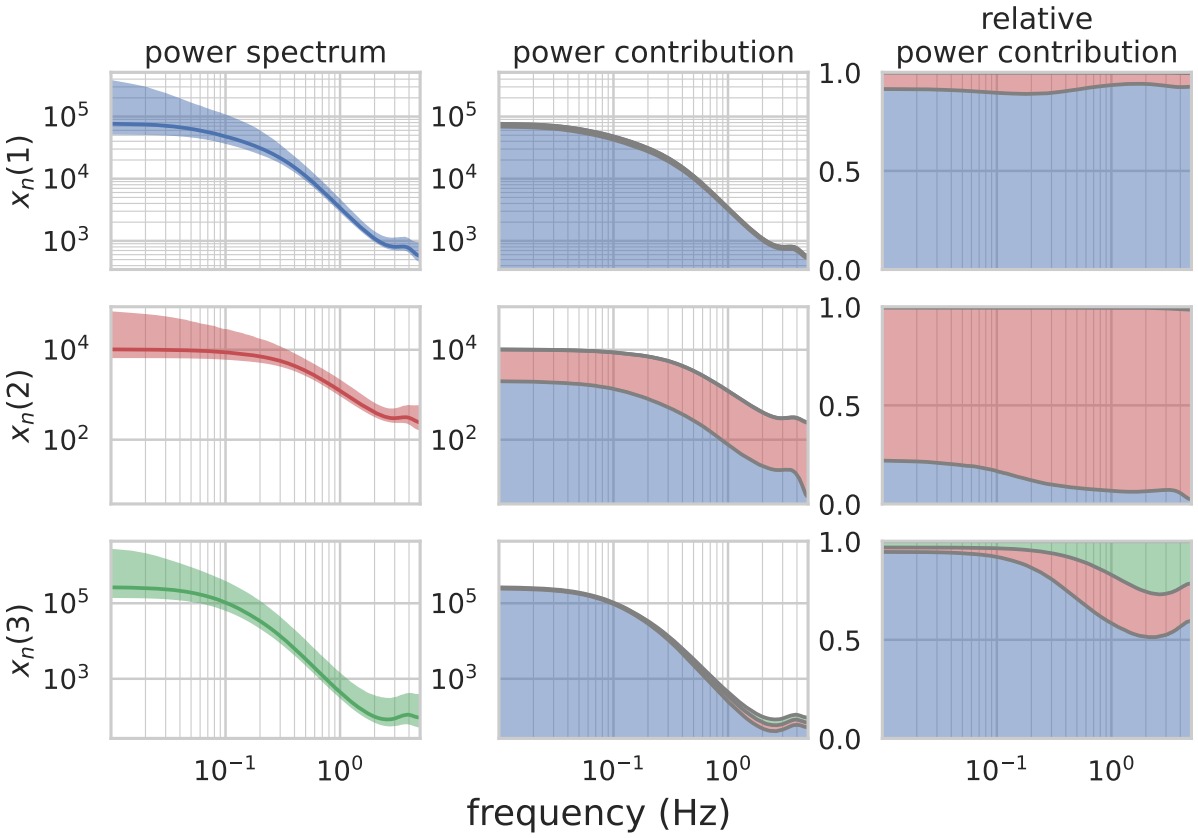


Figure 6.28: Power contributions and relative power contributions estimated from the VAR matrix. The power contributions and relative power contributions of $x_n(1)$, $x_n(2)$, and $x_n(3)$ are listed from top to bottom. The blue, red, and green areas represent the contributions from $x_n(1)$, $x_n(2)$, and $x_n(3)$ to each component.

Next, we calculated the power contribution and relative power contribution (Figure 6.28). The shaded areas are the widths of the posterior distribution with 95 % quantiles. For the relative power contributions of $x_n(1)$ and $x_n(2)$, we see that the contribution by themselves is the most dominant in all frequencies. On the other hand, for $x_n(3)$, the contribution by $x_n(1)$ is most dominant, which decreases as the frequency increases, where the contributions by $x_n(2)$ and $x_n(3)$ increase. For the relative power contribution of $x_n(2)$, the contribution of $x_n(1)$ decreases as the frequency increases. The relative power contribution gives little information about the periodic signal at 4 Hz as there is no significant feature around them.

6.3.5.2 Cross-Correlation Function

We calculate the VAR cross-correlation functions for $[x_n(1)\ x_n(2)\ x_n(3)]^T$ based on the VAR coefficients (Figure 6.29). The distinctive differences in breaking frequencies observed in the power spectrum are evident in their respective autocorrelation functions. Notably, $\mathrm{Cor}(x_n(3), x_{n-k}(3))$ exhibits the broadest width, while $\mathrm{Cor}(x_n(1), x_{n-k}(1))$ and $\mathrm{Cor}(x_{n-k}(2), x_{n-k}(2))$ have narrower widths. An explicit delayed signal is observed in

$\mathrm{Cor}(x_n(1), x_{n-k}(2))$ with a peak at 0.1 s. Similarly, $\mathrm{Cor}(x_n(3), x_{n-k}(2))$ shows a time-lagged signal with a peak at 0.2 s. Both signals originate from $x_n(2)$, but the latter is delayed by an additional 0.1 s. This suggests the signal sequence propagation as $x_n(2) \to x_n(1) \to x_n(3)$. However, in $\mathrm{Cor}(x_n(3), x_{n-k}(1))$, the propagation from $x_n(1)$ to $x_n(3)$ is obscured by a strong anti-correlation, where $x_n(3)$ exhibits an opposite fluctuation about 0.5 s after the variation in $x_n(1)$. This robust anti-correlation explains the dominance of $x_n(1)$ in the power contribution of $x_n(3)$.

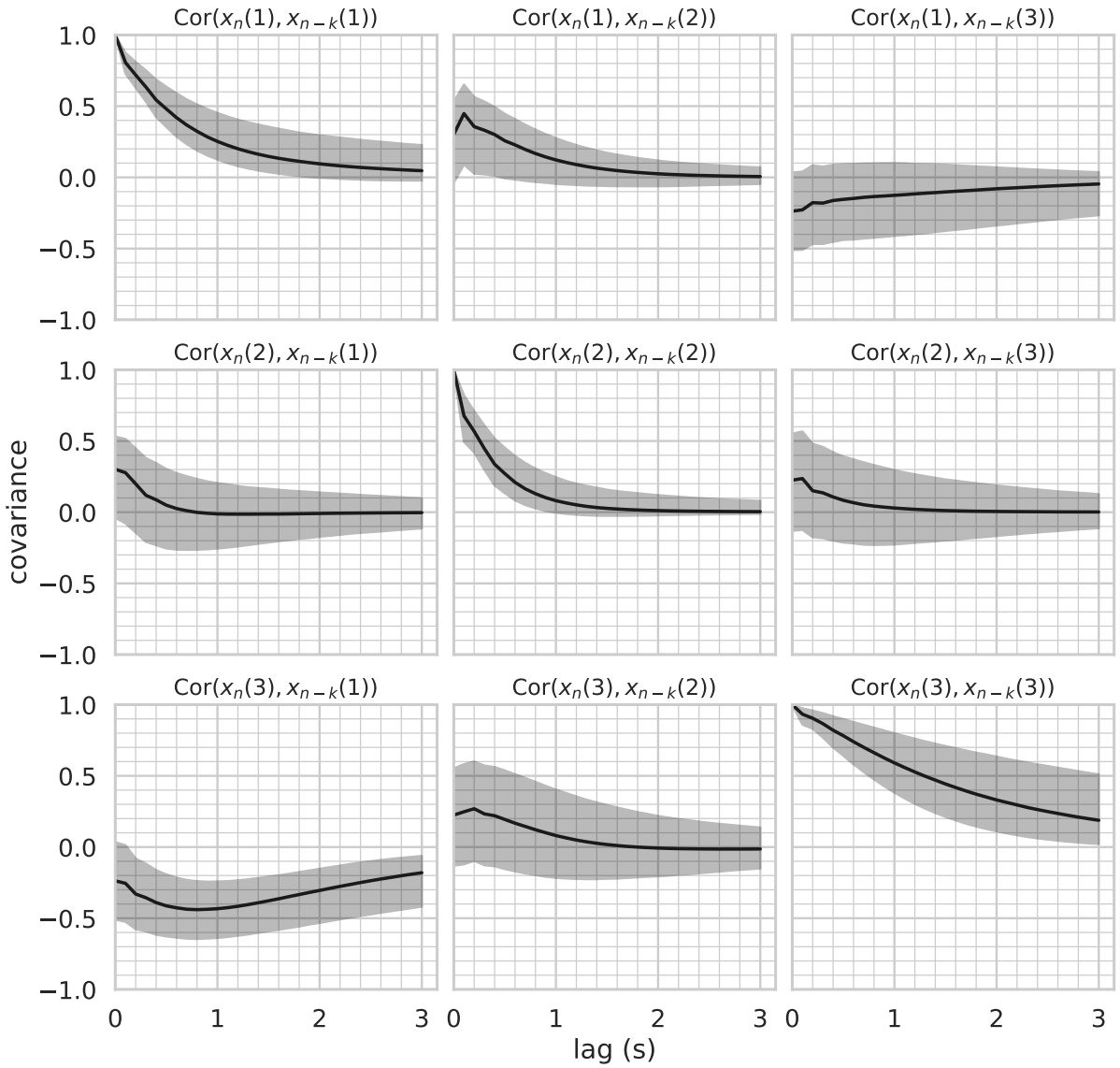


Figure 6.29: VAR cross-correlation function for the energy spectrum components. The variable i corresponds to the x-axis and the variable j corresponds to the y-axis in Equation (3.35). Thus, the diagonal panels are autocorrelation functions. The light curve of the Comptonization, disk, and soft excess are denoted as $x_n(1)$, $x_n(2)$, and $x_n(1)$, respectively.

Chapter 7

Discussion

Contents

7.1 Physical Interpretation of the Results

The traditional light curve analysis employed in previous work led to contradicting conclusions from the same data set on the geometry of the accretion disk around a black hole (§ 5.2). This is due in large part to the spectral dilution in the light curve of selected energy bands (§ 4.2.2). We employed statistical modeling to overcome this and obtained the description functions among different spectral components (§ 6.3). Using the new results, we discuss the relationship among the different spectral components.

7.1.1 Time Scales of Variation

We start with the power spectrum of the different spectral components (Figure 6.27). This is a significant improvement from the power spectra obtained from the light curve analysis (diagonal panels in Figure 6.5). The Comptonization ($x_n(1)$), disk blackbody ($x_n(2)$), and soft excess ($x_n(3)$) emission has a frequency break at around 0.05 Hz (= 20 s), 0.1 Hz (= 10 s) and 0.05 Hz (= 20 s), respectively. The break frequency in the power spectrum is indicative of the characteristic time scales (Gilfanov, 2010), which are shown in parentheses for each component. The difference in values provides valuable clues to investigate the sizes and radiation positions of each physical component.

