UNIVERSITY OF OXFORD

FINAL YEAR PROJECT

Exoplanet Detection in Large Astronomical Data Sets

Author:
Adam Cobb

Supervisor:
Prof. Stephen Roberts

Trinity 2015

FINAL HONOUR SCHOOL OF
ENG / EEM (delete as appropriate)

DECLARATION OF AUTHORSHIP

You should complete this certificate. It should be bound into your fourth year project report, immediately after your title page. Three copies of the report should be submitted to the Chairman of examiners for your Honour School, c/o Clerk of the Schools, examination Schools, High Street, Oxford.

Name (in capitals): ……

College (in capitals): …… **Supervisor:** ……

Title of project (in capitals): ……

Page count (excluding risk and COSHH assessments): ……

Please tick to confirm the following:

I have read and understood the University's disciplinary regulations concerning conduct in examinations and, in particular, the regulations on plagiarism (*Essential Information for Students. The Proctors' and Assessor's Memorandum*, Section 9.6; also available at www.admin.ox.ac.uk/proctors/info/pam/section9.shtml). ☐

I have read and understood the Education Committee's information and guidance on academic good practice and plagiarism at www.admin.ox.ac.uk/edc/goodpractice. ☐

The project report I am submitting is entirely my own work except where otherwise indicated. ☐

It has not been submitted, either partially or in full, for another Honour School or qualification of this University (except where the Special Regulations for the subject permit this), or for a qualification at any other institution. ☐

I have clearly indicated the presence of all material I have quoted from other sources, including any diagrams, charts, tables or graphs. ☐

I have clearly indicated the presence of all paraphrased material with appropriate references. ☐

I have acknowledged appropriately any assistance I have received in addition to that provided by my supervisor. ☐

I have not copied from the work of any other candidate. ☐

I have not used the services of any agency providing specimen, model or ghostwritten work in the preparation of this thesis/dissertation/extended essay/assignment/project/other submitted work. (See also section 2.4 of Statute XI on University Discipline under which members of the University are prohibited from providing material of this nature for candidates in examinations at this University or elsewhere: http://www.admin.ox.ac.uk/statutes/352-051a.shtml#_Toc28142348.) ☐

The project report does not exceed 50 pages (including all diagrams, photographs, references and appendices). ☐

I agree to retain an electronic copy of this work until the publication of my final examination result, except where submission in hand-written format is permitted. ☐

I agree to make any such electronic copy available to the examiners should it be necessary to confirm my word count or to check for plagiarism. ☐

Candidate's signature: …….. **Date:** ……..

Generic Display Screen Equipment Risk Assessment		
In Building: IEB/ Personal Computer		
Assessment undertaken by: Adam Cobb	**Signed**:	**Date:**
Assessment supervisor: Prof. Stephen Roberts	**Signed:**	**Date:**

Hazard	**Persons at Risk**	**Risk Controls In Place**	**Further Action Necessary To Control Risk**
Eyestrain/ Headaches	User	**Take regular breaks every hour**. - undertake a different task. - adjust screen location to prevent glare or bright reflections. - Angle screen downwards to prevent reflection. - ensure no screen flicker. - ensure screen surface is clean. - ensure lighting is adequate for the task. - have an eye test if problems persist. - close blinds to prevent glare (as appropriate)	Consult Supervisor and advise Departmental Safety Officer (DSO) if problems persist. Please refer to the following link for a picture of good posture: http://www.hse.gov.uk/pubns/indg36.pdf
Back pain	User	**Ensure Workplace is correctly set up** - e.g. height of chair needs to be set so that forearms are parallel to desk. - ensure good posture at all times, sitting upright or slightly reclining. - Lower back supported to maintain natural curves.	Refer any medical issues to Supervisor or Departmental Safety Officer (DSO)
Aching shoulders, wrists	User	**Check seat height is correct** - forearms horizontal, level with top of desk. - keep wrists straight, use wrist rest. - No overreaching, exercise muscles. - Arms relaxed by side.	Refer any medical issues to Supervisor or Departmental Safety Officer (DSO)
Aching neck	User	**Check screen height is correct** - eyes level with top of screen. - use document holder. - exercise muscles. - Check chair height e.g. forearms horizontal, level with top of desk	

Aching legs	User	**Check space under desk** to stretch legs, feet rest comfortably on floor otherwise get footrest. - exercise muscles. - Knees level with pelvis or slightly below. - Feet flat on the floor or use a footrest.	Remove items under desk which are preventing correct use e.g. boxes.
Water/Liquids	User	Please ensure that no liquids are sat on your hard drive or near to your monitor.	Building Inspections.
240 VAC Electrical shock	User	User to check that all electrical leads to their PC are in good working order. Contact Electronics (Thom 5th floor) if Portable Appliance Label 'out of date' or not visible.	Supervisor/Student to check validity of PAT test label.

Acknowledgements

I would like to thank Professor Stephen Roberts for all the guidance and enthusiasm put into the supervision of this fourth year project, which has made it extremely interesting and stimulating. This has inspired me to further my education in this area by applying for a DPhil.

I would also like to thank him for introducing me to the Astrostatistics group at Oxford whose weekly sessions have provided a great platform to discuss my project along with other people's current research.

I also thank Professor Stephen Roberts for some sections of code on the Gaussian process and Dr. Suzanne Aigrain for her 'Box Least Squares' code used in chapter 5.

Finally I would also like to express my gratitude to friends and family for their support during this project.

Abstract

This report demonstrates an alternative way of pre-processing Kepler Mission data that is better suited for exoplanet detection over large datasets. The results presented show how the developed pipeline is able to remove systematic trends using a Gaussian Process to model the underlying behaviour of a star's flux. This has enabled transit detection algorithms to be applied to the entire dataset consisting of 3.5 years of observations, rather than limiting the application to approximately three months of data at a time. This opens the possibility of being able to search for exoplanets with longer periods with a higher level of success. In particular, Kepler Objects of Interest 3, 87, 157, 172, 701 and 961, which all contain transiting exoplanets in the data, have been studied to develop and demonstrate these methods.

Furthermore, an investigation into the use of the autocorrelation as prior knowledge on the length of the periods of possible exoplanets has also been explored. This is with the aim of demonstrating how it can be combined with NASA's current Box-Least Squares algorithm to give a higher level of performance.

Contents

Chapter 1

Introduction

1.1 Background

Research in the area of extrasolar planets has gained a considerable amount of momentum since the first confirmed exoplanet discovery of *51 Pegasi b* in 1995 [1]. Now twenty years later, 1,821 extrasolar planets have been confirmed[1] according to the NASA Exoplanet Archive [2]. This surge in discoveries has been dramatically accelerated since NASA launched its Kepler Telescope on 6 March 2009 [3]. So far, the addition of Kepler has led to the discovery of 1,022 confirmed exoplanets, along with 3185 unconfirmed candidates.

In the era before Kepler, and up until October 2009, the most common way of detecting planets came through the radial velocity technique [4], as was used for *51 Pegasi b*. This technique is used to indirectly detect planets by measuring the radial velocity of the host star as it 'executes a small reflex orbit' [4] around the centre of mass of the celestial system. The centre of mass is also known as the barycentre. The radial velocity of the host star causes it to 'wobble' back and forth from the observer allowing the velocity to be measured through the Doppler shift of the star's light. The limits of this technique are pointed out on page 21 of Perryman's book [5]. It states that 'signal-to-noise considerations limit observations to brighter stars' and that it 'favour[s] the detection of massive planets' which are close to the host star.

The comparatively simple transit detection technique is now currently the most successful method and is the one used for the Kepler Mission. It relies on spotting the periodic dip in the brightness or flux of a star when a planet crosses the line of sight between the observer and the host star. Figure 1.1 demonstrates how the expected behaviour of the brightness of a star varies as an orbiting planet moves across the line of sight. One disadvantage of this technique comes from the high probability

[1] As of the 5th March 2015

that the transiting planet's orbit does not cross the line of sight from the view of the observer. Despite the drawback, this technique has seen a high level of success, implying that exoplanets must be a relatively common phenomena. The first example of a continuous light curve exhibiting this behaviour occurred in 1999 using the STARE telescope [4].

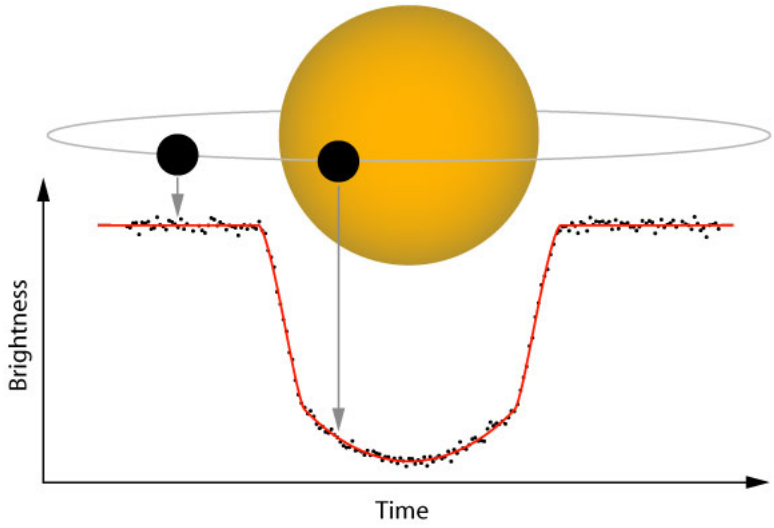


Figure 1.1: A diagram to display how the flux of a star varies due to a transiting planet [6].

There are, of course, other less widely used methods for planet detection such as astrometry, timing and microlensing. This report solely concerns itself with Kepler Mission data and because this uses the transit detection technique no further details will be given on these methods. The reader is encouraged to refer to Perryman's book 'The Exoplanet Handbook' [5] for further reading on these techniques.

1.2 The Kepler Mission

The launch of the Kepler Mission has opened up a vast quantity of large astronomical datasets that require analysing. This has produced the more general associated challenges of dealing with big data which will present itself in this report. The data used in this project comes from the Kepler Mission, for which a background summary is provided below.

The Kepler Spacecraft is in 'a 372.5 day Earth-trailing heliocentric orbit' [7] and has a field of view which encompasses approximately 156,000 target stars [8]. The flux of these stars is observed by a photometer, and the data is then stored ready to be downloaded once a month [8]. The data downloaded has a certain structure which is going to be very significant for the rest of the report. First of all, each data point collected by the Kepler telescope is sampled once every '29.4 minutes' [8] and second of all, the data is split up into quarters. A quarter contains about three months of data

and it is organised this way due to the need for the Spacecraft to 'roll 90 degrees about its axis every 93 days' [7]. This is to ensure the photovoltaic cells are facing the sun.

The paper from the Astrophysical Journal Letters by Jenkins et al. describes in detail the Kepler processing pipeline. It explains how the raw data is processed into the corrected light curves which are then suitable to be operated on by planet detection algorithms such as the one being developed in this report. The final process which takes place before applying a 'Transiting Planet Search' (TPS) is the 'Presearch Data Conditioning' (PDC) stage [8]. The task here is to remove certain systematic errors such as 'pointing errors, focus changes and thermal effects on instrument properties' [8]. Therefore this explains that any mention of the term 'PDC flux' is simply data that has been through this stage and is referred to as the 'original data' for the purposes of this report.

The data analysed in this report is the PDC flux. Therefore this report aims to improve upon NASA's TPS pipeline. Access to the Kepler data has been made possible through 'kplr' which is 'a Python interface to the Kepler data' [9]. The PDC flux for each star was then converted into MATLAB.

1.2.1 Kepler-2.0 Mission (K2)

In May 2013 the second of the four reaction wheels failed [10]. The Kepler telescope was thereafter no longer able to maintain its position pointing at the same field of view (FOV) that it had been facing during the first Kepler Mission. Therefore the K2 mission had to be instigated on a FOV that involves positioning the telescope so that it is balanced in the ecliptic plane. This has been deemed the most stable position to allow it to continue its mission of discovering exoplanets, whilst achieving the required level precision. The access to two reaction wheels means that one of the three principal axes, the roll axis, must be controlled by the thruster. Therefore the data collected from this scientific campaign will suffer from even more serious systematics which could benefit from prior trend removal.

1.3 Overview

The context is now in place to introduce the objectives and scope of this report. The ultimate aim of this project has been to develop a flexible, computationally practical method for robust exoplanet discovery and apply it to large astronomical data sets. This can be split into three main subsidiary goals:

- Develop an algorithm that removes stellar variability and other trends corresponding to unwanted systematics.

- Expand this algorithm to have the capability of analysing and joining together the large datasets collected by the Kepler telescope.
- Apply current methods to the newly preprocessed data and develop improvements for a computationally practical method for exoplanet discovery.

Therefore this report has been structured in a way that has allocated a chapter to each one of these aims. Chapter 3 tackles the problem of removing stellar variability and then chapter 4 evolves this algorithm into dealing with the large datasets of the Kepler Mission. Finally chapter 5 explores current techniques for planet discovery and then moves on into developing improvements on current exoplanet discovery methodology. However chapter 2, which introduces the Gaussian process, must precede the aforementioned chapters due to the vital role it plays in the overall pipeline.