We relate these break time scales to the time scales that characterize the accretion disk around the BHBs. One is the Keplerian orbital time, and the other is the viscous time. The Keplerian orbital time is approximated as

$$t_K \approx 0.3 \left(\frac{M}{10M_\odot}\right)\left(\frac{r}{50r_g}\right)^{3/2} \text{ (s)}, \tag{7.1}$$

where M is the blackhole mass in the unit of the solar mass $M_\odot$ and r is the radius in the accretion disk in the unit of the gravitational radius r_g. Meanwhile, the viscous time scale is estimated as

$$\begin{aligned} t_{\rm visc} &\sim \alpha^{-1}\left(\frac{h}{r}\right)^{-2}\Omega_K^{-1} \\ &= \frac{1}{2\pi\alpha}\left(\frac{h}{r}\right)^{-2} t_K \\ &= \frac{0.3}{2\pi\alpha}\left(\frac{h}{r}\right)^{-2}\left(\frac{M}{10M_\odot}\right)\left(\frac{r}{50r_g}\right)^{3/2} \text{ (s)}, \end{aligned} \tag{7.2}$$

Here, α is the viscosity parameter, h is the disk thickness, and $\Omega_k \equiv 2\pi/t_K$. In essence, the Keplarian orbital time is the time scale of the motion in the azimuthal direction around the BHB, while the viscous time scale is that for the motion in the radial direction toward the BHB. For simplicity, we assume $M = 10M_\odot$ in the following discussion.

We first associate the viscous time scale at the innermost radius of the accretion disk with the break frequency of the Comptonized component. This is based on the premise that the Comptonized component is from the accretion flow that extends inward from

the innermost radius of the accretion disk (Gilfanov, 2010). From Equation (7.2), we obtain the innermost radius of $r \sim 100 r_g$, where we assume $\alpha = 0.2$ and $h/r = 0.2$. This size is commonly perceived as typical for the innermost disk radius in the hard state in the truncated disk scenario (Esin et al., 1997).

We next associate the Keplerian orbital time scale at the innermost accretion disk radius with the break frequency of the disk blackbody component, which follows the idea of Gilfanov (2010). Equation (7.1) yields $r \sim 100 r_g$. Interestingly, this value is consistent with the estimate based on the break frequency of the Comptonized component.

The soft excess emission has a time scale similar to the Comptonized emission. We consider that this is because the soft excess emission is made as a response to the Comptonized emission as described in § 7.1.3.

7.1.2 Time Lags between Components

We next examine the time lag among the three spectral components (off-diagonal panels in Figure 6.29). This is again a significant improvement over the power spectra obtained from the light curve analysis (off-diagonal panels in Figure 6.5). The time lag from the disk blackbody emission $x_n(2)$ to the Comptonized emission $x_n(1)$ is evident at ∼0.1 s. This is interpreted as the origin of the hard lag derived from the light curve analysis (§ 4.2.1), as the Comptonized emission dominates more in the harder band, and its delay appears as the hard lag. For the first time, we revealed the hard lag to be between two spectral components in a spectrally decomposed cross-correlation function.

If we associate the delay with a light crossing time, it is of the order of $10^4 r_g$, which is too large for a BHB system. It is usually interpreted as the viscous time scale of the accretion disk or the hot inner accretion flow (Kotov et al., 2001; Arévalo and Uttley, 2006). For the disk, the delay time of 0.1 s equals to the viscous time at a radius of a few r_g. Variations in the disk blackbody emission close to the innermost radius of the accretion disk would serve as seed photons for the Comptonization, which makes the observed delay.

The lag from the Comptonized component $x_n(1)$ to the soft excess component $x_n(3)$ is difficult to interpret in Figure 6.29. It appears to show an anti-correlation peaking at the lag amplitude of 0.8 s. This requires further investigation with data from other epochs. In the light curve analysis, we did observe a ∼0.02 s lag between 0.5–1.0 and 1.0–10 keV light curves (Figure 4.8). This is likely the lag between the Comptonized and the soft excess component. However, this small lag amplitude is beyond the scope of linear Gaussian state-space modeling (§ 6.1.1), as we binned at 0.1 s to secure a Gaussian behavior of the noise.

Still, in Figure 6.29, we observe a time lag between the disk blackbody emission $x_n(2)$ and the soft excess emission $x_n(3)$ of an order of 0.1 s. Therefore, we consider that the order of causality is $x_n(2) \rightarrow x_n(1) \rightarrow x_n(3)$. The variability of the disk blackbody is the origin, which propagates to the Comptonized emission, which propagates to the soft excess emission.

7.1.3 Power Contribution

We further composed the mutual power contributions among the three components (Figure 6.27). Such a plot has not been used in the context of traditional light curve analysis, but is useful for investigating the causality of the time variability as a function of frequency. In the plot, we see that the variations in the Comptonized and disk blackbody emission are largely driven by their own contributions over all frequency ranges. In contrast, the soft-excess component is influenced by the Comptonized component, particularly at a lower frequency than a few Hz. This suggests that the fluctuations in the Comptonization and disk components are independent, while the variation of the soft excess component is largely dependent on the fluctuations of the other components. This is consistent with the causality order discussed from the time lag (§ 7.1.2). The result also suggests that the high-frequency content of the Comptonized emission is not propagated to the soft excess emission, indicating that the soft excess response behaves as a low-pass filter, presumably due to the viscous nature of the accretion disk (Gilfanov, 2010).

7.2 Physical Model Selection

We now revealed the causality order among the three spectral components and associated their variability time scales with the physical scales of the BHB system. Now, we combine these new findings into a picture of the accretion disk geometry of MAXI 1820+070.

The findings can be naturally interpreted in the truncate accretion disk geometry (Figure 7.1). Local fluctuation of the accreting material around the inner radius of the disk causes variability in the X-ray emission. These serve as seed photons for the Comptonization by the hot electron gas in the accretion flow. The time taken from the radiation in the disk to Comptonization is approximately ~ 0.1 s, derived from the lag between the disk blackbody and Comptonized emission components. These components are estimated to arise from a radius of $r \sim 100\ r_g$ based on their characteristic time scales. The Comptonized emission illuminates the accretion disk, which is reprocessed as the soft excess emission. From the lag identified in the differential cross-correlation function, the light crossing time yields a distance of 1000 r_g, which is a typical distance for reprocessing to occur. This is the scenario for presenting the findings in a schematic picture.

The lamppost model envisages the accretion disk extending close to the black hole, where the Keplerian orbital time is ~ 0.01 s (=100 Hz). However, in Figure 5.7, we did not observe substantial power at such high frequencies, but only a nearly monotonic decrease in power beyond $\sim$ 1Hz. Our findings show that the time scale of the disk blackbody emission is much slower of ~ 10 s, which introduces a challenge in interpreting the observed variation of the light curve within the lamppost configuration.

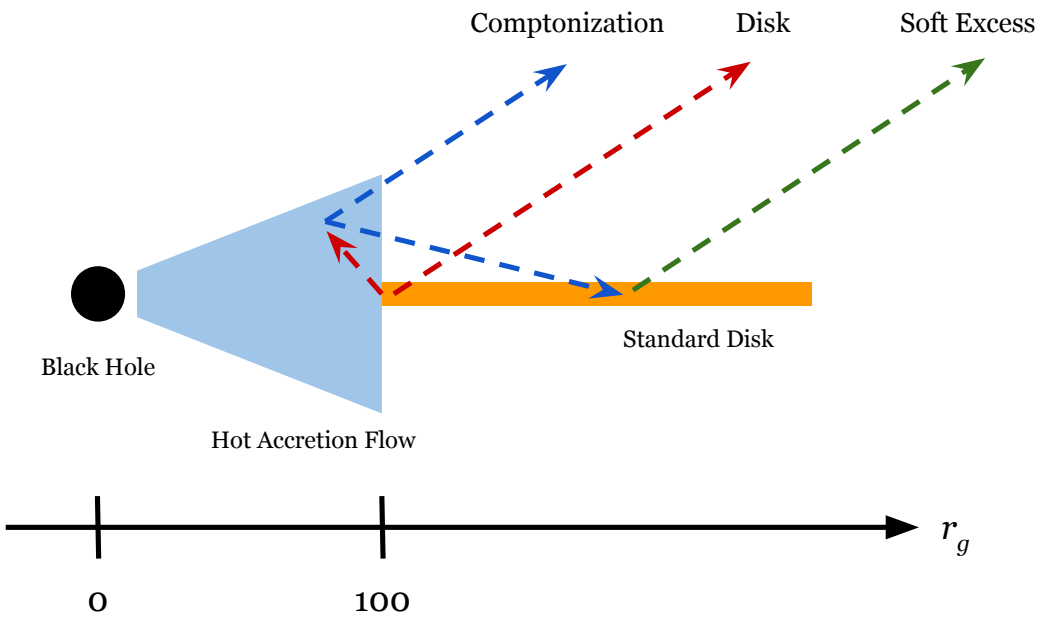


Figure 7.1: Schematics of truncate disk.

Chapter 8

Summary

Rapid X-ray flux variability is one of the most ubiquitous and distinctive characteristics of BHBs in their low hard state. The variability is considered to carry vital information about the geometry around the black hole, in particular, the accretion disk and flow. Numerous studies were made on this topic for its importance in physics. We started this thesis by pointing out the problems in almost all previous studies, in which energy and time series are independently modeled. We clarified the necessity to model both energy and time series jointly. We also stressed the necessity to model latent variables rather than observed variables for the time series modeling to avoid the spectral dilution problem inherent to the traditional analysis.

We demonstrated that such joint modeling is possible using multiband X-ray light curves of the BHB, MAXI J1820+070, observed with NICER. Considering the wide application of our approach in the future far beyond the BHB data analysis, we made a step-by-step progress, starting from some basics of the time series analysis, AR, VAR, and LGSST.

In the classical (AR and VAR) approach, we directly modeled the observed light curves to derive information from the combined contributions of various physical components. Initially, we used AR models that independently regress past values within each band. While we identified some features in the description functions derived from the estimated AR parameters, the results revealed essentially no novel insights compared to traditional analytical methods. Subsequently, we adopted VAR models, an extension of AR models to multivariate cases. The results obtained from the description functions of the VAR model elucidated the features observed in the AR model results, suggesting an enhancement in modeling when considering interactions between different bands.

By modeling the variability of spectral components, it becomes possible to understand various physical characteristics and interactions among them using their variability and propagation. In the LGSSM approach, we attempted to achieve this by modeling the flux of the spectral components as latent variables. They are added linearly to translate to the observed variables, which are the flux in selected energy bands. In this manner, we were successful in modeling the spectral component variability free from the spectral dilution.

Using the description functions of the spectral components, valuable insights were obtained:

(a) The time scales for the Comptonized, disk, and soft-excess components are 0.05 Hz, 0.5 Hz, and 0.05 Hz, respectively.

(b) A time lag of 0.1 s exists from the disk to the Comptonized component, and a time lag of 0.2 s exists from the disk to the soft excess components.

These findings lead to the following conclusions, which are interpreted in the truncated disk scenario.

(1) The time scales of the Comptonized and disk components align with the viscous and Kelparian time scales at a truncation radius of approximately $\sim 100 r_g$ (inferred from (a))

(2) The hard lag is determined as the time lag from the disk component to Comptonization (inferred from (b)).

These pieces of information, previously unattainable by merely examining the characteristics of observed light curves and spectra, provide crucial constraints for the geometry around black holes.