The reason for setting out these goals and the necessity of researching this area is highlighted in a very recent paper on the Kepler Mission data, 'Detection of Potential Transit Signals in 17 Quarters of Kepler Mission Data' [10], which was released 15 January 2015. In the paper it states that when doing an initial transit search 'we do not detrend the data in any way prior to a search'. In this report it will be shown that removing trends in the data prior to transit detection does offer improvements for detecting planets. It should also be pointed out that the data analysed in this report encompasses all 18 quarters rather than the 17 analysed in the paper. The paper has avoided using 'Quarter 0' (Q0) as it was deemed 'too short to avoid undesirable effects in the transit search'. However it will be demonstrated that this problem has been avoided in this report by applying the transit detecting algorithm over the entire data set rather than looking at one quarter at a time.

The use of Gaussian processes for exoplanet detection has only previously been used on a relatively small data set in the paper 'Precise time series photometry for the Kepler-2.0 mission' [11]. The reason for applying this more powerful method has come from the need of removing the badly behaved trends of the new K2 Mission. The success of using the Gaussian process on a small set of data demonstrated in the paper, has helped lead to using the Gaussian process for this report. However it has had to be adapted for use on much larger datasets.

The nature of the methods described in this report were developed in mind of the K2 mission which struggles with systematic trends in the data. This is due to the need to maintain its pointing position with the aid of the thrusters rather than the original intended reaction wheels that have had failures. This report should provide an improved technique for planet detection over large datasets in the presence of noisy systematic trends.

Chapter 2

The Gaussian Process

2.1 Introduction

This chapter gives an introduction to the Gaussian process (GP) within the context of this report. The aim is to give the reader a basic overview of this method and describe the key concepts. In order to receive a more detailed understanding of the Gaussian process, both Rasmussen's book [12] and the paper 'Gaussian processes for time-series modelling' [13] are very good references for going deeper into the methodology.

The GP will be an essential part of the signal processing throughout this report. It will be used to do one dimensional regression to fit to the time dependent flux of stars. The Gaussian process is designated as part of 'Bayesian non-parametric' modelling [13]. It is defined as a collection of random variables, of which any finite number have a joint Gaussian distribution [14]. This distribution is fully specified by a mean and covariance function which enable prior knowledge of the general behaviour of the data to be incorporated into the model. This is without describing the exact functional form of the model. The GP makes it possible to accommodate for uncertainty within the data, as demonstrated in section 2.3 when discussing the covariance function. Therefore the motivation for using the Gaussian process comes from being able to take into account prior knowledge of the behaviour of a light curve and use that to fit to the prevalent systematics by selecting appropriate mean and covariance functions. For example, if the only prior information about a light curve is that it is smooth and that the flux varies quickly over a short time scale, a GP would be able to incorporate this knowledge when fitting to the data, without limiting the model to a specific functional form.

Formally the Gaussian process can be written as a multivariate Gaussian.

$$p(\mathbf{y}(\mathbf{x})) = \mathcal{N}(\mathbf{m}(\mathbf{x}), \mathbf{K}(\mathbf{x}, \mathbf{x}')). \tag{2.1}$$

This equation along with equations 2.2, 2.3 and 2.4 have been taken from page 6 of [13] where $\mathbf{m}(\mathbf{x})$ is the mean function, $\mathbf{K}(\mathbf{x}, \mathbf{x}')$ is the covariance matrix and $\mathbf{y} = y_1, y_2, ..., y_n$ are the dependent function values, evaluated at $\mathbf{x} = x_1, x_2, ..., x_n$.

Having set up this multivariate Gaussian that defines the GP, we can use this distribution to make predictions at some 'test datum x_*' given all the other observations. This introduces the following augmented Gaussian distribution:

$$p\left(\begin{array}{c}\mathbf{y}\\ y_*\end{array}\right) = \mathcal{N}\left(\left[\begin{array}{c}\mathbf{m}(\mathbf{x})\\ m(x_*)\end{array}\right], \left[\begin{array}{cc}\mathbf{K}(\mathbf{x},\mathbf{x}) & \mathbf{K}(\mathbf{x},x_*)\\ \mathbf{K}(x_*,\mathbf{x}) & \mathbf{K}(x_*,x_*)\end{array}\right]\right). \tag{2.2}$$

This then leads to the posterior distribution over y_* to being Gaussian with a mean and variance given by [13]

$$\mu_* = m(x_*) + \mathbf{K}(x_*, \mathbf{x})\mathbf{K}(\mathbf{x}, \mathbf{x})^{-1}(\mathbf{y} - \mathbf{m}(\mathbf{x})), \tag{2.3}$$

$$\sigma_* = \mathbf{K}(x_*, x_*) + \mathbf{K}(x_*, \mathbf{x})\mathbf{K}(\mathbf{x}, \mathbf{x})^{-1}\mathbf{K}(\mathbf{x}, x_*). \tag{2.4}$$

Of course, this can be extended from a single test datum x_* to multiple values outside the set of observations. This is done by rewriting equations 2.2, 2.3 and 2.4 and replacing x_* and y_* with the vectors $\mathbf{x}_*$ and $\mathbf{y}_*$.

2.2 Mean Function

The inferred mean function will play a vital role when applying a GP for this report. Like the covariance function described in the next section, there are many options available for choosing an appropriate mean function. As an example if there is prior knowledge that a certain dataset decays linearly with time then this can be included in the form of a linear mean function. The mean function is what the Gaussian process will predict as the most likely function in regions where the data is sparse. This property is demonstrated in figure 2.1 where a zero mean function is used. The solid red line, which is the predicted mean function, briefly reverts to its prior zero mean in the region where there are no observations. The measure of uncertainty in this area of the plot is shown by the expansion of the two standard deviation zone, represented by the shaded region.

In this report, the data will always have its mean removed before the Gaussian process is applied and then reinserted afterwards. There is also no further prior information available about the behaviour of the star. Therefore it has been deemed appropriate to use a zero mean function.

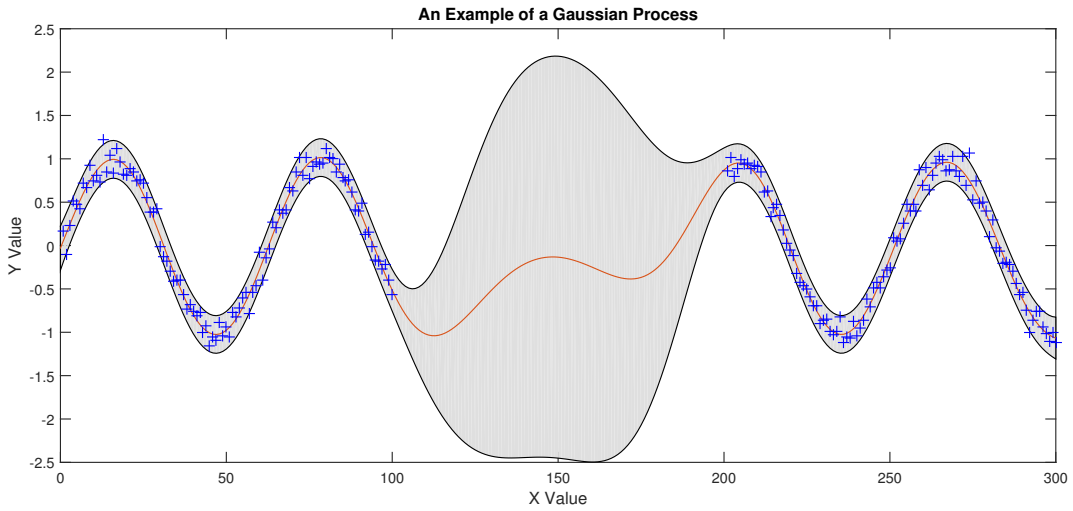


Figure 2.1: An example of Gaussian process regression on data containing a sinusoid wave with additive Gaussian noise. A sparse region with no observations has been included to demonstrate how the GP copes.

2.3 Covariance Function

The covariance is central to Gaussian process inference. The choice of covariance comes from what is known about how it is expected the data will behave. In order to determine an appropriate choice for the covariance function, prior information about certain attributes of the data should be taken into account. For example, if it is known that the data is periodic and smooth, then these characteristics ought to be included in the covariance function.

The *covariance kernel function* $k(x, x)$ must be symmetric positive semi-definite and forms the $n \times n$ covariance matrix

$$\mathbf{K}(\mathbf{x}, \mathbf{x}) = \begin{pmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \cdots & k(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \cdots & k(x_n, x_n) \end{pmatrix}. \tag{2.5}$$

One of the most common covariances used is the squared exponential (SE). That follows the idea that points near each other are much more closely related than ones further away. The relationship between two points x_i and x_j is governed by exponential decay [13].

$$k(x_i, x_j) = h^2 \exp\left[-\left(\frac{x_i - x_j}{\lambda}\right)^2\right] \tag{2.6}$$

The hyperparameters h and λ correspond to the output scale length and the input scale length respectively. The output scale length sets the initial value for the expected magnitude of the amplitude.

Since all data is normalised before applying the GP, h has been initially set to 1 for the entirety of the report. The input scale length specifies how strong the data is correlated in the x-direction. Deciding upon a suitable initial input scale length is a significant part of chapters 3 and 4, when looking at including the autocorrelation as an aid for achieving this.

So far the the GP has been developed assuming that the observations are noiseless. This is unreasonable in almost all physical systems and therefore the covariance matrix formed from the kernel will include the addition of uncorrelated Gaussian noise.

$$\mathbf{K} = \mathbf{K}_{noiseless} + \sigma_n^2 \mathbf{I} \tag{2.7}$$

Where σ_n is a hyperparameter that will account for the noise as well as improving the condition of the covariance matrix, which will be significant for numerically stable matrix inversion. High values in σ_n allows the GP to explain a larger amount of the data as being the product of white noise, whereas a small σ_n automatically puts a higher level of confidence over the observations[1].

The reason for selecting the squared exponential is because it assumes smooth variations within the data [13]. This is precisely what is required as the GP will be used for removing smooth trends in the data. Therefore the final choice for the covariance function takes the form of the squared exponential with additive white noise. There are many other covariances not mentioned here that fit the necessary symmetric, semi-definite conditions. It also ought to be stated that covariances can be combined through addition and multiplication in order to create covariance functions that better capture any known inherent behaviour of data. The details of which will not be discussed here, but the reader is referred to chapter 4 of [12] for a detailed discussion.

2.4 Inferring the Hyperparameters

The reason for using the term 'initial' in the previous section for describing the initial hyperparameters comes from the fact these values are only the starting conditions before optimising over a function to try and infer a more suitable set of hyperparameters.

The function to be optimised over to find this set is the log marginal likelihood in equation 2.8. The derivation of which can be found in chapter 2 of Rasmussen's book.

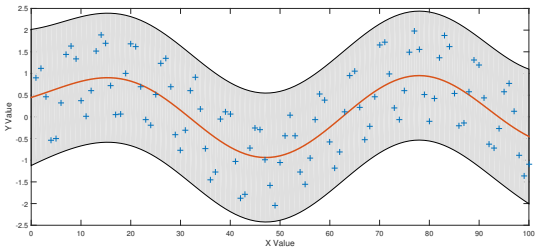
$$\log p(\mathbf{y}|\mathbf{X}, \theta) = -\frac{1}{2}\mathbf{y}^\top \mathbf{K}_y^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K}_y| - \frac{n}{2}\log 2\pi. \tag{2.8}$$

[1]This noise hyperparameter has been initialised as 0.1 in this project

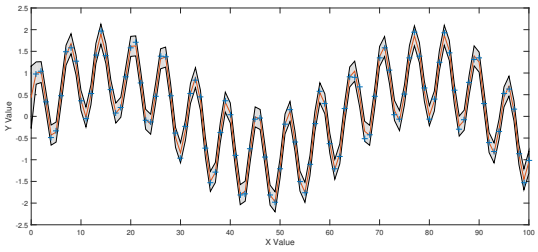
Maximising equation 2.8 by taking partial derivatives with respect to the hyperparameters θ gives equation 2.9 which enables the application of conjugate gradient descent to set the hyperparameters [12, Chapter 5].

$$\begin{aligned}\frac{\partial}{\partial\theta_j}\log p(\mathbf{y}|\mathbf{X},\theta) &= \frac{1}{2}\mathbf{y}^\top\mathbf{K}^{-1}\frac{\partial\mathbf{K}}{\partial\theta_j}\mathbf{K}^{-1}\mathbf{y} - \frac{1}{2}tr(\mathbf{K}^{-1}\frac{\partial\mathbf{K}}{\partial\theta_j}) \\ &= \frac{1}{2}tr\left((\alpha\alpha^\top - \mathbf{K}^{-1})\frac{\partial\mathbf{K}}{\partial\theta_j}\right) \text{ where } \alpha = \mathbf{K}^{-1}\mathbf{y}.\end{aligned} \tag{2.9}$$

Unfortunately the log marginal likelihood is not a convex function. This means optimisation can lead to getting trapped in local minima, which can cause the selection of inappropriate hyperparameters. Therefore selecting initial values for the hyperparameters that encourage the global minimum of the log marginal likelihood to be found is an important part of the process. Figures 2.2a and 2.2b demonstrate the importance of the starting conditions of the hyperparameters, in this case varying the input scale length whilst keeping the other hyperparameters constant.



(a) Initial Input Scale Length 1.00 Optimised to 0.54



(b) Initial Input Scale Length 0.01 Optimised to 0.07

Figure 2.2: An example of how the initial choice of hyperparameters can modify the resulting GP. The true function is $y = \sin(x) + \sin(0.1x)$ with additive Gaussian noise. Figure 2.2a shows how picking a relatively large input scale length has caused the GP to only fit to the low frequency sinusoid component. Picking a much smaller value of 0.01 for 2.2b has allowed the GP to include the higher frequency component, hence giving a very good fit to the true function.