Appendix A

Linear-Gaussian State-Space Modeling Tutorials with TensorFlow Probability

Contents

In the appendix, we provide a tutorial for the linear-Gaussian State-space modeling using TensorFlow Probability. We explain the technical details to the code level of what we did in the main part of this thesis, thereby give a cookbook that can be used to other applications.

A.1 Fundamental Concepts in TFP

TensorFlow Probability (TFP) is a Python library designed for probabilistic inference that leverages TensorFlow (TF) as its numerical computing backbone. It provides implementations of probability distributions and inference tools, allowing users to construct models by combining classes. Users need to understand the essential concepts and implementations for building statistical models with TFP. In this section, we introduce key concepts and provide references. For detailed explanations and tutorials, readers should consult to the resources provided by the TFP development team[1].

Layer In TFP, users can flexibly design probability distributions according to their complexity by segmenting probabilistic inference tools into layers. There are four layers: Layer 0: TensorFlow, Layer 1: statistical building blocks, Layer 2: model construction, and Layer 3: probabilistic inference. For a better understanding of the probabilistic inference aspects defined by each layer, it is best to consult the official page[2].

Distributions `tfp.distributions`[3] is a package that provides various probability distributions, typically imported using `tfd = tfp.distributions`. All classes defined here inherit from the base class `tfd.Distribution`[4], allowing access to characteristics like moments through methods if implemented. Utilizing these classes allows for experimenting with different distributions without major modifications to the model. For instance, assuming counts follow a normal distribution can be described with `tfd.Normal`, and if a Poisson distribution is deemed more appropriate, simply replacing it with `tfd.Poisson` suffices.

Bijectors In statistical modeling, we can limit the range of variable space through transformations of variables and probability distributions. The package for these transformations is `tfp.bijectors`[5], usually imported as `tfb = tfp.bijectors`. The classes in this package inherit from the base class `tfb.Bijector`, enabling bijective transformations through the `forward` or `inverse` methods.

Shape In TFP, there are two important shapes: `event_shape` and `batch_shape`. `event_shape` describes the shape of a single output of a probability distribution, while `batch_shape` represents the shape of outputs of independently distributed objects of a probability distribution. For a deeper understanding, it is recommended to refer to the official website tutorial[6]. In the following, we will demonstrate an example using a multidimensional Gaussian distribution.

[1] https://www.tensorflow.org/probability/overview
[2] https://www.tensorflow.org/probability/overview
[3] https://www.tensorflow.org/probability/api_docs/python/tfp/distributions
[4] https://www.tensorflow.org/probability/api_docs/python/tfp/distributions/Distribution
[5] https://www.tensorflow.org/probability/api_docs/python/tfp/bijectors
[6] https://www.tensorflow.org/probability/examples/Understanding_TensorFlow_Distributions_Shapes

```
import tensorflow as tf
import tensorflow_probability as tfp

tfd = tfp.distributions

# Initialize a single 2-variate Gaussians.
mvn = tfd.MultivariateNormalDiag(
    loc=[1., -1],
    scale_diag=[1, 2.])

print(mvn.event_shape)
# ==> (2,)

print(mvn.batch_shape)
# ==> ()


# Initialize a 2-batch of 3-variate Gaussians.
mvn = tfd.MultivariateNormalDiag(
    loc=[[1., 2, 3],
         [11, 22, 33]],          # shape: [2, 3]
    scale_diag=[[1., 2, 3],
                [0.5, 1, 1.5]])  # shape: [2, 3]


print(mvn.event_shape)
# ==> (3,)

print(mvn.batch_shape)
# ==> (2,)

print(mvn.sample().shape)
# ==> (2, 3)
```

A.2 Time Series Modeling

A.2.1 Univariate Random Walk

We begin by exploring univariate time series modeling to grasp the fundamental aspects of the modeling process. To keep it straightforward, we opt for a model with fewer parameters, starting with the random walk model. In this model, we have only two parameters: the level scale and the observation noise scale.

The state space model for the univariate random walk model is defined as:

$$\begin{aligned} x_n &= x_{n-1} + v_n, \quad v_n \sim \mathcal{N}(0, q) \\ y_n &= x_n + w_n, \quad w_n \sim \mathcal{N}(0, r), \end{aligned} \tag{A.1}$$

Here, q and r represents the level scale and the observation noise scale. This formulation is derived from Equation (3.75) by setting $F = G = H = [1]$.

Generation of Synthetic Data We generate the synthetic data that follows a random walk. We set the level scale and the observation noise scale to 1.0 and 5.0 respectively.

```
import numpy as np

num_timesteps = 100
level_scale = 1.0
observation_noise_scale = 5.0

timestamps = np.arange(num_timesteps)

np.random.seed(123)
latent_noises = np.random.normal(loc=0.0, scale=level_scale,
                                 size=num_timesteps)
latent_values = np.cumsum(latent_noises)
observation_noises = np.random.normal(
    loc=0.0, scale=observation_noise_scale,
    size=num_timesteps)
observation_values = latent_values + observation_noises

# Time series must have the shape of
# (batch_shape, num_timesteps, variable_size).
latent_values = latent_values[:, np.newaxis]
observation_values = observation_values[:, np.newaxis]
```

It is noted that in the time series modeling by TFP, the canonical shape is `(batch_shape, num_timesteps, variable_size)`. Since we treat a single univariate time series as an observation variable, the shape would be `[batch_shape, num_timesteps, 1]`. To achive this requirement, we add a new axis to the `latent_values` and `observation_values`.

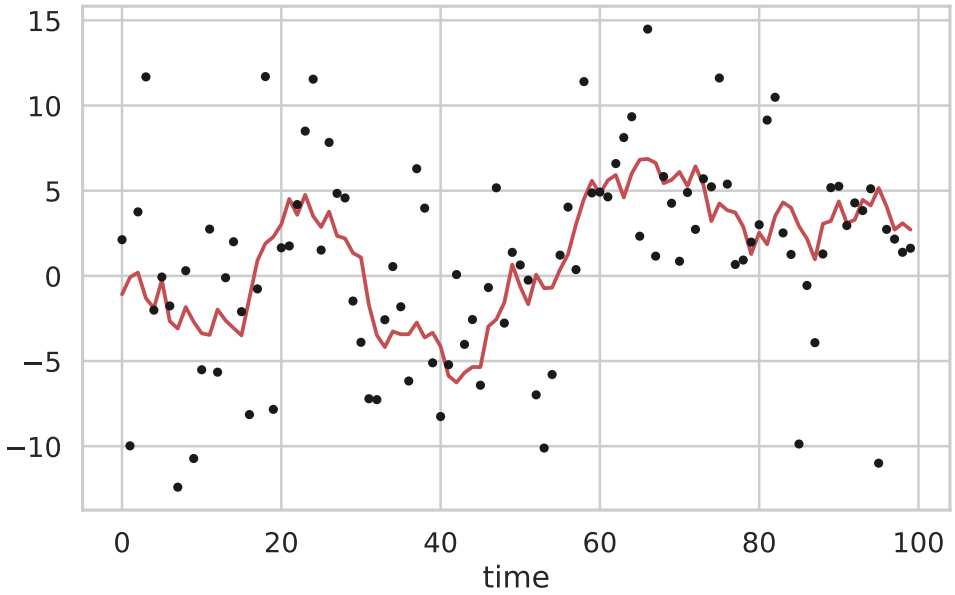


Figure A.1: Samples from the univariate random walk. y_n and x_n in Equation (A.1) are represented by the black points and the red curve.

State Estimation Let us estimate unobserved state variables through observed values. Although the true parameter values are unknown in most cases, we consider the situation where we already have in order to focus on the state estimation process here. The state estimation code is written as follows.

```
import tensorflow as tf
import tensorflow_probability as tfp
from tensorflow_probability import sts

tfd = tfp.distributions
sts = tfp.sts

# Define data type.
tf_dtype = tf.float32

# Data type of numpy.array is usually double (`np.float64`).
# The model inference needs no precise calculation here.
observation_values = tf.convert_to_tensor(observation_values, dtype=tf_dtype)

# The model is defined using a \texttt{tfp.sts.*StateSpaceModel}.
# The `tfp.sts` library is only support the univariate structural time series
# (STS) model.
rw_ssm = sts.LocalLevelStateSpaceModel(
    num_timesteps,
    level_scale=1.0,
    observation_noise_scale=5.0,
    # The batch shape must match the state size of 1, in this case.
    initial_state_prior=tfd.MultivariateNormalDiag(
        loc=[0.],
```

```
        scale_diag=[1.]
    )
)

(
  log_likelihoods,
  filtered_means,
  filtered_covs,
  predicted_means,
  predicted_covs,
  observation_means,
  observation_covs,
) =  rw_ssm.forward_filter(observation_values)

(
  posterior_means,
  posterior_covs
) = rw_ssm.backward_smoothing_pass(
  filtered_means,
  filtered_covs,
  predicted_means,
  predicted_covs
)
```

We present the filtered and smoothed states alongside the true states in Figure A.2. The filtering state demonstrates that the smoothed state lies within less than two standard deviations of the normal distribution. The smoothed state approximates the true state more closely than the filtered state. This discrepancy originates from the number of observations considered under the respective probability conditions: $p(x_n|y_n)$ for filtering and $p(x_n|y_N)$ for smoothing.

A.2.2 Multivariate Dynamic Factor Model

Generation of the Synthetic Data Synthetic data is created in two steps to align with the system model and the observation model in the state-space model. First, state variables are generated following a vector autoregression model (VAR), and then observation variables are derived from these state variables based on the observation model. Sample generation for state variables employs `statsmodels.tsa.vector_ar.var_model.VARProcess`[7], while observation variables are generated using linear calculations.

```
import numpy as np
from statsmodels.tsa.vector_ar.var_model import VARProcess


```

[7] https://www.statsmodels.org/stable/generated/statsmodels.tsa.vector_ar.var_model.VARProcess.html