2.5 Gaussian Processes in MATLAB

Fortunately applying GPs in MATLAB has been made possible through an extensive toolbox that was derived for Rasmussen's book [15]. This toolbox has been used to produce all the GP plots in this

project.

One advantage that this toolbox offers is overcoming the difficulty in solving equations 2.3, 2.4 and 2.8. These equations can be computationally challenging due to the expensive nature of inverting the $n \times n$ covariance matrix $\mathbf{K}^{-1}$. Under normal circumstances this would be of $\mathcal{O}(n^3)$, however the GPML MATLAB toolbox uses the Cholesky decomposition to reduce the complexity to $\mathcal{O}(\frac{1}{3}n^3)$ [16] and improve the stability of the operation.

2.6 Summary

We are now in a position to apply the Gaussian process in order to solve the problem at hand. This chapter has given an overview of the Gaussian process and introduced the main formulae. The mean and covariance functions have been explained, leading to the decision to go with a zero mean function and a SE covariance function with additive noise.

The challenge of getting trapped in local minima when optimising the log marginal likelihood to find the hyperparameters has been described. This issue will be dealt with again in more detail when it arises during the design of the pipeline in chapters 3 and 4. A comment was also made as to the computational difficulties of optimising the log marginal likelihood and calculating the posterior mean and covariance due to the cost of inverting a large matrix. It was then briefly explained how the MATLAB GPML toolbox tries to alleviate this issue through using the Cholensky decomposition. The next chapter starts the process of applying this theory to data containing flux from the Kepler Mission, where the challenges of applying a GP will be confronted to overcome the example specific problems.

Chapter 3

Gaussian Process Regression on a Single Quarter

3.1 Introduction

The objective of this chapter is to present an algorithm that will model the underlying behaviour of a star's brightness, when given a set of data consisting of the flux and the corresponding time. In particular the data analysed in this chapter only contains the values from Quarter 1 (Q1) of the collected Kepler Mission dataset. This is in comparison to chapter 4, where the quarters Q0-Q17 are considered.

Figure 3.1 displayed below is a good example of a GP applied to an eclipsing binary without treating the data beforehand. The grey shaded in area in the plot represents two standard deviations either side of the mean function which is the solid blue line. One thing to notice about the figure is that the Gaussian process has not managed to fit the data as a result of the extremely deep transits associated with the eclipsing binary.

The reason for presenting this diagram is to provide motivation for a method that can detect outliers and remove them in order to aid the GP in a more successful fit. Therefore the next section describes how this outlier removal will be achieved. The other challenge that will be dealt with is the requirement to vary one of the hyperparameters of the GP, the input scale length, in order to further ensure that the desired plot is achieved.

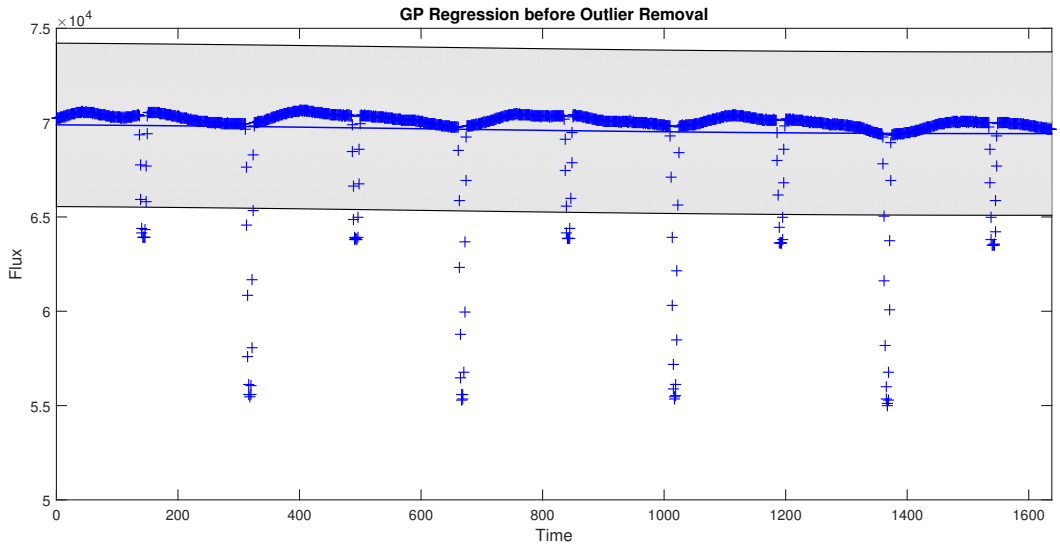


Figure 3.1: Star 1:A Star clearly showing the behaviour of an eclipsing binary.

3.2 Outlier Detection and Removal

As briefly touched upon in the previous section, applying a Gaussian process to the original data has not aided in modelling the underlying behaviour of the star because of the prevalence of large systematic events or outliers. In order to further investigate the light curves a method of removing these outliers must be implemented. There are two methods described here which both work in their own right but have been used sequentially as they combine to form a powerful outlier remover.

3.2.1 Removing Data Based on the Distance from the Predictive Mean

This idea closely follows the method described in section 3.3 of [11] where it is suggested that 'Any data points lying further than 3σ from the predictive mean are flagged as outliers'. Note that σ refers to one standard deviation. This is exactly the method used here but the threshold has been set to 2σ because a more stringent threshold was seen to remove a greater amount of the unwanted systematics. Now it is possible to compare figure 3.1 with 3.2 below after any data points greater than the 2σ boundary set by the initial GP have been removed.

Keeping the graphs in the same scale clearly shows that a large proportion of the deep transits have been removed and this has allowed the Gaussian process to fit a mean function through the star. However, these transits have not been completely removed which suggests a need for another method for removing the systematics still left in the data. This is because ensuring the deduction of 'transit-like' behaviour at this stage will be shown to be extremely significant in the analysis of the pipeline in chapter 4.

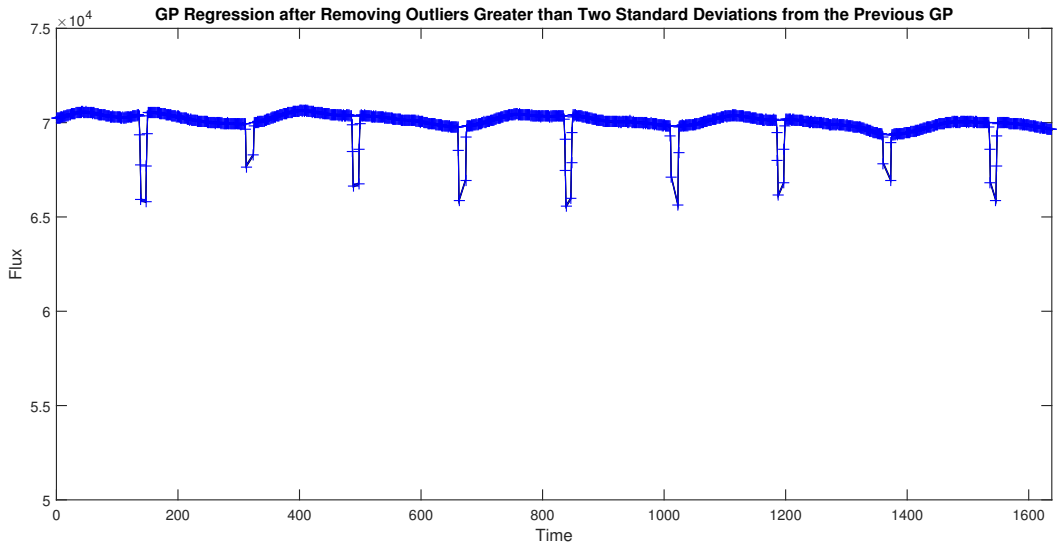


Figure 3.2: Star 1 after just removing data points greater than the two standard deviation boundary set by the previous GP.

3.2.2 Removing Data Based on the Euclidean Distance between Points

This concept has developed from the idea that outliers lie far from their surrounding data points. Therefore a method has been devised to take advantage of this by computing the Cartesian distance between data points and flagging up the distances that exceed a predetermined threshold. The algorithm must then decide which data points should be considered outliers.

The threshold selected for this 'outlier removal algorithm' comes from utilising the median absolute deviation (MAD). This is defined in equation 3.1 taken from the book 'Designing Experiments & Analyzing Data' [17], where Y corresponds to each distance and M is the median of the distances.

$$MAD = median\{|Y - M|\}. \tag{3.1}$$

The book describes MAD as a 'robust type of standard deviation' as it offers itself as an effective way of locating outliers that lie outside a scaled version of the MAD. In fact in the description of the MATLAB function it explains that for normally distributed data:

$$Standard\ Deviation,\ \sigma = \frac{MAD}{0.6745}. \tag{3.2}$$

This implies that $2\sigma \approx 3MAD$ and gave a provisional starting point for the threshold to be set. It has been concluded to choose 5 as the multiplying factor instead of the intitial choice of 3; a decision which has been consolidated through testing on many quarters.

Once a length f has been flagged as being above the threshold, the code must then select one

of the two data points, x_1 or x_2, that make up the length to be the outlier. The procedure is shown in figure 3.3. The decision is made by summing up the lengths in the local area either side of f. x_1 is

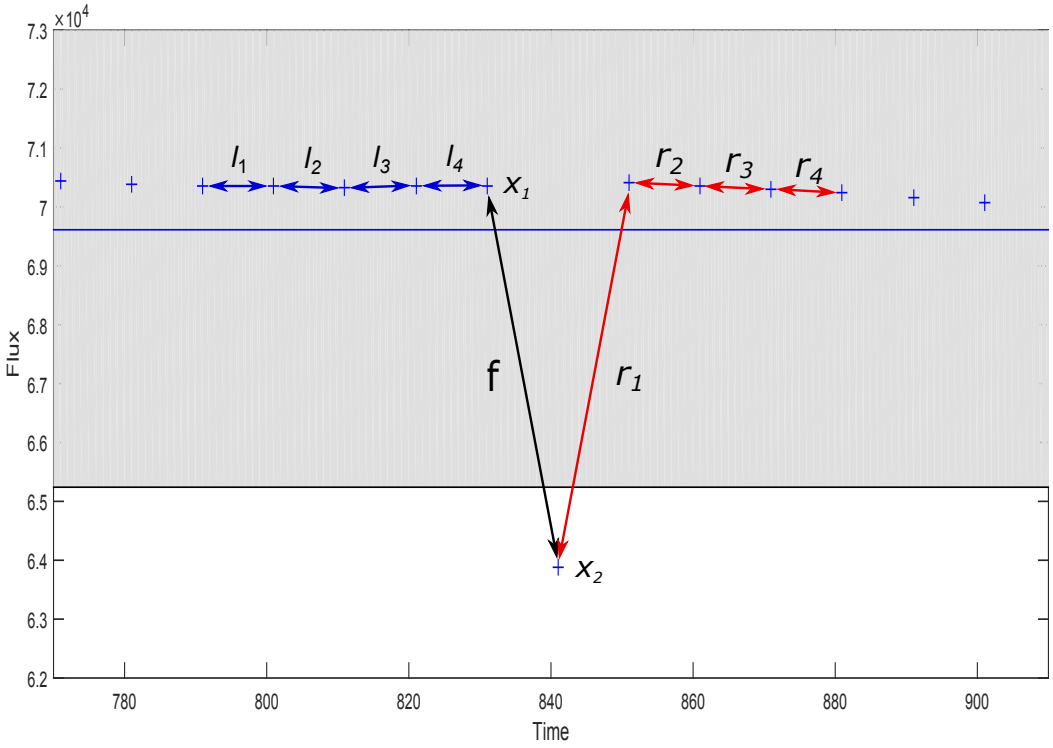


Figure 3.3: Explanation of outlier detection algorithm.

selected as the outlier if:

$$l_1 + l_2 + l_3 + l_4 > r_1 + r_2 + r_3 + r_4, \tag{3.3}$$

and x_2 is chosen for when the inequality is the reverse. Note that the code has been written so that the number of data points used for comparison can be altered. However a window the size of 4 data points either side has been provisionally chosen due to its success through testing. Figure 3.4 demonstrates the success of this method. Unlike the previous method, after just one application of the MAD outlier removal algorithm (MADORA) we have removed almost all of the deep transits associated with the eclipsing binary, with only a few remnant data points left. An important advantage of this method is that it does not require the mean function and standard deviation function to be calculated before applying it which makes it computationally less expensive to apply on its own. Therefore this could reduce the computation time required for each iteration. One potential disadvantage to this method is that it is capable of removing too many data points. For example in the instance that over half the distances between the points have the same value, the MAD would be calculated as zero which would lead to lengths which are only slightly larger being classified as outlier lengths [18].