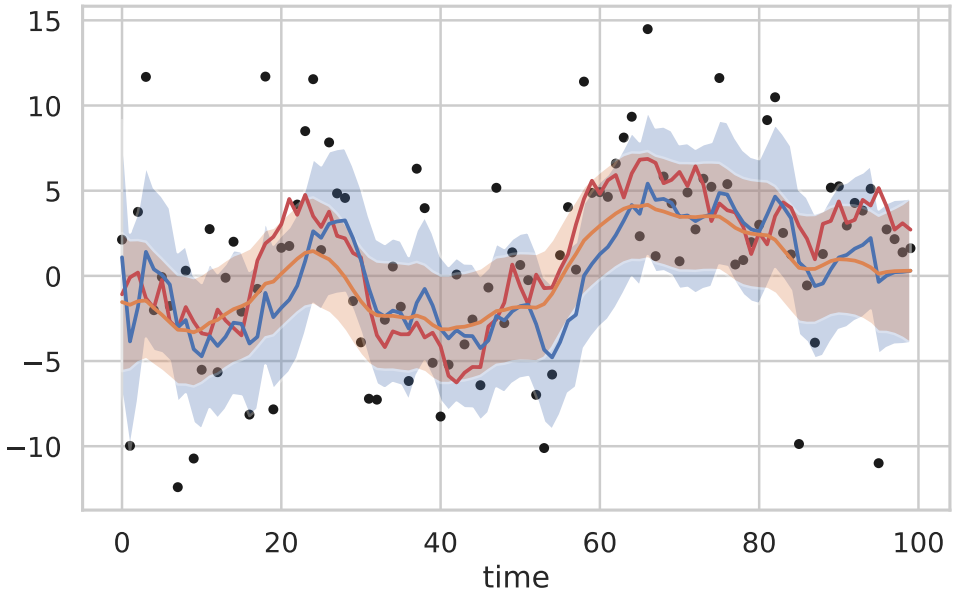


Figure A.2: Filetered (blue) and smoothed (orange) states with observations (black circle). The true states are illustraited by the red curve. The shadded areas represent the 95 % confidence intervals for filtered and smoothed states.

```
batch_shape = ()
latent_size = 2
observation_size = 3
order = 2

num_timesteps = 100

var_coefficients = np.array([
  [[ 0.60, -0.01],
   [ 0.04,  0.50]],
  [[ 0.10,  0.02],
   [ 0.01, -0.09]]
])
transition_noise_cov = np.array(
  [[1.0, 0.0],
   [0.0, 1.0]])
observation_matrix = np.array([
  [0.9, 0.0],
  [0.5, 0.4],
  [0.3, 0.2]
])
observation_noise_cov = np.array([
  [0.1, 0.0, 0.0],
  [0.0, 0.1, 0.0],
  [0.0, 0.0, 0.1]
])
```

```
31
32 # Generate state variables.
33 var_process = VARProcess(coefs=var_coefficients,
34                          coefs_exog=None,
35                          sigma_u=transition_noise_cov,)
36
37 timestamps = np.arange(num_timesteps)
38 latent_samples = var_process.simulate_var(
39     num_timesteps, seed=123)
40
41 # Compute Observation variables.
42 observation_values = np.matmul(
43     latent_samples,
44     observation_matrix.T,
45 )
46 observation_noises = np.random.multivariate_normal(
47     mean=np.zeros(observation_size),
48     cov=observation_noise_cov,
49     size=num_timesteps
50 )
51
52 observation_samples = observation_values + observation_noises
```

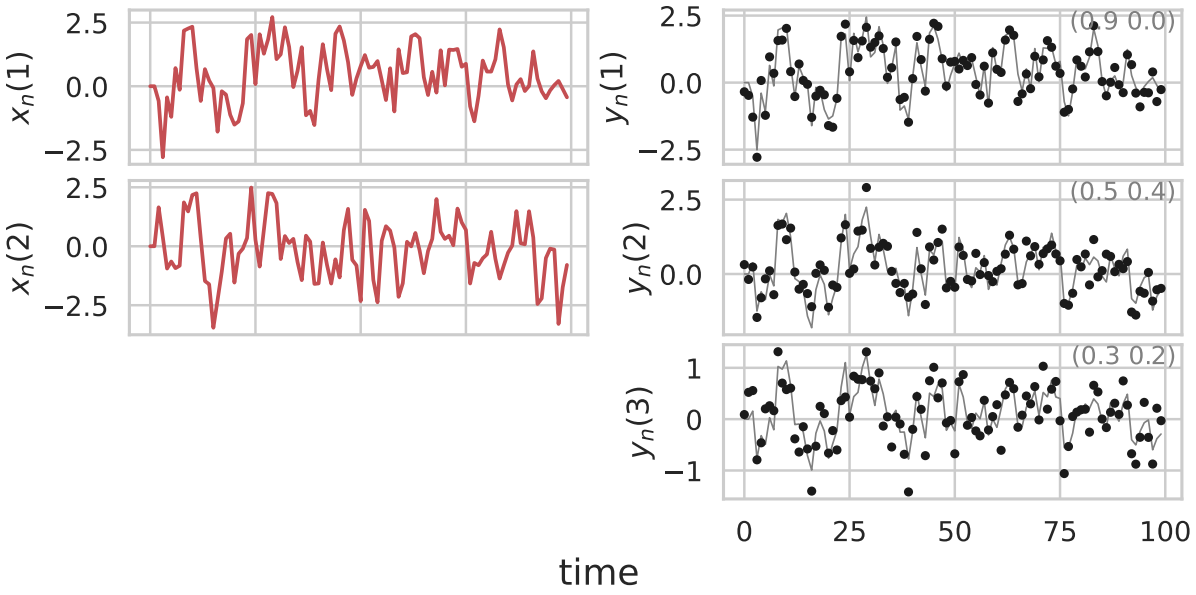


Figure A.3: Latent variables (left) and observation variables (right). The observation variables are the mixture of $x = [x_n(1)\ x_n(2)]^T$ and coefficients, shown in the upper right of each panel. The gray curves in the right panels indicate values where noise terms have been removed.

Definition of Dynamic Factor Model with VAR model We start with importing the libraries.

```
import tensorflow as tf
import tensorflow_probability as tfp

tfb = tfp.bijectors
tfd = tfp.distributions
```

As a reference, in this coding, we write down the state-space representation of the dynamic factor model, where the system model follows the VAR model. The system model is expressed as

$$\begin{bmatrix} x_n \\ \vdots \\ x_{n-m-1} \\ x_{n-m} \end{bmatrix} = \begin{bmatrix} A_1 & \cdots & A_{m-1} & A_m \\ I & \cdots & O & O \\ \vdots & \ddots & \ddots & \vdots \\ O & \cdots & I & O \end{bmatrix} \begin{bmatrix} x_{n-1} \\ \vdots \\ x_{n-m-2} \\ x_{n-m-1} \end{bmatrix} + \begin{bmatrix} v_n \\ O \\ \vdots \\ O \end{bmatrix}, \quad v_n \sim \mathcal{N}(0, Q), \tag{A.2}$$

where

$$x_n = \begin{bmatrix} x_n(1) \\ \vdots \\ x_n(k) \end{bmatrix}, \quad v_n = \begin{bmatrix} v_n(1) \\ \vdots \\ v_n(k) \end{bmatrix}, \quad A_i = \begin{bmatrix} a_{i,11} & \cdots & a_{i,1\ell} \\ \vdots & \ddots & \vdots \\ a_{i,\ell 1} & \cdots & a_{i,\ell\ell} \end{bmatrix}. \tag{A.3}$$

The observation model is

$$y_n = \begin{bmatrix} H_1 & O & \cdots & O \end{bmatrix} \begin{bmatrix} x_n \\ \vdots \\ x_{n-m-1} \\ x_{n-m} \end{bmatrix} + w_n, \quad w_n \sim \mathcal{N}(0, R), \tag{A.4}$$

where

$$y_n = \begin{bmatrix} y_n(1) \\ \vdots \\ y_n(\ell) \end{bmatrix}, \quad H_1 = \begin{bmatrix} h_{11} & \cdots & h_{1k} \\ \vdots & \ddots & \vdots \\ h_{\ell 1} & \cdots & h_{\ell k} \end{bmatrix}, \tag{A.5}$$

We explicitly differentiate between the terms "observation matrix" and "loading matrix". The observation matrix refers to the matrix that is multiplied by the state variable, represented as $[H_1 \ O \ \cdots \ O]$. On the other hand, the loading matrix, denoted by H_1, specifically contributes to the variable x_n without involving additional zero matrices.

In the system model expressed by Equation (A.2), the transition matrix has the $A_1, \cdots, A_m$ at the top row, which is the state-space form of the VAR model. It is convenient to define the function that converts the VAR coefficients of the shape $(m \times k \times k)$ into the transition matrix of the VAR model with the shape $(mk \times mk)$.

```
def make_var_transition_matrix(coefficients):
  """Build transition matrix for an vactor autoregressive StateSpaceModel.

    When applied to a vector of previous values, this matrix computes
    the expected new value (summing the previous states according to the
    autoregressive coefficients) in the top dimension of the state space,
    and moves all previous values down by one dimension, 'forgetting' the
```

```
    final (least recent) value. That is, it looks like this:
    ```
    var_matrix =
        [c[0, 0, 0], ..., c[0, 0, M], c[1, 0, 0], ..., c[order, 0, M]
          ...,
          c[0, M, 0], ..., c[0, M, M], c[1, M, 0], ..., c[order, M, M]
          1.,   ..., 0 ,   0.,   ..., 0.
          ...
          0.,   ..., 1.,   0.,   ..., 0.
          ...
          0.,   ..., 0.,   1.,   ..., 0.   ]
    ```
    Args:
        coefficients: float `Tensor` of shape `concat([batch_shape, [order]])`.
    Returns:
        ar_matrix: float `Tensor` with shape `concat([batch_shape,
        [order, order]])`.

    ### Example
        ```python
        make_var_transition_matrix(
            [[[1.0, 0.0],
              [0.5, 0.5]],
             [[0.3, 0.7],
              [0.1, 0.9]]]
        )
        )
        # ===> [[1.0, 0.0, 0.3, 0.7],
        #       [0.5, 0.5, 0.1, 0.9],
        #       [1.0, 0.0, 0.0, 0.0],
        #       [0.0, 1.0, 0.0, 0.0]]
        ```
    """
    dtype = coefficients.dtype

    coef_shape = coefficients.shape
    batch_shape = coef_shape[:-3]
    order, latent_size = coef_shape[-3], coef_shape[-2]

    top_row = tf.concat(tf.unstack(coefficients, axis=-3), axis=-1)
    remaining_rows = tf.concat([
        tf.eye(latent_size * (order - 1), dtype=dtype,
               batch_shape=batch_shape),
        tf.zeros(tf.concat(
                    [batch_shape, (latent_size * (order - 1), latent_size)],
                    axis=0
                ),
                dtype=dtype)
```

```
    ], axis=-1)
    var_matrix = tf.concat([top_row, remaining_rows], axis=-2)
    return var_matrix
```

The observation matrix has a unique shape: the loading matrix is located on the leftmost, and the remaining are zero matrices. The function used to compute the dynamic factor model with VAR.

```
def make_dfmvar_observation_matrix(matrix, var_order):
  """Builds observation matrix for a Dynamic Factor Model with VAR model.

    Appends zeros to the right of the given matrix to form the observation
    matrix.

    Args:
        matrix: float or int `Tensor` of shape
            `concat([batch_shape, [observation_size, latent_size]])`.
            representing the matrix to be extended.
        var_order: int. The order of the VAR model.

    Returns:
        observation_matrix: float `Tensor` with shape
        `concat([batch_shape, [observation_size, latent_size * order]])`
            representing the observation matrix.

    Example:
        ```python
        make_dfmvar_observation_matrix(
                [[1, 2],
                 [3, 4],
                 [5, 6]], var_order=3
        )
        # ===> [[[1, 2, 0, 0, 0, 0],
        #        [3, 4, 0, 0, 0, 0],
        #        [5, 6, 0, 0, 0, 0]]]
        ```
    """
    batch_shape = matrix.shape[:-2]
    observation_size, latent_size = matrix.shape[-2:]
    remaining_rows = tf.zeros(
        tf.concat([batch_shape,
                   [observation_size],
                   [(var_order-1) * latent_size]],
                  axis=-1),
        dtype=dtype)
    observation_matrix = tf.concat([matrix, remaining_rows], axis=-1)
    return observation_matrix
```

We define the class for the dynamic factor model incorporating the VAR approach utilizing these functions. Our coding style aligns with the conventions in the state space modeling classes, such as `tfp.sts.AutoregressiveStateSpaceModel`[8].

```
# The class inherits from `tfp.distributions.LinearGaussianStateSpaceModel`.
class DynamicFactorVectorAutoregressiveStateSpaceModel(
        tfd.LinearGaussianStateSpaceModel):
    """State space model for dynamic factor models.

    A state space model (SSM) posits a set of latent (unobserved) variables that
    evolve over time with dynamics specified by a probabilistic transition model
    `p(z[t+1] | z[t])`. At each timestep, we observe a value sampled from an
    observation model conditioned on the current state, `p(x[t] | z[t])`. The
    special case where both the transition and observation models are Gaussians
    with mean specified as a linear function of the inputs, is known as a linear
    Gaussian state space model and supports tractable exact probabilistic
    calculations; see `tfp.distributions.LinearGaussianStateSpaceModel` for
    details.

    In a dynamic factor model, the number of the latent size is smaller than
    the observation size.

    ```python
    x[t] = coefficient x[t-1] + MultivariateNormal(0, state_noise_cov)
    y[t] = loading_matrix x[t] + MultivariateNormal(0, observation_noise_cov)
    ```

    The system process is characterized by a matrix `coefficients` whose size
    of the first dimension determines the order of the process (how many
    previous values it looks at), by `state_noise_cov`, the covariance matrix of
    the state noise added at each step. The observation process is characterized
    by a matrix `loading_matrix` which determines the contribution of a state
    variable to an observation variable, and by `observation_noise_cov`, the
    covariance matrix of the noise for the observation process.

    This is formulated as a state space model by letting the latent state encode
    the most recent values; see 'Mathematical Details' below.

    The parameters `coefficients` is a (batch) tensor of shape `[order,
    latent_size, latent_size]`, and `loading_matrix`, `state_noise_cov` and
    `observation_noise_cov` are each (a batch of) matrix. The batch shape of
    this `Distribution` is the broadcast batch shape of these parameters and
    of the `initial_state_prior`.

    #### Mathematical Details

```

[8] https://www.tensorflow.org/probability/api_docs/python/tfp/sts/AutoregressiveStateSpaceModel

```
    The dynamic factor model implements a
    `tfp.distributions.LinearGaussianStateSpaceModel` with `latent_size = order*
    ndims_latent` and `observation_size = ndims_observation`. The latent state
    vector encodes the recent history of the process, with the current value in the
     topmost dimensions. At each timestep, the transition sums the previous values
    to produce the new expected value, shifts all other values down by a dimension,
     and adds noise to the current value. This is formally encoded by the
    transition model:

    ```
    transition_matrix = [ coefs[0], coefs[1], ..., coefs[order]
                          I,        0,        ..., 0
                          0,        I,        ..., 0
                          ...
                          0,        0,   ...,   I, 0           ]
    transition_noise ~ MN(loc=0., cov=state_noise_cov)
    ```

    The observation model simply extracts the current (topmost) value, and
    optionally adds independent noise at each step:

    ```
    observation_matrix = [[loading_matrix, 0, ..., 0]]
    observation_noise ~ MN(loc=0, cov=observation_noise_cov)
    ```

    #### Examples

    A simple model definition:

    ```python
    dfm_model = DynamicFactorModelStateSpaceModel(
        num_timesteps=50,
        coefficients=[
            [[ 0.8, -0.1],
             [ 0.5, -0.2]],
            [[ 0.3,  0.1],
             [-0.2,  0.0]]
        ],
        loading_matrix=[
            [1.0, 0.5],
            [0.3, 0.2],
            [0.1, 0.0]
        ],
        state_noise_cov=[
            [1.0, 0.0],
            [0.0, 1.0]
        ],
```
```

```
        observation_noise_cov=[
            [ 1.0, 0.3, -0.1],
            [ 0.3, 0.4,  0.0],
            [-0.1, 0.0,  0.1]
        ],
        initial_state_prior=tfd.MultivariateNormalDiag(
          scale_diag=[1., 1., 1., 1.]))

    y = dfm_model.sample() # y has shape [50, 3]
    lp = dfm_model.log_prob(y) # log_prob is scalar
    ```

    """

    def __init__(self,
                 num_timesteps,
                 coefficients,
                 loading_matrix,
                 state_noise_cov,
                 observation_noise_cov,
                 initial_state_prior,
                 name=None,
                 **linear_gaussian_ssm_kwargs):
        """Build a state space model implementing a dynamic factor model.
        Args:
            num_timesteps: int. Number of time steps in the model.
            coefficients: Tensor. Autoregressive coefficients.
            loading_matrix: Tensor. Loading matrix.
            state_noise_cov: Tensor. Covariance matrix for state noise.
            observation_noise_cov: Tensor. Covariance matrix for observation
                noise.
            initial_state_prior: tfd.MultivariateNormal. Prior distribution for
                initial state.
            name: str. Optional name for the model.
            **linear_gaussian_ssm_kwargs: Additional keyword arguments passed
                to `tfd.LinearGaussianStateSpaceModel`.
        """

        parameters = dict(locals())
        parameters.update(**linear_gaussian_ssm_kwargs)
        del parameters["linear_gaussian_ssm_kwargs"]

        with tf.name_scope("DynamicFactor") as name:

            dtype = initial_state_prior.dtype

            coefficients = tf.convert_to_tensor(
                value=coefficients,
```
```

```
133                 name="coefficients", dtype=dtype)
134             loading_matrix = tf.convert_to_tensor(
135                 value=loading_matrix,
136                 name="loading_matrix", dtype=dtype)
137             state_noise_cov = tf.convert_to_tensor(
138                 value=state_noise_cov,
139                 name="state_noise_cov", dtype=dtype)
140             observation_noise_cov = tf.convert_to_tensor(
141                 value=observation_noise_cov,
142                 name="observation_noise_cov", dtype=dtype)
143
144             constant_offset = tf.convert_to_tensor(
145                 value=constant_offset,
146                 name="constant_offset", dtype=dtype)
147
148             batch_shape = coefficients.shape[:-3]
149             order = coefficients.shape[-3]
150             latent_size = loading_matrix.shape[-1]
151
152             super(DynamicFactorVectorAutoregressiveStateSpaceModel,
153                   self).__init__(
154                 num_timesteps=num_timesteps,
155                 transition_matrix=make_var_transition_matrix(coefficients),
156                 # The transition noise must be an instance of
157                 # `tfd.MultivariateNormalLinearOperator`.
158                 transition_noise=tfd.MultivariateNormalTriL(
159                     scale_tril=tf.linalg.LinearOperatorBlockDiag(
160                         [tf.linalg.LinearOperatorFullMatrix(
161                             tf.linalg.cholesky(state_noise_cov)),
162                          tf.linalg.LinearOperatorZeros(
163                             num_rows=(order-1)*latent_size,
164                             batch_shape=batch_shape)]).to_dense()
165                 ),
166                 observation_matrix=make_dfmvar_observation_matrix(
167                     loading_matrix, order),
168                 # The observation noise must be an instance of
169                 # `tfd.MultivariateNormalLinearOperator`.
170                 observation_noise=tfd.MultivariateNormalTriL(
171                     scale_tril=tf.linalg.cholesky(observation_noise_cov)),
172                 initial_state_prior=initial_state_prior,
173                 name=name)
174
175             self._parameters = parameters
```

Model Inference with MCMC As discussed in Section 6.3, it is necessary to constrain the parameter shapes of the dynamic factor model to ensure identifiability.

For the loading matrix, we impose a condition requiring that the loading matrix is composed of a lower-triangular matrix combined with a rectangular matrix. Consequently, it is practical to define a function that constructs this matrix from a one-dimensional array.

```
def fill_trapezoidal(x, ncols):
    """Creates a (batch of) trapezoidal matrix from a vector of inputs.

    Fills a tensor with trapezoidal, that is an upper triangular part and
    a rectangular part.

    Args:
        x (tf.Tensor): The input tensor.
        ncols (int): The number of columns for the rectangular part.

    Returns:
        tf.Tensor: A tensor with the upper triangular part followed by the
    rectangular part.

    Raises:
        ValueError: If the input tensor's shape is not compatible with the
    operation.

    ### Example:
        ```python
        fill_tripezoidal(
            x=[1, 2, 3, 4, 5, 6, 7, 8, 9],
            ncols=3)
        # ===> [[4, 0, 0],
        #       [6, 5, 0],
        #       [3, 2, 1],
        #       [7, 8, 9]]
        ```
    """
    x = tf.convert_to_tensor(x, dtype=dtype)
    batch_shape = x.shape[:-1]

    ntri = ncols*(ncols+1) // 2

    x_tril = x[..., :ntri]
    tril = tfp.math.fill_triangular(x_tril)

    x_rect = x[..., ntri:]
    nrows_rect = x_rect.shape[-1] // ncols
    rect = tf.reshape(x_rect, (*batch_shape, nrows_rect, ncols))

    trapezoidal = tf.concat([tril, rect], axis=-2)

```

```
    return trapezoidal

def fill_trapezoidal_inverse(x):
    """Creates a vector from a (batch of) trapezoidal matrix.

    Reconstructs the original vector from a (batch of) trapezoidal matrix.

    Args:
        x: Tensor representing trapezoidal elements.

    Returns:
        x_flat: (Batch of) vector-shaped Tensor representing
        vectorized trapezoidal elements from x.

    Example:
        ```python
        fill_trapezoidal_inverse(
            x=[[4, 0, 0],
               [6, 5, 0],
               [3, 2, 1],
               [7, 8, 9]])
        # ===> [1, 2, 3, 4, 5, 6, 7, 8, 9]
        ```
    """
    x = tf.convert_to_tensor(x)

    batch_shape = x.shape[:-2]
    ncols = x.shape[-1]

    x1 = tfp.math.fill_triangular_inverse(x[..., :ncols, :])
    x2 = tf.reshape(x[..., ncols:, :],
                    tf.concat([batch_shape, [-1]], axis=0))
    x_flat = tf.concat([x1, x2], axis=-1)

    return x_flat
```

Given that the synthetic data consists of three observed variables and one of the latent variables is set to two, the parameters in Equations (A.2), (A.3), and (A.5) are specified as follows:

$$A_1 = \begin{bmatrix} a_{1,11} & a_{1,12} \\ a_{1,21} & a_{1,22} \end{bmatrix}, \; A_2 = \begin{bmatrix} a_{2,11} & a_{2,12} \\ a_{2,21} & a_{2,22} \end{bmatrix}, \quad H_1 = \begin{bmatrix} h_3 & 0 \\ h_2 & h_1 \\ h_4 & h_5 \end{bmatrix}, \quad L_R = \begin{bmatrix} r_4 & 0 & 0 \\ r_6 & r_5 & 0 \\ r_3 & r_2 & r_1 \end{bmatrix}. \tag{A.6}$$