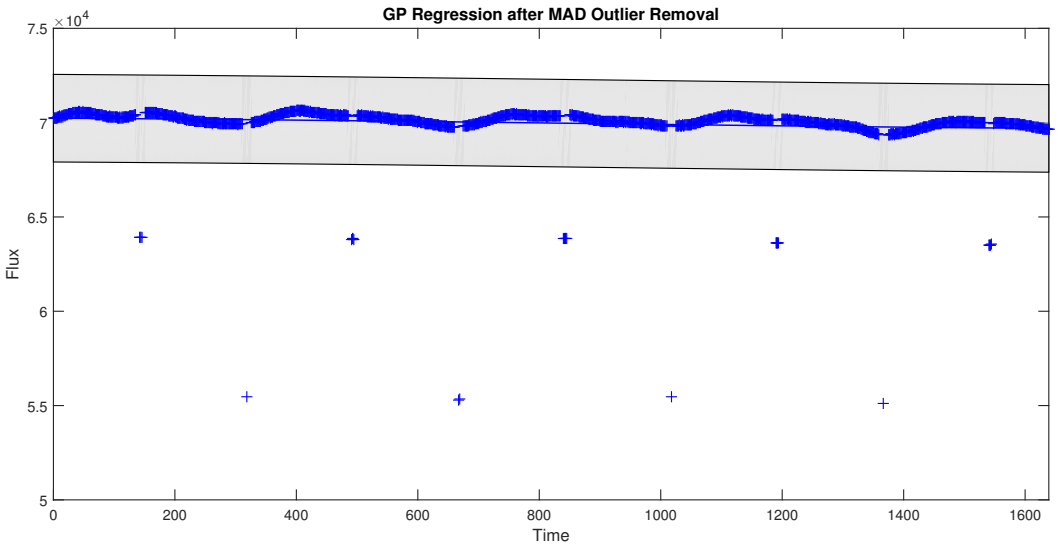


Figure 3.4: Star 1 after applying the MAD outlier removal algorithm.

3.2.3 Overview of Outlier Removal

In the example of this eclipsing binary used throughout this section the advantages and disadvantages of both outlier removal techniques have been demonstrated. Removing all the data points outside two standard deviations has managed to get rid of the majority of deep transits, but it could not eliminate points within this region that exhibit the transiting behaviour. The MADORA is shown to be much better at removing these points but can fail on points further into a transit due to the length of the window used. It should become clear from these reasons and from looking at the results in the plots that combining both methods into one powerful outlier removal algorithm will build a procedure that takes advantage of the positives of both methods. This combined process can be seen in action in figure 3.5.

3.3 Varying the Input Scale Length According to the Autocorrelation

In some cases, even after having removed the appropriate outliers, the Gaussian process can still struggle to fit to the behaviour of certain stars. This is because one of the hyperparameters being inferred, the input scale length, does not suit the data. The minimisation of equation 2.8 is not convex which means that the optimisation of the GP log-likelihood can get trapped in local minima depending on the initial hyperparameters chosen at the start of the optimisation. Therefore in order to encourage the GP's input scale length to reach a more appropriate value the autocorrelation function of the residuals has been used to alter the input scale length, thus enabling the GP to adapt to higher or lower frequencies. In this report, the residuals refer to the data points minus the mean function

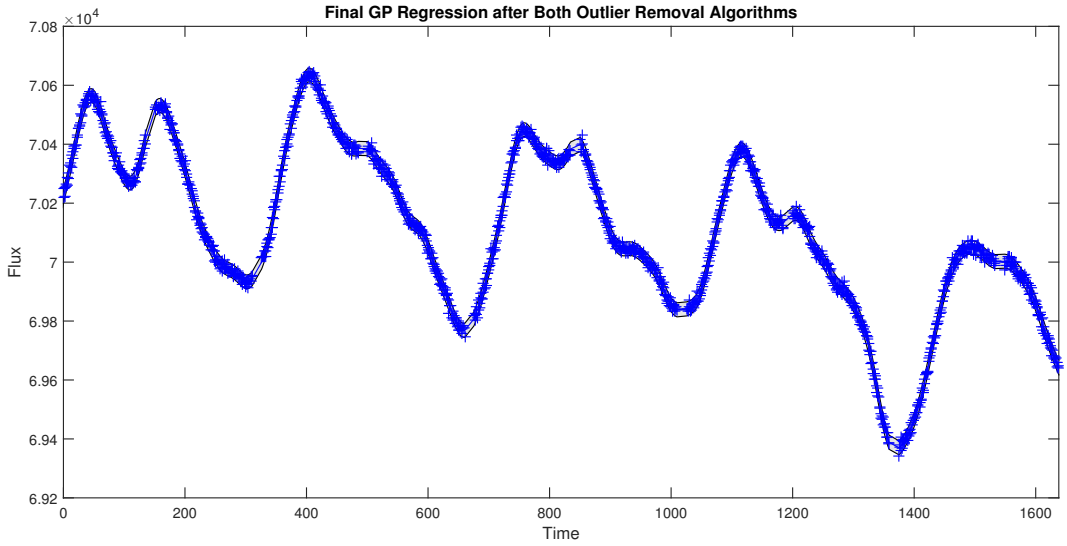


Figure 3.5: Star 1: The final approach after applying both algorithms. This indicates the value of combining these methods together when comparing to previous figures.

inferred by the GP.

To provide clarity an example of using the autocorrelation function to change the input length scale of the GP is included here. Initialy, applying a GP to this new example quarter gives the result shown in figure 3.6. The initial input scale length of 1 which after optimisation has been set to 2.1071 has clearly not been successful as the mean function cannot follow the data. This is because the input scale length reflects a measure of how fast the data is varying in the time direction and a final value of 2.1071 has restricted the flexibility of the mean function to fit to the data.

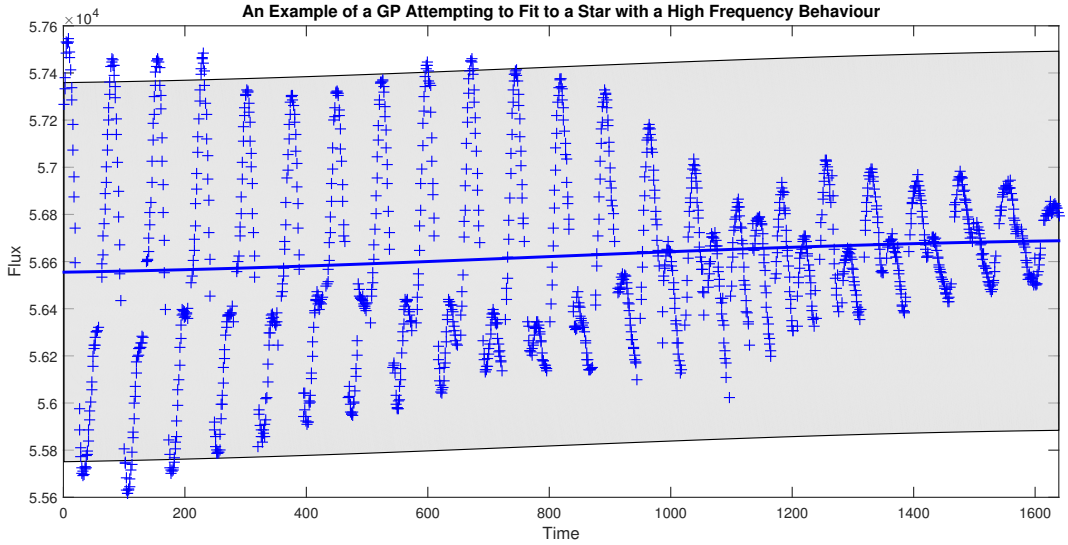


Figure 3.6: Star 2: Shows an unsuccessful fit to the data due to a final optimised input scale length of 2.1071.

The autocorrelation is used here as an approach to provide the Gaussian process with a much

better prior of the starting value for the input scale length. Figure 3.7, which is the autocorrelation of star 2, displays a clear repeating pattern which suggest that a lower value corresponding to a higher frequency is more suitable. The value chosen of 0.04 comes from the following procedure:

1. Pick out the peaks of the autocorrelation and calculate a mean lag between the peaks. In this case the mean lag between the peaks is approximately 74 units of lag.
2. Divide this value by the overall length of the data. Therefore in this example: $\frac{74}{1638} = 0.045$.
3. Reapply the code with the starting input scale length now set to 0.045.

When this sequence was automated for star 2 the initial input scale length was chosen to be 0.04498 which was then optimised to 0.01615. The result can be seen in figure 3.8 and shows a mean function that is now able to fit to the behaviour of star 2 after its outliers have been removed. In the plots of the autocorrelations, the solid horizontal blue lines correspond to 95% confidence intervals that will be discussed in the next chapter.

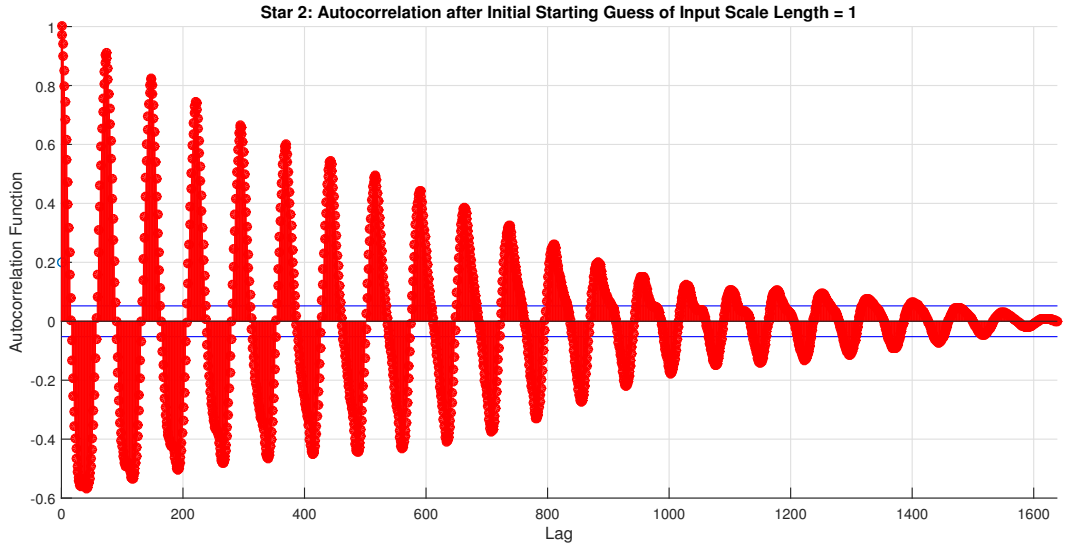


Figure 3.7: Autocorrelation function of star 2 with an initial input scale length of 1.

3.4 Summary

In summary, the procedure developed takes a quarter of PDC flux and initially applies a Gaussian process with the hyperparameters initially set to 1, 1 and 0.1 for the input scale length, output scale length and a white noise term respectively. Then unwanted outliers are removed from the data and another GP with the same starting hyperparameters is reapplied. Then the peaks of the autocorrelation of the residuals of the second GP are studied and the input scale length is changed accordingly.

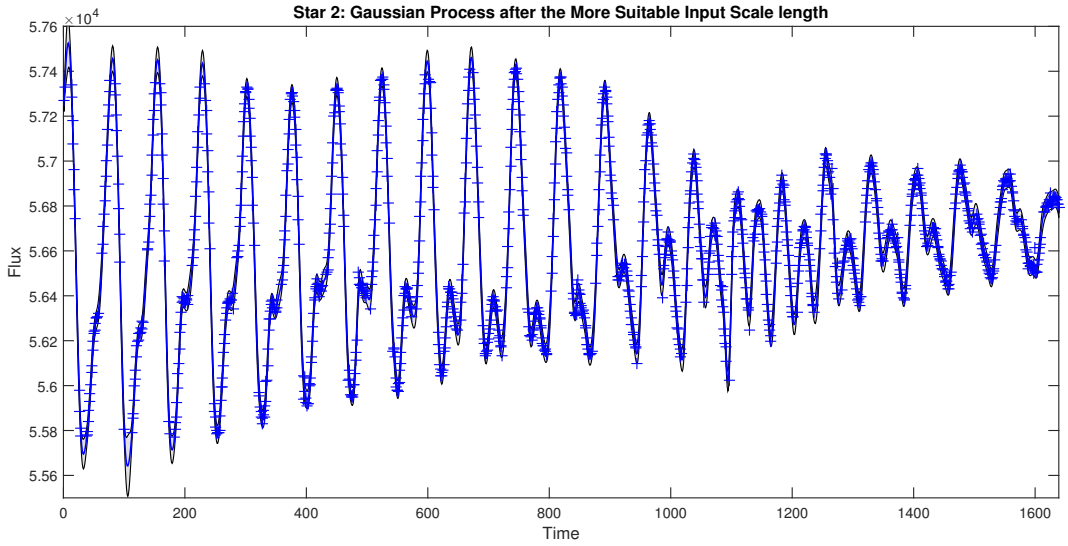


Figure 3.8: Star 2 after outlier removal and an initial input scale length of 0.04498.

The objective in this chapter has been to fit a Gaussian process to the underlying behaviour of a star. Now that all the tools have been developed to achieve this, the next step is to integrate it into a pipeline that is able to deal with the full set of 18 quarters. This comes with a new set of challenges and goals that will be dealt with in the next chapter.

Chapter 4

Gaussian Process Regression over the Whole Data Set

4.1 Introduction

In this chapter, the methods developed and detailed in the previous chapter are used to develop an automated pipeline that can process a full set of quarters from the Kepler Mission data. A large proportion of this chapter is focused on ensuring that the residuals left over after a GP has been fitted consist of white noise. This necessary condition is a requirement of the Box Least Squares method for detecting transits in data as it operates most effectively when the residuals resemble Gaussian noise (see chapter 5). Note that the residuals of the flux are calculated according to equation 4.1:

$$\mathbf{Residuals} = \mathbf{x} - \boldsymbol{\mu}(\mathbf{x}). \tag{4.1}$$

Where $\mathbf{x}$ is the vector of flux and $\boldsymbol{\mu}(\mathbf{x})$ is the corresponding inferred mean function of the GP.