The code to infer these parameters using class and functions defined above is given below. The HMC inferece process written here refers to the tutorial given by the TFP

developing team[9], which should be read for further details.

```
num_timesteps = 100
order = 2
latent_size = 2
observation_size = 3

observed_values = tf.covert_to_tensor(observed_values, dtype=tf.float32)


def ssm_dfm():

    coefficients_flat = yield tfd.MultivariateNormalDiag(
            scale_diag=order*latent_size*latent_size*[1.])
    coefficients = tf.reshape(coefficients_flat,
                              (-1, order, latent_size, latent_size))

    num_loadings = int(
        latent_size * (latent_size + 1) / 2
        + (observation_size - latent_size) * latent_size
    )
    factor_matrix_flat = yield tfd.HalfCauchy(
        loc=num_loadings*[0.],
        scale=num_loadings*[5.]
    )
    factor_matrix = fill_tripezoidal(
        factor_matrix_flat, latent_size)

    observation_noise_nondiag = yield tfd.CholeskyLKJ(
        dimension=observation_size,
        concentration=3)
    observation_noise_diag = yield tfd.HalfCauchy(
        loc=observation_size*[0.],
        scale=observation_size*[5.]
    )
    observation_noise_cov_chol = tf.linalg.diag(
        observation_noise_diag) @ observation_noise_nondiag
    observation_noise_cov = tf.matmul(
        observation_noise_cov_chol,
        observation_noise_cov_chol,
        transpose_b=True)

    batch_shape = coefficients_flat.shape[:-1]
    state_noise_cov = tf.eye(latent_size,
                             batch_shape=batch_shape)

```

[9] https://www.tensorflow.org/probability/examples/STS_approximate_inference_for_models_with_non_Gaussian_observations

```
    initial_state_prior = tfd.MultivariateNormalDiag(
            loc=0.0*tf.ones(order*latent_size),
            scale_diag=1.0*tf.ones(order*latent_size))

    yield DynamicFactorVectorAutoregressiveStateSpaceModel(
        num_timesteps,
        coefficients,
        factor_matrix,
        state_noise_cov,
        observation_noise_cov,
        initial_state_prior,
        name="observed_values")

joint_model = tfd.JointDistributionCoroutineAutoBatched(ssm_dfm)

pinned_model = joint_model.experimental_pin(observed_values=observed_values)
constraining_bijector = (
  pinned_model.experimental_default_event_space_bijector()
)

# Allow external control of sampling to reduce test runtimes.
num_results = 500
num_results = int(num_results)

num_burnin_steps = 1000
num_burnin_steps = int(num_burnin_steps)

sampler = tfp.mcmc.TransformedTransitionKernel(
    tfp.mcmc.NoUTurnSampler(
        target_log_prob_fn=pinned_model.unnormalized_log_prob,
        step_size=0.1),
    bijector=constraining_bijector)

adaptive_sampler = tfp.mcmc.DualAveragingStepSizeAdaptation(
    inner_kernel=sampler,
    num_adaptation_steps=int(0.8 * num_burnin_steps),
    target_accept_prob=0.75)

initial_state = constraining_bijector.forward(
    type(pinned_model.event_shape)(
        *(tf.random.normal(part_shape)
          for part_shape in constraining_bijector.inverse_event_shape(
              pinned_model.event_shape))))


# Speed up sampling by tracing with `tf.function`.
@tf.function(autograph=False, jit_compile=True)
def do_sampling():
```

```
    return tfp.mcmc.sample_chain(
        kernel=adaptive_sampler,
        current_state=initial_state,
        num_results=num_results,
        num_burnin_steps=num_burnin_steps,
        trace_fn=None,
        seed=123)
mcmc_samples_list = do_sampling()
```

Sanity Check for MCMC Sampling To verify the results of the inference, we examine the parameter traces. The MCMC samples for the VAR coefficients, elements of the loading matrix, and elements of the observation noise covariance matrix are illustrated in Figures A.4, A.5, and A.6, respectively. The distribution modes for all parameters closely approximate their true values, with the exceptions of h_1 and h_5. These parameters correspond to the weights from $x_n(2)$ to $y_n(2)$ and from $x_n(2)$ to $y_n(3)$, respectively. Given their lower assigned values compared to those of $x_n(1)$, accurately estimating their posterior distributions may pose a challenge. However, the fact that the true values likely fall within the main body of the distributions suggests that the estimations are, on the whole, reasonably accurate.

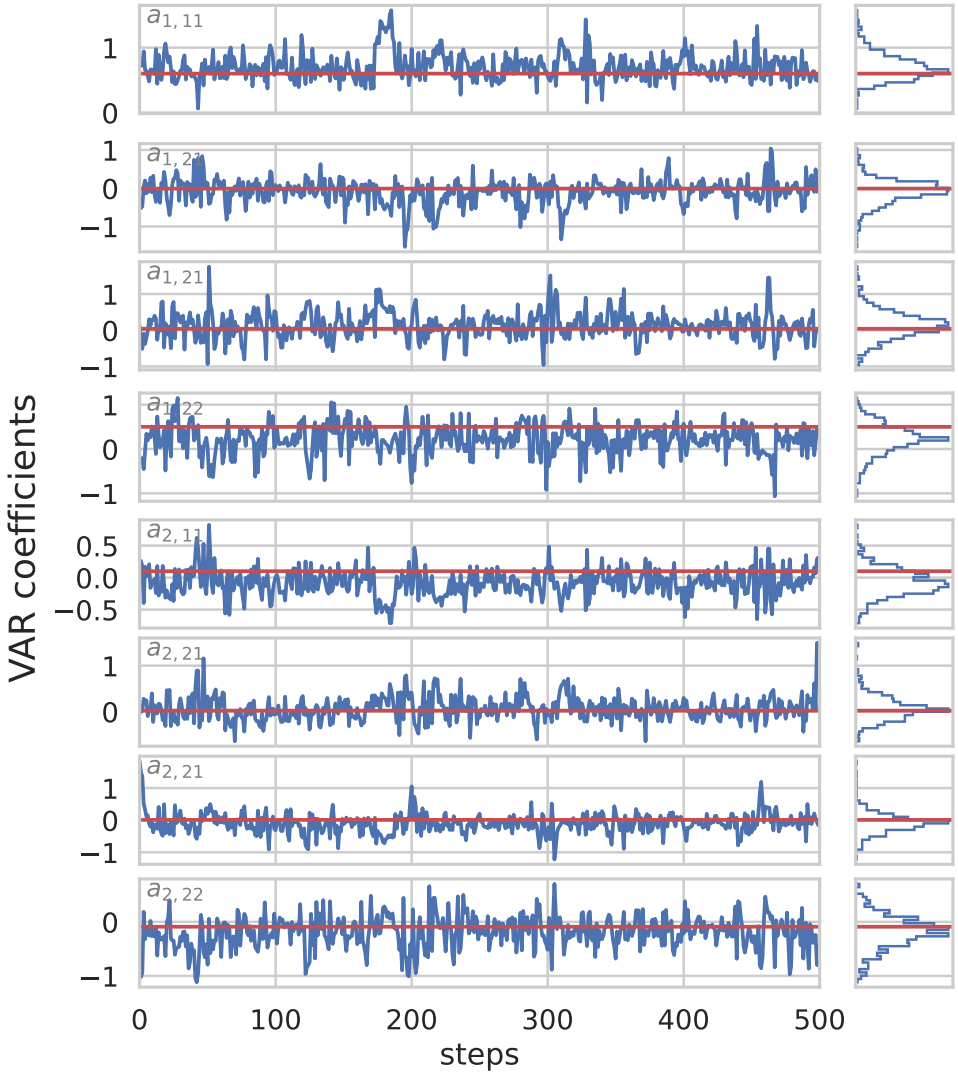


Figure A.4: The trace plots (left) and histograms (right) for the MCMC samples of the VAR coefficients. The red line represents the true values. The parameter names, shown in Equation (A.6), are indicated at the upper left. The burnin steps are not illustraited.

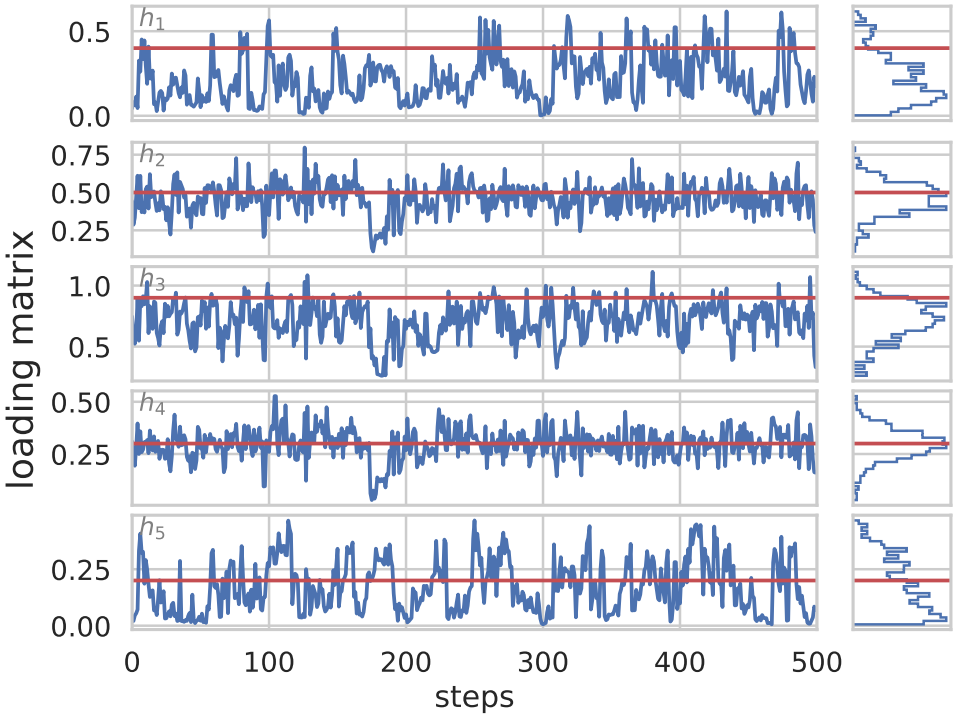


Figure A.5: The trace plots (left) and histograms (right) for the MCMC samples of the loading matrix elements. The red line represents the true values. The parameter names, shown in Equation (A.6), are indicated at the upper left.

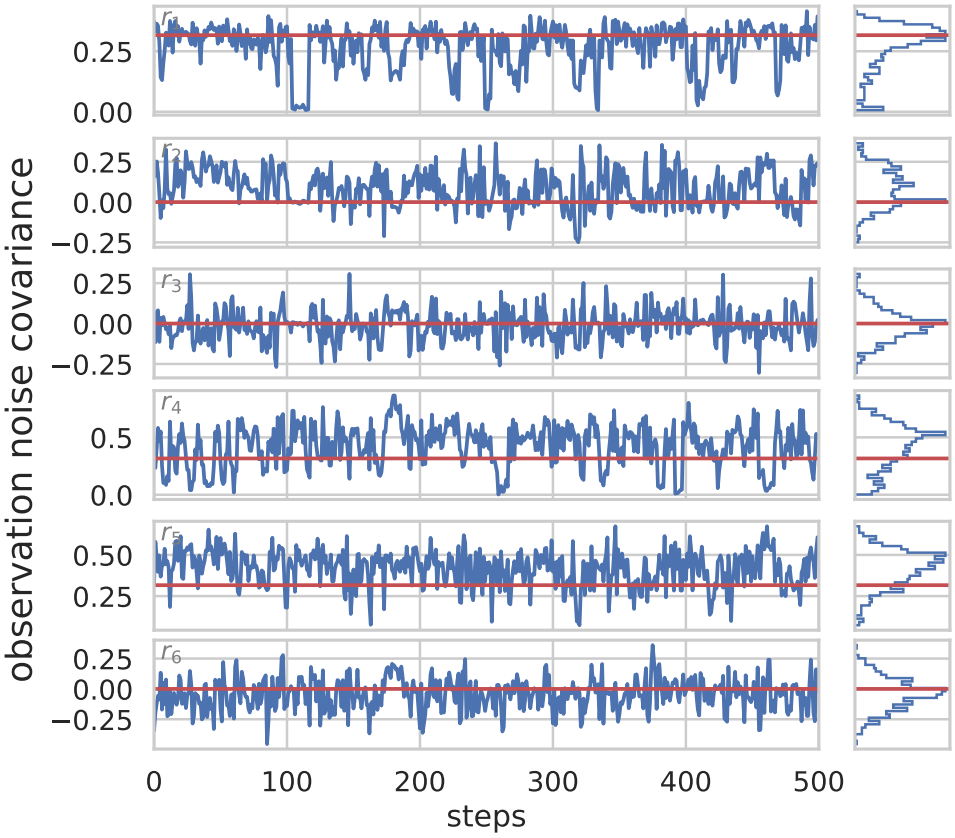


Figure A.6: The trace plots (left) and histograms (right) for the MCMC samples of the covariance matrix. The red line represents the true values. The parameter names, shown in Equation (A.6), are indicated at the upper left.

```
var_coefficients_mcmc = tf.reshape(
  mcmc_samples_list_new[0],
  (-1, order, latent_size, latent_size)
)
obs_matrix_mcmc =  fill_tripezoidal(
  mcmc_samples_list_new[1], latent_size)

obs_noise_cov_chol_mcmc = tfp.math.fill_triangular(mcmc_samples_list_new[2])
obs_noise_cov_mcmc = tf.matmul(
    obs_noise_cov_chol_mcmc,
    obs_noise_cov_chol_mcmc,
    transpose_b=True
)
state_noise_cov = tf.eye(latent_size, dtype=tf.float32)

initial_state_prior = tfd.MultivariateNormalDiag(
        loc=0.0*tf.ones(order*latent_size),
        scale_diag=1.0*tf.ones(order*latent_size))

```

```
ssm_dfm_mcmc = DynamicFactorVectorAutoregressiveStateSpaceModel(
  num_timesteps=num_timesteps,
  coefficients=var_coefficients_mcmc,
  loading_matrix=obs_matrix_mcmc,
  state_noise_cov=state_noise_cov,
  observation_noise_cov=obs_noise_cov_mcmc,
  initial_state_prior=initial_state_prior
)

# Make filtered, predicted, and smoothed latent/obervation curves.
observed_values = tf.convert_to_tensor(observed_values, dtype=tf.float32)
(log_likelihoods_mcmc,
  filtered_means_mcmc, filtered_covs_mcmc,
  predicted_means_mcmc, predicted_covs_mcmc,
  observation_means_mcmc, observation_covs_mcmc) = ssm_dfm_mcmc.forward_filter(
    observed_values)
smoothed_means_mcmc, smoothed_covs_mcmc = ssm_dfm_mcmc.backward_smoothing_pass(
    filtered_means_mcmc, filtered_covs_mcmc,
    predicted_means_mcmc, predicted_covs_mcmc)
```

Figure A.7 shows the state variables estimated from the MCMC samples. The estimated values for both $x_n(1)$ and $x_n(2)$ roughly match the true values, but the fluctuation in $x_n(2)$ is larger than that in $x_n(1)$. This is because $x_n(1)$ serves as the regression curve for $y_n(1)$, making error evaluation straightforward. In contrast, estimating $x_n(2)$ is more challenging as it requires evaluating regression coefficients after removing its influence.

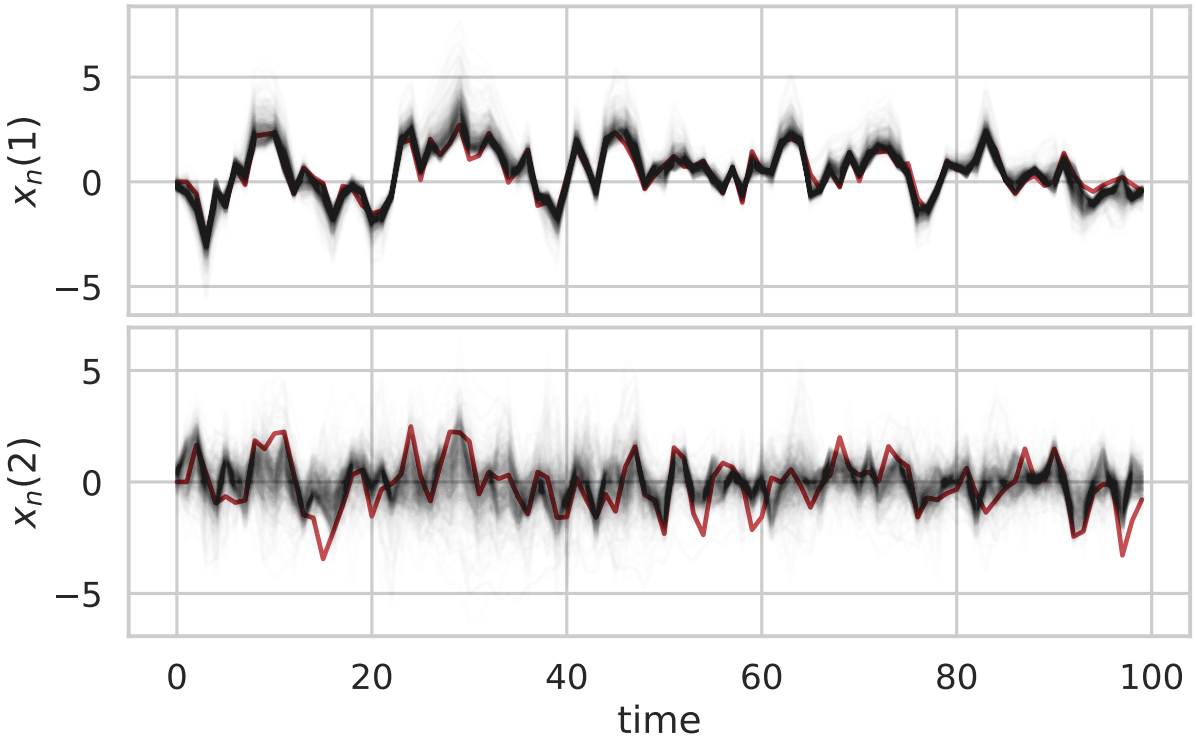


Figure A.7: Estimated states and true states. The estimated states are computed from the MCMC samples. The true states are represented by the red curves.

Bibliography

NICER Mission Guide (2023). https://heasarc.gsfc.nasa.gov/docs/nicer/mission_guide/.

NICER Mission Overview (2023). https://heasarc.gsfc.nasa.gov/docs/nicer/nicer_about.html.

Abadi, M. et al. (2016). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. doi: 10.48550/arXiv.1603.04467. arXiv: 1603.04467 [cs].

Arévalo, P. and P. Uttley (2006). "Investigating a Fluctuating-Accretion Model for the Spectral-Timing Properties of Accreting Black Hole Systems". In: *Monthly Notices of the Royal Astronomical Society* 367.2, pp. 801–814. doi: 10.1111/j.1365-2966.2006.09989.x.

Arnaud, K. A. (1996). "XSPEC: The First Ten Years". In: 101, p. 17.

Atri, P. et al. (2020). "A Radio Parallax to the Black Hole X-ray Binary MAXI J1820+070". In: *Monthly Notices of the Royal Astronomical Society: Letters* 493.1, pp. L81–L86. doi: 10.1093/mnrasl/slaa010. arXiv: 1912.04525.

Axelsson, M. and A. Veledina (2021). "Accretion Geometry of the Black Hole Binary MAXI J1820+070 Probed by Frequency-Resolved Spectroscopy". In: *Monthly Notices of the Royal Astronomical Society* 507.2, pp. 2744–2754. doi: 10.1093/mnras/stab2191. arXiv: 2103.08795.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. 1st ed. 2006. Corr. 2nd printing 2011 . New York: Springer.

Brinkman, A. C. et al. (1974). "Correlation Analysis of X-Ray Emission from Cygnus X-1". In: *The Astrophysical Journal* 188, pp. 603–608. doi: 10.1086/152753.

Buisson, D. J. et al. (2019). "Maxi J1820+070 with Nustar I. An Increase in Variability Frequency but a Stable Reflection Spectrum: Coronal Properties and Implications for the Inner Disc in Black Hole Binaries". In: *Monthly Notices of the Royal Astronomical Society* 490.1, pp. 1350–1362. doi: 10.1093/mnras/stz2681. arXiv: 1909.04688.

Burrows, D. N. et al. (2005). "The Swift X-Ray Telescope". In: *Space Science Reviews* 120.3, pp. 165–195. doi: 10.1007/s11214-005-5097-2.

De Marco, B. et al. (2021). "The Inner Flow Geometry in MAXI J1820+070 during Hard and Hard-Intermediate States". In: *Astronomy and Astrophysics* 654, pp. 1–20. doi: 10.1051/0004-6361/202140567. arXiv: 2102.07811.

Dillon, J. V. et al. (2017). *TensorFlow Distributions*. doi: 10.48550/arXiv.1711.10604. arXiv: 1711.10604 [cs, stat].

Done, C. et al. (2007). "Modelling the Behaviour of Accretion Flows in X-ray Binaries". In: *The Astronomy and Astrophysics Review* 15.1, pp. 1–66. doi: 10.1007/s00159-007-0006-1.

Duane, S. et al. (1987). "Hybrid Monte Carlo". In: *Physics Letters B* 195.2, pp. 216–222. doi: 10.1016/0370-2693(87)91197-X.

Esin, A. A. et al. (1997). "Advection-Dominated Accretion and the Spectral States of Black Hole X-Ray Binaries: Application to Nova Muscae 1991". In: *The Astrophysical Journal* 489.2, p. 865. doi: 10.1086/304829.

Fabian, A. C. et al. (2020). "The Soft State of the Black Hole Transient Source MAXI J1820+070: Emission from the Edge of the Plunge Region?" In: *Monthly Notices of the Royal Astronomical Society* 493.4, pp. 5389–5396. doi: 10.1093/mnras/staa564.

Fabian, A. C. et al. (1989). "X-Ray Fluorescence from the Inner Disc in Cygnus X-1." In: *Monthly Notices of the Royal Astronomical Society* 238, pp. 729–736. doi: 10.1093/mnras/238.3.729.

Gandhi, P. et al. (2019). "Gaia Data Release 2 Distances and Peculiar Velocities for Galactic Black Hole Transients". In: *Monthly Notices of the Royal Astronomical Society* 485.2, pp. 2642–2655. doi: 10.1093/MNRAS/STZ438.