Despite analysing the complete set of quarters for each star, each quarter is processed independently and then the quarters are all subsequently joined together. This has both computational and practical benefits associated with the data which will be discussed. Then any removed data points can be reinserted allowing any patterns associated with planets to be looked for thereafter.

4.2 Automating the Pipeline

The separate blocks of code that were written for the previous section must now be organised so that the process of fitting a GP to the data and removing outliers is automatic. This requires a pipeline which will run until certain conditions are met. This particular element of the MATLAB script was

named the 'gpml_pipeline' and runs through the following steps:

```
% Apply initial Gaussian process.
% Remove outliers greater than 2 standard deviations.
% MAD outlier removal.
% Calculate GP.
% Check if GP has been successful.
```

The first four steps have been detailed in chapter 3. Therefore the key part of this section stems from what is required from stage 5 of the pipeline.

4.2.1 Validating the Success of the Gaussian Process

There is always the possibility that upon reaching the fifth step in the pipeline that the Gaussian process does not model the underlying behaviour of the star. This is exemplified when looking at figures 3.6 and 3.7 from section 3.3. In this example it was checked by eye that the GP did not follow the data and the process was applied once more with a different input scale length. In order to validate the fit automatically, there must be a condition that has to be met.

It is expected that removing the mean function from the outlier-removed flux will leave residuals that just contain white noise. In order to confirm whether the residuals consist of Gaussian noise, the autocorrelation function is used. The calculated residuals are only deemed to be white noise if the autocorrelation resides within the confidence region. The confidence interval calculated is one for an assumed autocorrelation of white noise. That is, for a 95% confidence interval, the boundaries are:

$$0 \pm \frac{1.96}{\sqrt{N}}, \tag{4.2}$$

where N is the number of samples [19]. For the example of star 2, figure 4.1 displays the autocorrelation of the residuals of figure 3.8 and indicates independent Gaussian noise with the autocorrelation meeting the condition that 95% of the values reside between the two solid blue confidence boundaries, thus validating the success of the GP. This can directly be compared to the initial autocorrelation, figure 3.7, which indicates a clear structure.

If the residuals are confirmed to be white noise, then the pipeline has been successful and the algorithm can proceed to the next stage. However, if the test has resulted in rejecting the hypothesis that the residuals are white noise, the algorithm continues into a further stage of the pipeline.

Two possible reasons for the routine's failure in modelling the underlying behaviour of the star are

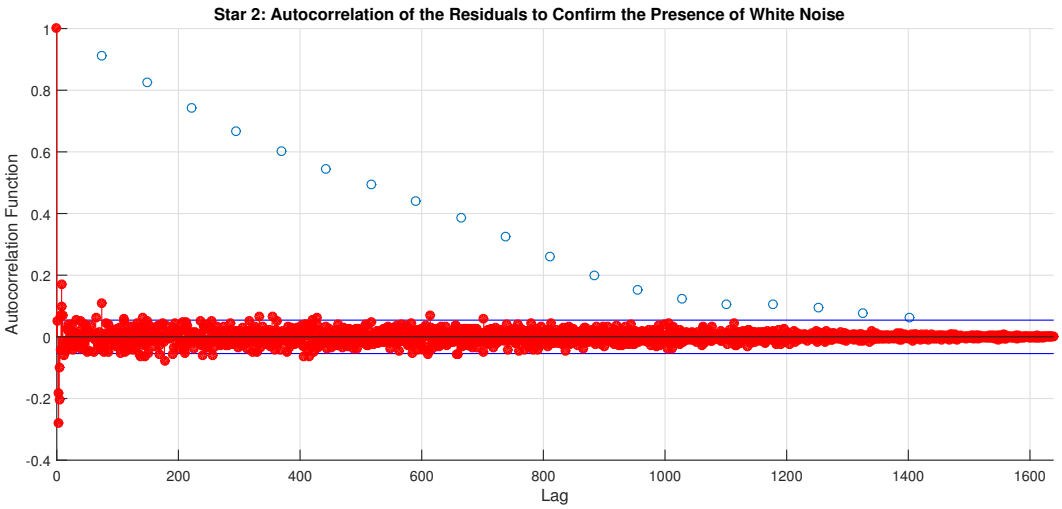


Figure 4.1: The residuals of star 2 confirm the presence of white noise. The circles show the positions of the peaks in the autocorrelation before the second iteration of the pipeline.

as follows:

1. The outlier removal algorithms have not managed to remove all the appropriate points, thus preventing the Gaussian process to fit the data.

2. The initial hyperparameters have hindered the Gaussian process's ability to fit to the data.

Both of these problems are resolved by iterating through the pipeline again, until the autocorrelation meets the noise floor criterion. We note that this also allows for the option of resetting the input scale length.

Determining what the new input scale length has already been touched on in section 3.3. It was concluded that the autocorrelation function (ACF) should be used as part of the procedure for setting the appropriate input scale length for the GP. An initial choice of 1 is sometimes too large to capture the faster moving components of the signal. Therefore the following pseudo code demonstrates the process of modifying this value for further iterations of the Gaussian process.

```
1  if 95% confidence bound condition on autocorrelation is met then
2      Continue to the next stage of the pipeline;
3  else
4      if The input scale length == 1 then
           /* Find the peaks of the autocorrelation and their corresponding
              locations                                                    */
5          [Peaks,Locations] = findpeaks(Autocorrelation);
           /* Calculate the mean difference between the locations of the
              local maxima in the autocorrelation to find the 'period'     */
6          period = mean(diff(Locations));
7          if Number of Peaks > 3 then
               /* Input scale length 'L' is equal to the period divided by
                  the dimension of the flux vector                         */
8              L = period/length(flux);
9          else
               /* Set the input scale length to half the initial length    */
10             L = 0.5;
11         end
12     end
13     Iterate through the outlier removal stage of the pipeline again;
14 end
```

Algorithm 1: Procedure for Modifying Input Scale Length

Algorithm 1 is called upon to validate the success of the Gaussian process. One thing to note about the code is that it can only continue on to the next stage of the pipeline upon meeting the 95% confidence bound condition. It is also important to mention that the initial input scale length is only changed from the original value of 1 on the first occasion of passing through this section of the code. Thereafter, each iteration simply identifies any further outliers and recalculates the GP. Line 7 of algorithm 1 ensures that the initial input scale length is only calculated from the autocorrelation if it has sufficient structure. This is the case if the autocorrelation has more than three peaks. Otherwise line 10 sets the initial input scale length as 0.5 to try and capture components of the data which span half the quarter. This is in comparison to the initial value of 1 that attempted to fit its systematics over the entire length of the quarter.

4.3 Pipeline Termination Conditions

The pipeline has been automated to run until any of the following conditions are met:

1. The ACF of the data implies white noise, as determined from the 95% confidence bounds.
2. No further data points are removed after an iteration.
3. Ten iterations have taken place.

Ten has been chosen as a reasonable compromise as this was empirically observed to be adequate.

Finally the flow diagram succeeding this paragraph demonstrates the process that the pipeline follows during the running of the program. After the data is split into its appropriate quarters, each quarter is then shown to go through the flow diagram below. The outcome is the finally processed quarter which will then be stored and recombined, as described in the next section. Splitting the

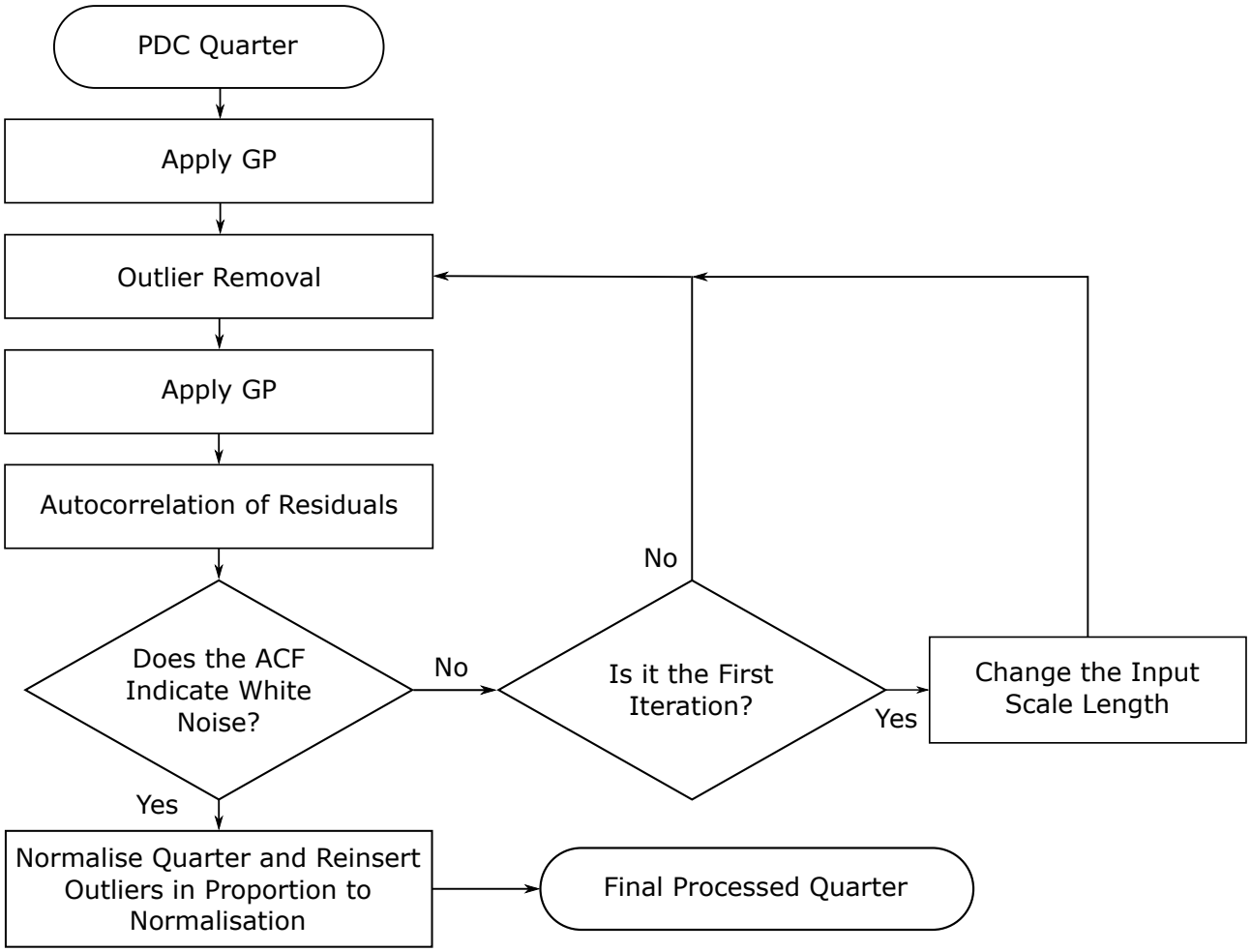


Figure 4.2: Flow diagram showing how the pipeline processes each quarter.

Kepler data into quarters has made the pipeline much faster than if it were to be applied to the whole dataset in one go due to the expensive nature of inverting large covariance matrices. It is

also the logical way that the data presents itself as it naturally divides up this way. This enables the parallelization of the process when multiple cores are available.

4.4 Matching the Quarters

Before splitting a star into its quarters the PDC flux for each star is passed through the function '`madNorm`'. This light pre-processing of the data removes the median from the flux and scales the flux according to the median absolute deviation. This is demonstrated in equation 4.3 where the MAD is the median absolute deviation of the '$Input\ Flux$'. The reason for using the median and MAD is because they are both robust measures and are unaffected by outlying points.

$$Output\ Flux = \frac{Input\ Flux - median\{Input\ Flux\}}{MAD}. \tag{4.3}$$

After this simple rescaling the data is split into its quarters and each one is passed into the pipeline represented by the flow diagram in figure 4.2.

After each quarter is passed through the '`gpml_pipeline`', the resulting residuals of each quarter are operated on by the function '`madNorm`' once again. The importance of applying this normalising function to each quarter is highlighted when comparing figure 4.3 with 4.4. Figure 4.3 shows the data before being processed by the necessary normalising function and this is directly compared with figure 4.4, which displays how the quarters have been matched to have similar variance and the same median. This now leaves the outlier removed residuals to closely resemble white noise. The reason for having to scale each quarter separately is due to the clear discrepancy in the variances between quarters. This is likely to be due to the rotation of the Kepler telescope every quarter which leads to the movement of the stars to 'new positions in the focal plane'[20].

4.5 Reinserting Outliers

The final phase of the process, before being able to start detecting possible planets is to reinsert the outliers which were removed during the pipeline. This is achieved by storing the median m_Q and the MAD_Q values of each quarter. Then equation 4.4 is applied to the original PDC flux, $\mathbf{S}_Q$ for each quarter. This utilises the mean function $\boldsymbol{\mu}_Q$ calculated by the GP for each quarter. Note that the subscript Q denotes a particular quarter.

$$Final\ Preprocessed\ Quarterly\ Flux,\ \mathbf{F}_Q = \frac{\mathbf{S}_Q - \boldsymbol{\mu}_Q - m_Q}{MAD_Q} \tag{4.4}$$

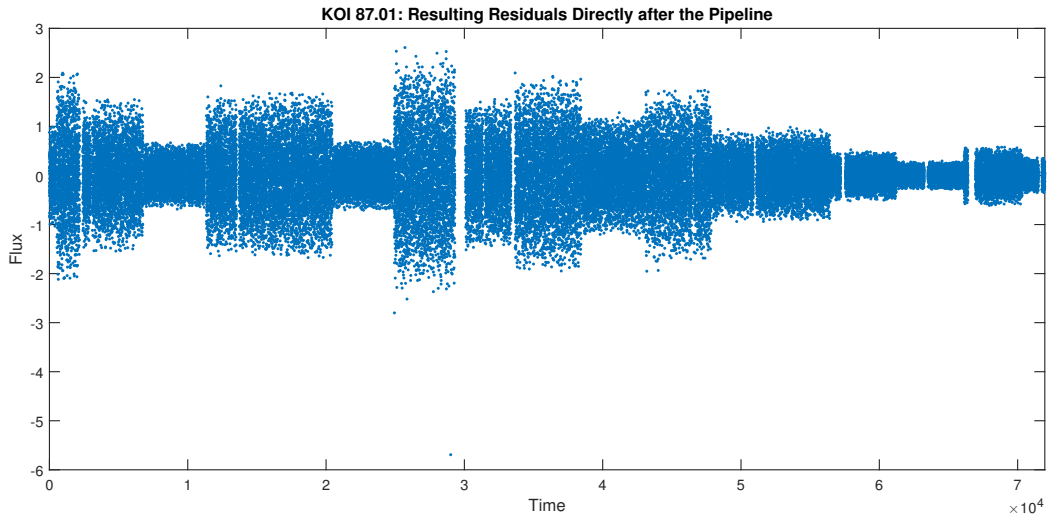


Figure 4.3: Residuals of KOI 87 before being normalised.

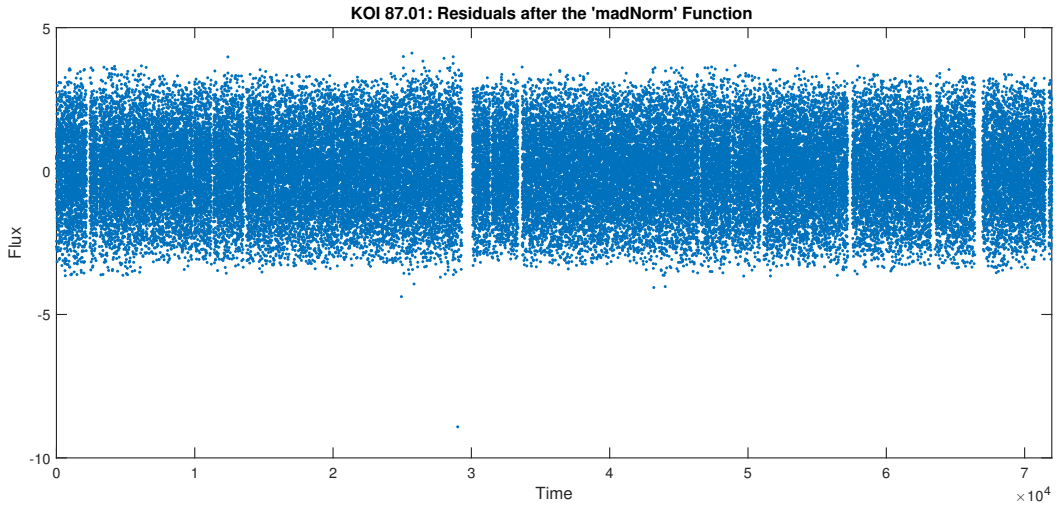
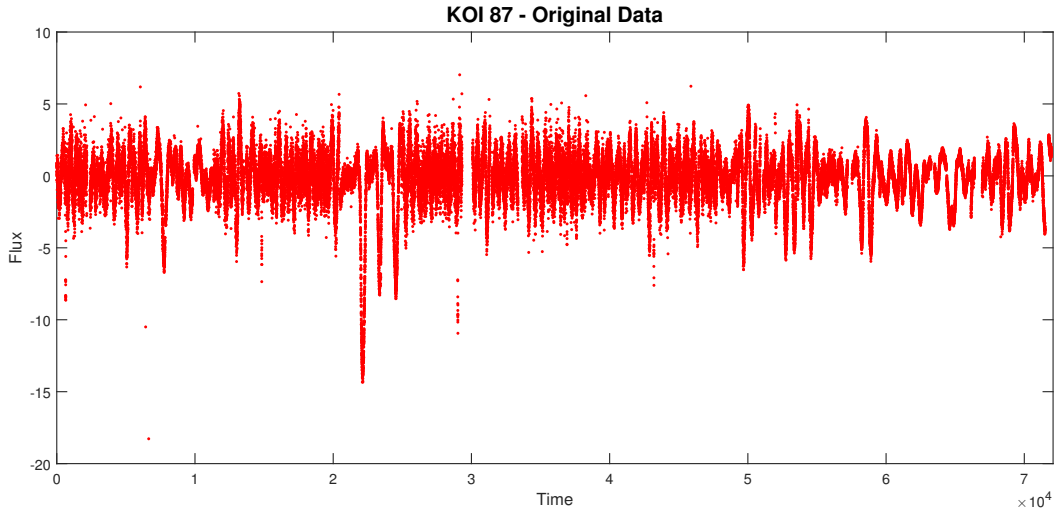


Figure 4.4: Residuals of KOI 87 after being normalised.

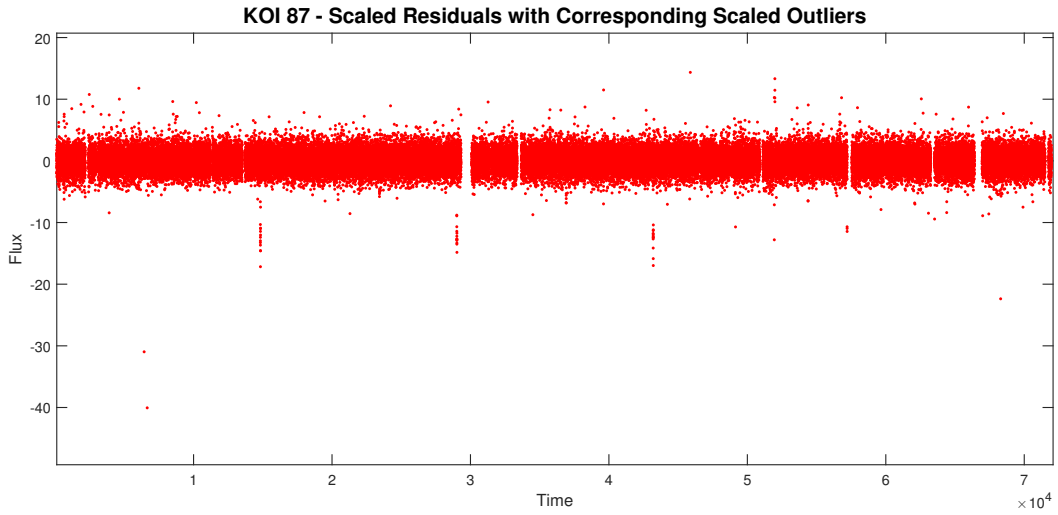
This results in a plot being produced such as figure 4.5b, which is the final output from the pipeline. This flux is now a combination of Gaussian noise and signals corresponding to potential exoplanets.

4.6 Summary

The pipeline for preprocessing the data has now been developed. The right tools are now in place to condition the data into the form of figure 4.5b, when given the entire dataset of a star. Data in this form is now ready to be analysed for exoplanets using planet detection algorithms. This is explored in the next chapter.



(a) Original data for KOI 87 before preprocessing in the pipeline.



(b) The result for KOI 87 after preprocessing in the pipeline and reinserting the scaled outliers.

Figure 4.5: An example of how the '`gpml_pipeline`' has processed the original PDC flux into Gaussian noise and potential planetary signals. This data is now suitable for planet detection algorithms.

Chapter 5

Planet Detection

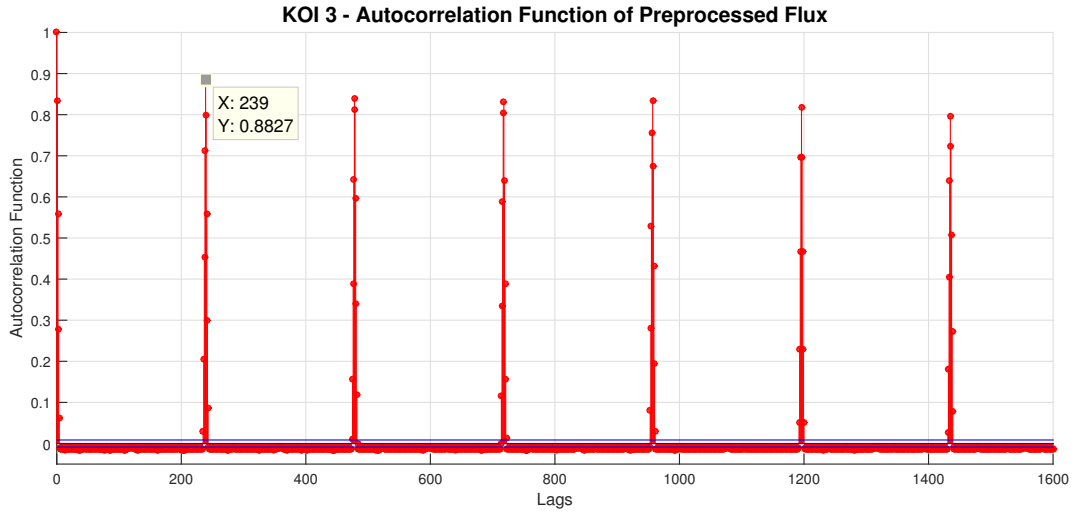
5.1 Introduction

The aim of this chapter is to present possible methods of detecting planets given a selection of data which is known to contain exoplanets, emphasising on the use of the autocorrelation as an important aid in identifying transits in the signal.

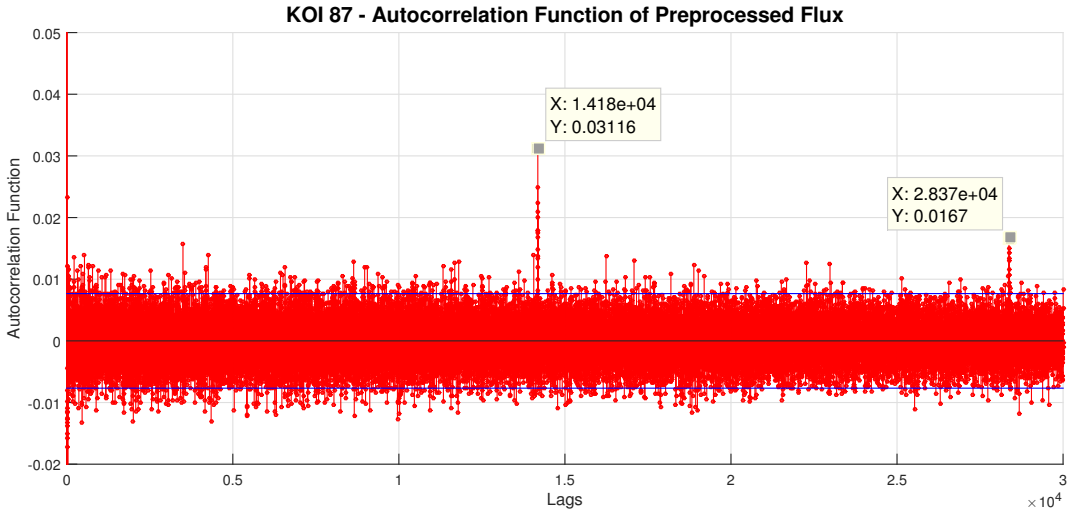
The current technique used in NASA's pipeline for detecting transits is the Box-Least Squares method [21]. This method will be applied to stars which have been through the automated process described in chapter 5. A comparison will then be made to demonstrate the success of preprocessing the data in this manner. Finally the idea of incorporating the autocorrelation with the BLS algorithm will be put forward and shown to have promising results.

5.2 Autocorrelation Function for Initial Planet Detection

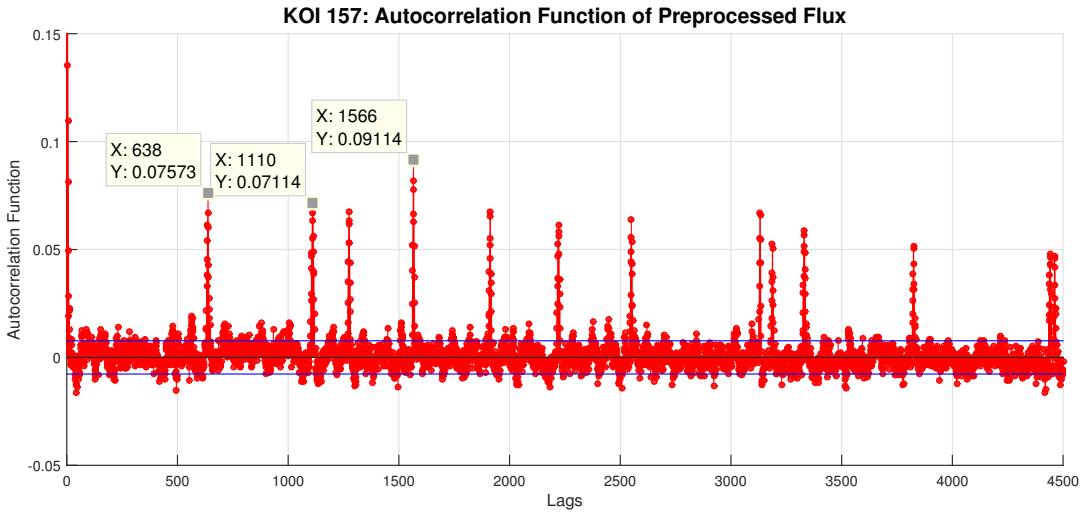
The inherent periodic nature of a signal containing a transiting planet has led to the idea of using the autocorrelation for initial planet detection. Our working hypothesis is that taking the autocorrelation of the residuals with their outliers reinserted should reveal if there are any possible planets in the data. Therefore any repeating patterns such as transits should be picked up by the autocorrelation function as peaks. Figure 5.3 demonstrates this method and the results are tabulated in table 5.1. Note that the identified peaks in the autocorrelation plots have been manually labelled and that KOI refers to Kepler Object of Interest.