Gendreau, K. C. et al. (2012). "The Neutron Star Interior Composition ExploreR (NICER): An Explorer Mission of Opportunity for Soft x-Ray Timing Spectroscopy". In: *Space Telescopes and Instrumentation 2012: Ultraviolet to Gamma Ray*. Vol. 8443. SPIE, pp. 322–329. doi: 10.1117/12.926396.

Gendreau, K. C. et al. (2016). "The Neutron Star Interior Composition Explorer (NICER): Design and Development". In: *Space Telescopes and Instrumentation 2016: Ultraviolet to Gamma Ray* 9905.July 2016, 99051H. doi: 10.1117/12.2231304.

Gilfanov, M. (2010). "X-Ray Emission from Black-Hole Binaries". In: *The Jet Paradigm: From Microquasars to Quasars*. Ed. by T. Belloni. Lecture Notes in Physics. Berlin, Heidelberg: Springer, pp. 17–51. doi: 10.1007/978-3-540-76937-8_2.

Hamilton, J. D. (1994). *Time Series Analysis*. Princeton, N.J: Princeton Univ Pr.

Harrison, F. A. et al. (2013). "THE NUCLEAR SPECTROSCOPIC TELESCOPE ARRAY (NuSTAR) HIGH-ENERGY X-RAY MISSION". In: *The Astrophysical Journal* 770.2, p. 103. doi: 10.1088/0004-637X/770/2/103.

Harris, C. R. et al. (2020). "Array Programming with NumPy". In: *Nature* 585.7825, pp. 357–362. doi: 10.1038/s41586-020-2649-2.

Hastings, W. K. (1970). "Monte Carlo Sampling Methods Using Markov Chains and Their Applications". In: *Biometrika* 57.1, pp. 97–109. doi: 10.1093/biomet/57.1.97.

Hoffman, M. D. and A. Gelman (2014). "The No-U-turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo". In: *The Journal of Machine Learning Research* 15.1, pp. 1593–1623.

Hunter, J. D. (2007). "Matplotlib: A 2D Graphics Environment". In: *Computing in Science & Engineering* 9.3, pp. 90–95. doi: 10.1109/MCSE.2007.55.

Kalemci, E. et al. (2022). "Black Holes: Timing and Spectral Properties and Evolution". In: *Handbook of X-ray and Gamma-ray Astrophysics*. Ed. by C. Bambi and A. Santangelo. Singapore: Springer Nature, pp. 1–43. doi: 10.1007/978-981-16-4544-0_100-1.

Kara, E. et al. (2019). "The Corona Contracts in a Black-Hole Transient". In: *Nature* 565.7738, pp. 198–201. doi: 10.1038/s41586-018-0803-x. arXiv: 1901.03877.

Kawamuro, T. et al. (2018). "MAXI/GSC Detection of a Probable New X-ray Transient MAXI J1820+070". In: *The Astronomer's Telegram* 11399, p. 1.

Kawamura, T. et al. (2022). "A Full Spectral-Timing Model to Map the Accretion Flow in Black Hole Binaries: The Low/Hard State of MAXI J1820+070". In: *Monthly Notices of the Royal Astronomical Society* 511.1, pp. 536–552. doi: 10.1093/mnras/stac045.

Kitagawa, G. and W. Gersch (1996). *Smoothness Priors Analysis of Time Series*. Ed. by P. Bickel et al. Vol. 116. Lecture Notes in Statistics. New York, NY: Springer. doi: 10.1007/978-1-4612-0761-0.

Kitagawa, G. (2022). *Introduction to Time Series Modeling with Applications in R*. 2 . Boca Raton London New York: Chapman and Hall/CRC.

Kotov, O. et al. (2001). "On the X-ray Time-Lags in the Black Hole Candidates". In: *Monthly Notices of the Royal Astronomical Society* 327.3, pp. 799–807. doi: 10.1046/j.1365-8711.2001.04769.x. arXiv: astro-ph/0103115.

Lewandowski, D. et al. (2009). "Generating Random Correlation Matrices Based on Vines and Extended Onion Method". In: *Journal of Multivariate Analysis* 100.9, pp. 1989–2001. doi: 10.1016/j.jmva.2009.04.008.

Ljung, G. M. and G. E. P. Box (1978). "On a Measure of Lack of Fit in Time Series Models". In: *Biometrika* 65.2, pp. 297–303. doi: 10.1093/biomet/65.2.297.

Lütkepohl, H. (2005). *New Introduction to Multiple Time Series Analysis*. Berlin, Heidelberg: Springer. doi: 10.1007/978-3-540-27752-1.

Matsuoka, M. et al. (2009). "The MAXI Mission on the ISS: Science and Instruments for Monitoring All-Sky X-Ray Images". In: *Publications of the Astronomical Society of Japan* 61.5, pp. 999–1010. doi: 10.1093/pasj/61.5.999.

Miyamoto, S. et al. (1988). "Delayed Hard X-rays from Cygnus X-l". In: *Nature* 336.6198, pp. 450–452. doi: 10.1038/336450a0.

Mizumoto, M. et al. (2019). "X-Ray Reverberation Lags of the Fe-K Line Due to AGN Disc Winds". In: *MNRAS* 482.4, pp. 5316–5326. doi: 10.1093/mnras/sty3056. arXiv: 1808.06625.

Nasa High Energy Astrophysics Science Archive Research Center (Heasarc) (2014). "HEAsoft: Unified Release of FTOOLS and XANADU". In: *Astrophysics Source Code Library*, ascl:1408.004.

Oda, M. et al. (1971). "X-Ray Pulsations from Cygnus X-1 Observed from UHURU". In: *The Astrophysical Journal* 166, p. L1. doi: 10.1086/180726.

Oda, M. (1974). "On Cyg X-1". In: 67.3, p. 80.

Oda, M. (1977). "CygX-1/A Candidate of the Black Hole". In: *Space Science Reviews* 20.6, pp. 757–813. doi: 10.1007/BF02431835.

Okajima, T. et al. (2016). "Performance of NICER Flight X-Ray Concentrator". In: *Space Telescopes and Instrumentation 2016: Ultraviolet to Gamma Ray*. Vol. 9905. SPIE, pp. 1495–1501. doi: 10.1117/12.2234436.

Omama, T. et al. (2023). "X-Ray Time Lag Evaluation of MAXI J1820+070 with a Differential Cross-correlation Analysis". In: *The Astrophysical Journal* 945.2, p. 92. doi: 10.3847/1538-4357/acba00.

Pena, D. and G. E. P. Box (1987). "Identifying a Simplifying Structure in Time Series". In: *Journal of the American Statistical Association* 82.399, pp. 836–843. doi: 10.2307/2288794. JSTOR: 2288794.

Pottschmidt, K. et al. (1998). "Analyzing Short-Term X-ray Variability of Cygnus X-1 with Linear State Space Models". In: *Astronomy and Astrophysics* 334, pp. 201–209. doi: 10.48550/arXiv.astro-ph/9802339.

Poutanen, J. (2002). "Impact of Reverberation in Flared Accretion Discs on Temporal Characteristics of X-ray Binaries". In: *Monthly Notices of the Royal Astronomical Society* 332.2, pp. 257–270. doi: 10.1046/j.1365-8711.2002.05272.x.

Prigozhin, G. et al. (2016). "NICER Instrument Detector Subsystem: Description and Performance". In: *Space Telescopes and Instrumentation 2016: Ultraviolet to Gamma Ray*. Vol. 9905. SPIE, pp. 436–447. doi: 10.1117/12.2231718.

Remillard, R. A. and J. E. McClintock (2006). "X-Ray Properties of Black-Hole Binaries". In: *Annual Review of Astronomy and Astrophysics* 44, pp. 49–92. doi: 10.1146/annurev.astro.44.051905.092532. arXiv: astro-ph/0606352.

Salvatier, J. et al. (2015). *Probabilistic Programming in Python Using PyMC*. doi: 10.48550/arXiv.1507.08050. arXiv: 1507.08050 [stat].

Seabold, S. and J. Perktold (2010). "Statsmodels: Econometric and Statistical Modeling with Python". In: *Proceedings of the 9th Python in Science Conference*, pp. 92–96. doi: 10.25080/Majora-92bf1922-011.

Shidatsu, M. et al. (2018). "X-Ray, Optical, and Near-infrared Monitoring of the New X-Ray Transient MAXI J1820+070 in the Low/Hard State". In: *The Astrophysical Journal* 868.1, p. 54. doi: 10.3847/1538-4357/aae929.

Shidatsu, M. et al. (2019). "X-Ray and Optical Monitoring of State Transitions in MAXI J1820+070". In: *The Astrophysical Journal* 874.2, p. 183. doi: 10.3847/1538-4357/ab09ff. arXiv: 1903.01686.

Stan Development Team (2024). *Stan Modeling Language Users Guide and Reference Manual*. //mc-stan.org/users/documentation/.

Torres, M. A. P. et al. (2020). "The Binary Mass Ratio in the Black Hole Transient MAXI J1820+070". In: *The Astrophysical Journal* 893.2, p. L37. doi: 10.3847/2041-8213/ab863a. arXiv: 2003.02360.

Uttley, P. et al. (2011). "The Causal Connection between Disc and Power-Law Variability in Hard State Black Hole X-ray Binaries". In: *Monthly Notices of the Royal Astronomical Society: Letters* 414.1, pp. L60–L64. doi: 10.1111/j.1745-3933.2011.01056.x. arXiv: 1104.0634.

Uttley, P. et al. (2014). "X-Ray Reverberation around Accreting Black Holes". In: *Astronomy and Astrophysics Review* 22.1, pp. 1–66. doi: 10.1007/S00159-014-0072-0/FIGURES/32. arXiv: 1405.6575.

Uttley, P. et al. (2018). "NICER Observations of MAXI J1820+070 Suggest a Rapidly-Brightening Black Hole X-ray Binary in the Hard State". In: *The Astronomer's Telegram* 11423, p. 1.

van der Klis, M. et al. (1987). "The Complex Cross-Spectra of Cygnus X-2 and GX 5-1". In: *The Astrophysical Journal* 319, p. L13. doi: 10.1086/184946.

van der Klis, M. (2006). "Rapid X-ray Variability". In: *Compact Stellar X-ray Sources*. Ed. by M. van Der Klis and W. Lewin. Cambridge Astrophysics. Cambridge: Cambridge University Press, pp. 39–112. doi: 10.1017/CBO9780511536281.003.

Virtanen, P. et al. (2020). "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17.3, pp. 261–272. doi: 10.1038/s41592-019-0686-2.

Waskom, M. L. (2021). "Seaborn: Statistical Data Visualization". In: *Journal of Open Source Software* 6.60, p. 3021. doi: 10.21105/joss.03021.

West, M. and J. Harrison (1989). *Bayesian Forecasting and Dynamic Models*. Springer Series in Statistics. New York, NY: Springer. doi: 10.1007/978-1-4757-9365-9.

Zdziarski, A. A. et al. (2021). "Accretion Geometry in the Hard State of the Black Hole X-Ray Binary MAXI J1820+070". In: *The Astrophysical Journal Letters* 909.1, p. L9. doi: 10.3847/2041-8213/abe7ef. arXiv: 2101.04482.

Revision History

Table A.1: Version history

Date	Version	Change points
2024/01/10	2.0	First submission
2024/02/02	2.1	Revised based on the examiners' comments
2024/03/08	3.0	Final version