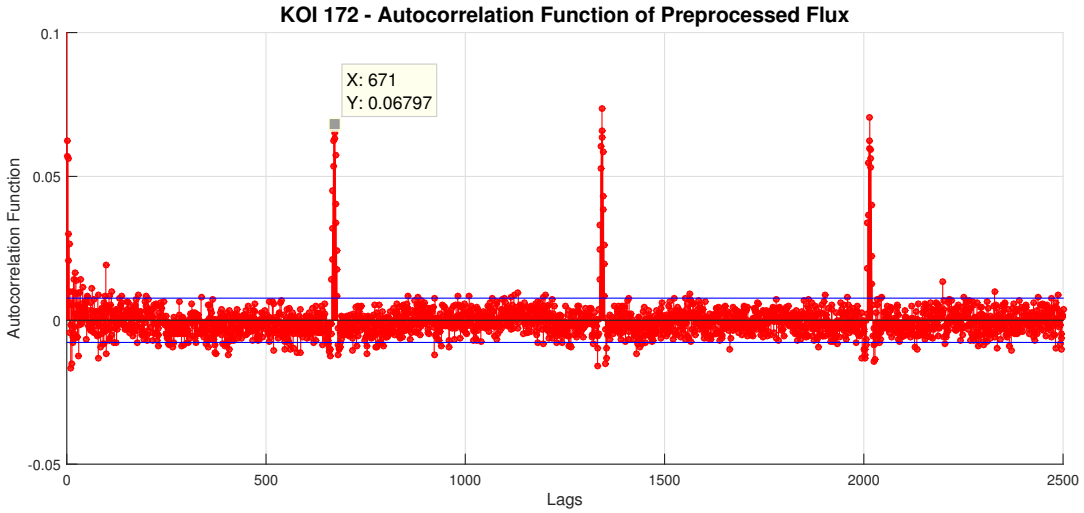
(a) Labelled peak corresponds to the planet KOI 3.01.



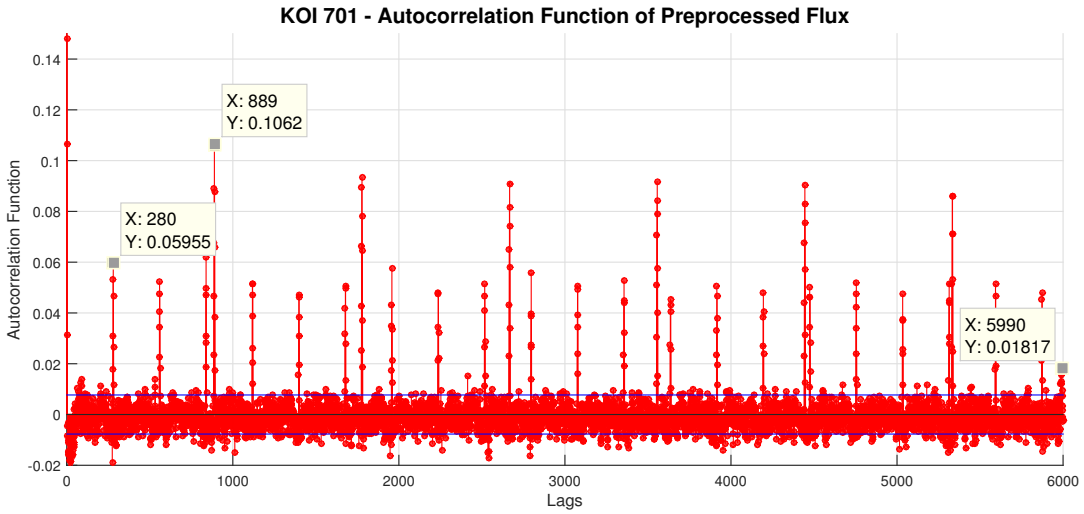
(b) Labelled peaks correspond to the first and second harmonic of the planet KOI 87.01.



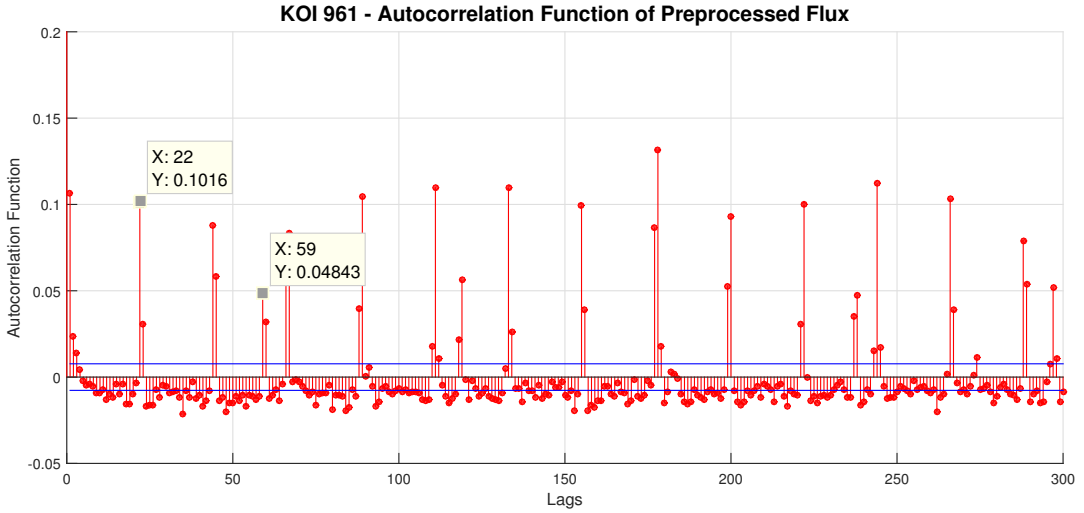
(c) Labelled peaks correspond to the planets KOI 157.01, KOI 157.02 and KOI 157.03.



(d) Labelled peak corresponds to the planet KOI 172.01.



(e) Labelled peaks correspond to the planets KOI 701.01, KOI 701.02 and KOI 701.03.



(f) Labelled peaks correspond to the planets KOI 961.01 and KOI 157.02.

Figure 5.1: The results of taking the autocorrelation of the preprocessed residuals of six Kepler datasets that have passed through the '`gpml_pipeline`'. The peaks of the autocorrelations are the harmonics of the transiting planets.

5.2.1 Table of Results

The table included in this section contains the planets which have been identified when analysing the peaks in the autocorrelation function. The success of this initial detection is quantified in the 'Percentage Difference' column of the table. This is calculated from the difference between the planet periods from the NASA Exoplanet Archive [2] and the predicted periods from the autocorrelations. These percentages are small, validating the use of the autocorrelation for planet detection.

The column labelled 'Peak' corresponds to the lag in the ACF. For example a peak at a lag of 59 corresponds to 59 multiplied by the time between measurements, which has already been mentioned as 29.4 minutes. This gives a period of 1.20 days to three significant figures as demonstrated for planet 961.01 in the table.

KOI	Planet	Peak	Predicted Period (days)	NASA Period (days)	Percentage Difference
3	3.01	239	4.879583333	4.8878162	0.168%
87	87.01	14184	289.59	289.8623	0.094%
157	157.01	638	13.02583333	13.0241	0.013%
	157.02	1110	22.6625	22.6845	0.097%
	157.03	1566	31.9725	31.9996	0.085%
	157.04			46.6888	
	157.05			118.3807	
	157.06			10.3039	
172	172.01	671	13.69958333	13.722341	0.166%
	172.02			242.4613	
701	701.01	889	18.15041667	18.16406	0.075%
	701.02	280	5.716666667	5.714932	0.030%
	701.03	5990	122.2958333	122.3874	0.075%
	701.04			267.291	
	701.05			12.4417	
961	961.01	59	1.204583333	1.2137672	0.757%
	961.02	22	0.449166667	0.45328509	0.909%
	961.03			1.865169	

Table 5.1: Planets confirmed via looking for peaks in the autocorrelation of known planetary data [2].

Note that in some instances it is difficult to manually identify planets from the ACF. This is why the table contains some empty cells. One possible cause of this could be due to some harmonics of planets coinciding with other exoplanets within the same system. As an example, KOI 961.03 could not be seen because it overlaps with the fourth harmonic of KOI 961.01. This introduces the challenges associated with locating planets in highly populated systems.

5.3 Box-Least Squares (BLS)

The application of the BLS algorithm explains the necessary requirement of conditioning the flux to comprise of Gaussian noise and transit signals. BLS is essentially a form of least squares fitting which assumes that 'all data points have the same noise level' as quoted from Kovács' original paper on the BLS algorithm [21]. This confirms the requirement of reducing the data to white noise and the transit signals.

The example in figure 5.2 shows how a box shape has been least squares fitted to data that resembles a planetary signal consisting of white noise and a planet. This is similar to the data produced at the end of chapter 4. The BLS method operates using the following algorithmic process:

1. Select a 'test period' from the set of evenly spaced test periods that span the data.

2. Partition off the flux into period long sections. For example if the flux contains 20,000 data points and the period chosen was 1,000, then this step would lead to dividing up the data into 20 sections, each of 1,000 in length.

3. Fold the flux. This means putting each section on top of one another and summing up the values at the same corresponding time steps.

4. This step involves varying the width and position of the box function within the folded flux to find the best fit. Formally, this is the minimisation of D which is a function of the indices i_1 and i_2. Note that these indices correspond the the start and end of a transit signal and they determine the width and position of the box. Equations 5.1, 5.2 and 5.3 are taken from Kovács' paper [21].

$$D = \sum_{i=1}^{i_1-1} \bar{w}_i(\bar{x}_i - \hat{H})^2 + \sum_{i=i_2+1}^{n} \bar{w}_i(\bar{x}_i - \hat{H})^2 + \sum_{i=i_1}^{i_2} \bar{w}_i(\bar{x}_i - \hat{L})^2. \tag{5.1}$$

This equation can also be rewritten as:

$$D = \sum_{i=1}^{n} \bar{w}_i \bar{x}_i^2 - \frac{s^2}{r(1-r)}. \tag{5.2}$$

Where $s = \sum_{i=i_1}^{i_2} \bar{w}_i \bar{x}_i$, $\hat{L} = \frac{s}{r}$ and $\hat{H} = \frac{s}{1-r}$. The weightings $\bar{w}_i$ are related to the variance of the data and since effort has been made to ensure that the noise of the flux is now white, the weightings can be assumed constant and have no impact on the minimisation. $\hat{H}$ and $\hat{L}$ are the calculated upper and lower levels that produce the box signal as demonstrated by the solid red line in figure 5.2. As the signal has had its mean removed, $\hat{H}$ is assumed to be zero in the MATLAB BLS program written for this report.

Since only the second term on the right of equation 5.2 is dependent on the indices to be minimised over, the problem can be restructured to maximising this term. This is defined as the 'Signal Residue', (SR), by Kovács' [21] and has been included here:

$$SR = MAX\left\{\left[\frac{s^2(i_1, i_2)}{r(i_1, i_2)(1 - r(i_1, i_2)}\right]^{\frac{1}{2}}\right\}. \tag{5.3}$$

5. These steps are repeated for the entire set of periods to find the maximum SR value. This will correspond to finding a period that best describes transits in the data and the associated width, depth and the position of the first epoch in the flux.

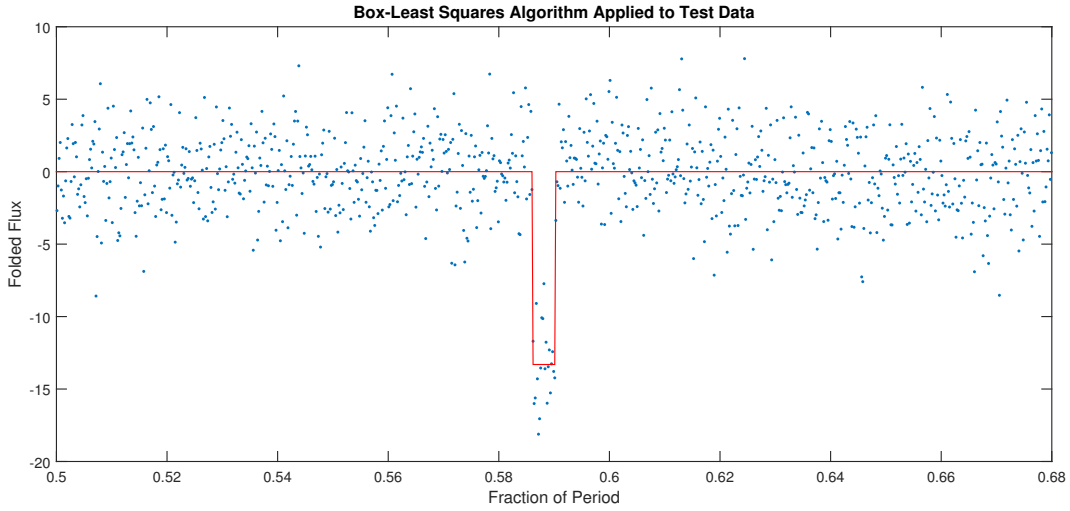
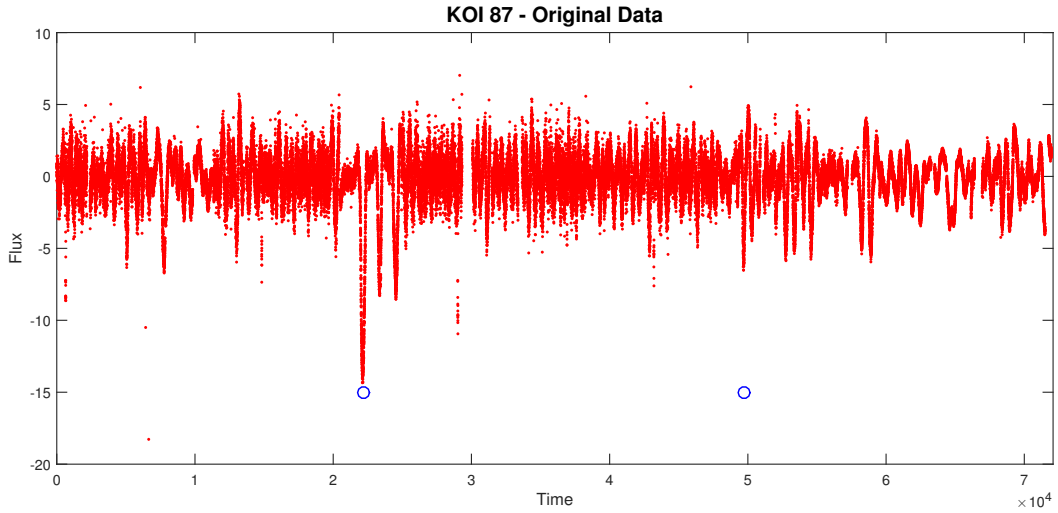


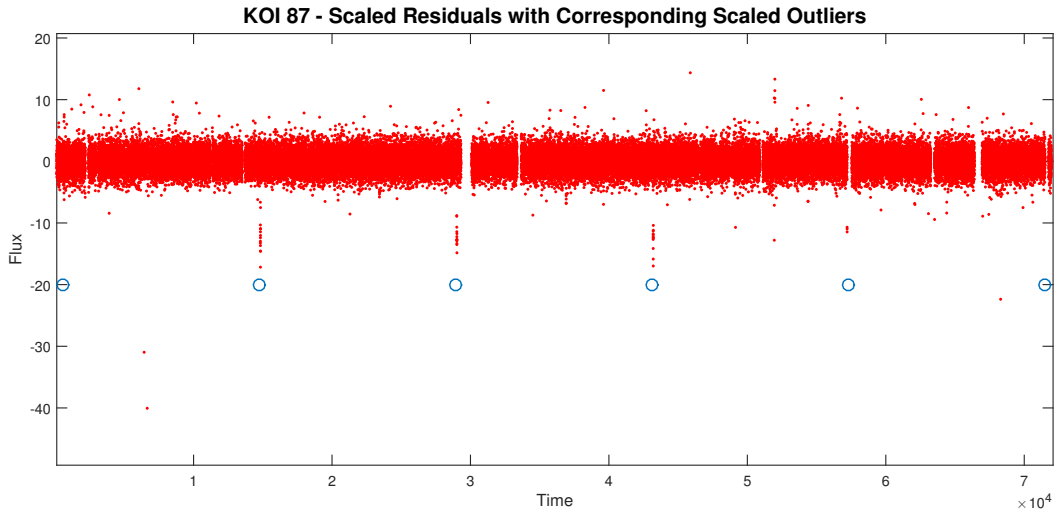
Figure 5.2: An example of a Box-Least Squares fit to test data consisting of zero mean Gaussian noise and a transit signal.

5.3.1 Current Application

The current algorithm used as part of NASA's pipeline transit detection has been made available by Dr. Suzanne Aigrain. This code has been applied to planetary data both before and after passing it through the '`gpml_pipeline`'. The results of which have been inserted onto figures 5.3a and 5.3b as blue circles. These are the positions of the predicted transits after running this BLS code on the data. The depths of these circles have merely been chosen for clarity. It is clear from looking at figure 5.3b and by verifying with the the NASA Exoplanet Archive [2] that preprocessing the data has enabled the BLS code to correctly locate the exoplanet, KOI 87.01, when fed the whole 18 quarter dataset. This successful detection was not possible without the '`gpml_pipeline`'. This is demonstrated in figure 5.3a as shown by the incorrectly estimated transits.



(a) Original data for KOI 87 before preprocessing in the pipeline. The blue circles correspond to the predicted transits according to the BLS algorithm.



(b) The result for KOI 87 after preprocessing in the pipeline and reinserting the scaled outliers.The blue circles correspond to the predicted transits according to the BLS algorithm.

Figure 5.3: Figure 5.3b shows the detection of the planet KOI 87.01 via the BLS method being applied to the preprocessed data. This is compared to figure 5.3a which shows the falsely predicted transit locations due to the systematics in the original data.

5.3.2 Incorporating the Autocorrelation Function

It should be evident from the description above that the BLS requires a broad parameter space, for example the periods tested could span from one day to as much as one hundred days. The limitation of this crude grid search is the potential to miss transits. The autocorrelation function provides a successful alternative technique for the detection of extrasolar planets. Therefore an improvement on the current BLS method could be made through using the ACF of the flux to hint at periods that we wish the BLS algorithm to investigate.

Incorporating the ACF with the BLS algorithm requires using the peaks of the autocorrelation as the input periods for the BLS method which will be referred to as the BLS with autocorrelation inputs

(BLSAI). To give an indication as to how effective the autocorrelation will be for selecting suitable prior periods, figure 5.4 shows how the Signal-to-Noise Ratio (SNR) behaves according to test data. This data consists of Gaussian noise with a variance of one and a synthetic transit signal with an input period and input depth which have been varied. The colour scheme corresponds to the SNR which is defined here as:

$$SNR = \frac{ACF_p}{\frac{1.96}{\sqrt{N}}}. \tag{5.4}$$

This is the value of the autocorrelation at the known input period ACF_p divided by the 95% confidence interval.

In the yellow region where the SNR is greater than 2, it is expected that the BLS algorithm will have no trouble correctly identifying transits from the peaks of the ACF. The relationship confirms the belief that exoplanets with shorter periods and a large transit depth are easier to find.

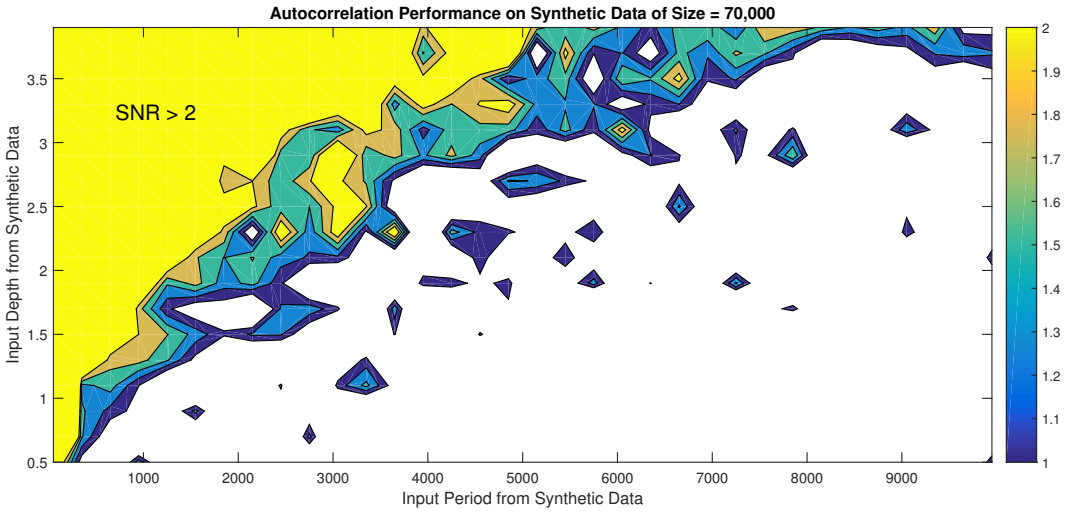


Figure 5.4: A contour plot to show how the SNR is related to the depth and period of synthetic exoplanets. The yellow region corresponds to a SNR which is greater than 2. It is expected that any planet with a period and depth falling in this region would be detectable. The SNR decreases according to the colour scheme located to the right of the plot and the white region is where the SNR is less than 1. Any planets in this white region would not be easily detectable.

Testing the BLSAI Algorithm

Applying the BLSAI to the flux in figure 5.3a and 5.3b gives figures 5.5 and 5.6 respectively. Figure 5.5 once again demonstrates how the original data is not suitable for any form of BLS to be applied and also shows that when BLSAI fails it tends to predict a period near the cut off frequency. Figure 5.6 correctly identifies the extrasolar planet KOI 87.01 at a peak of 14184 and also picks out a significant period at 3545 Kepler time units[1].

[1]This peak may correspond to a new exoplanet in the KOI 87 system.

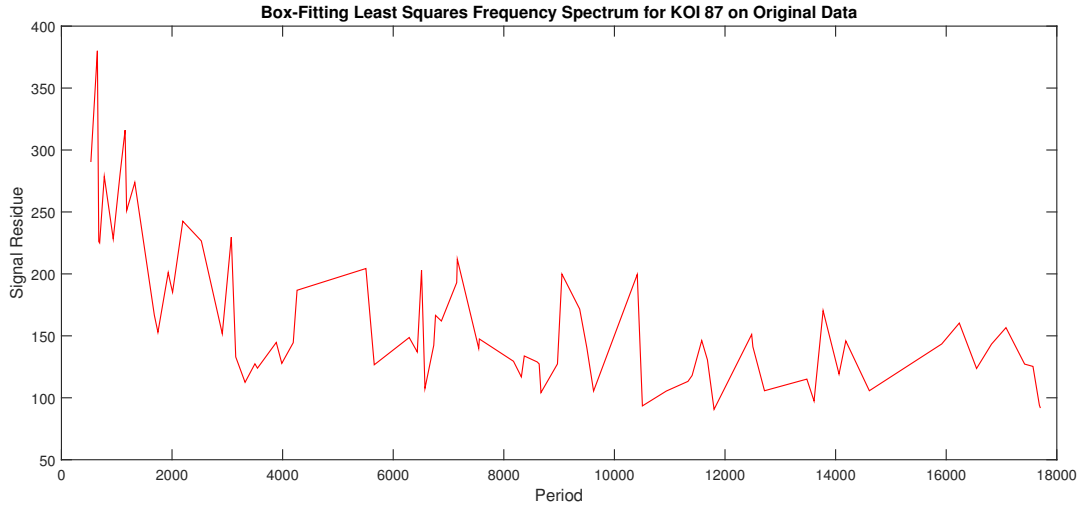


Figure 5.5: Signal Residue of KOI 87 before preprocessing using BLSAI.

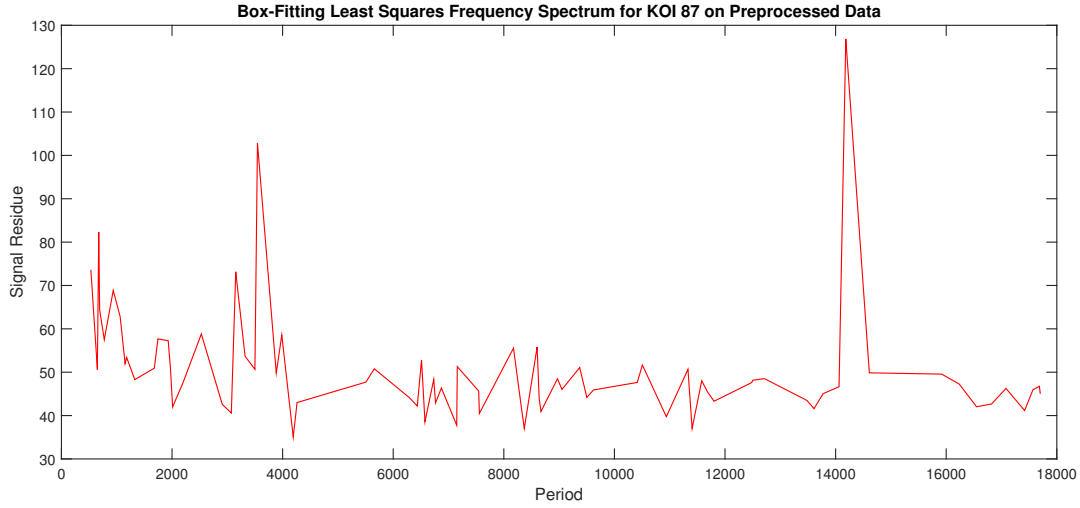


Figure 5.6: Signal Residue of KOI 87 after preprocessing using BLSAI.

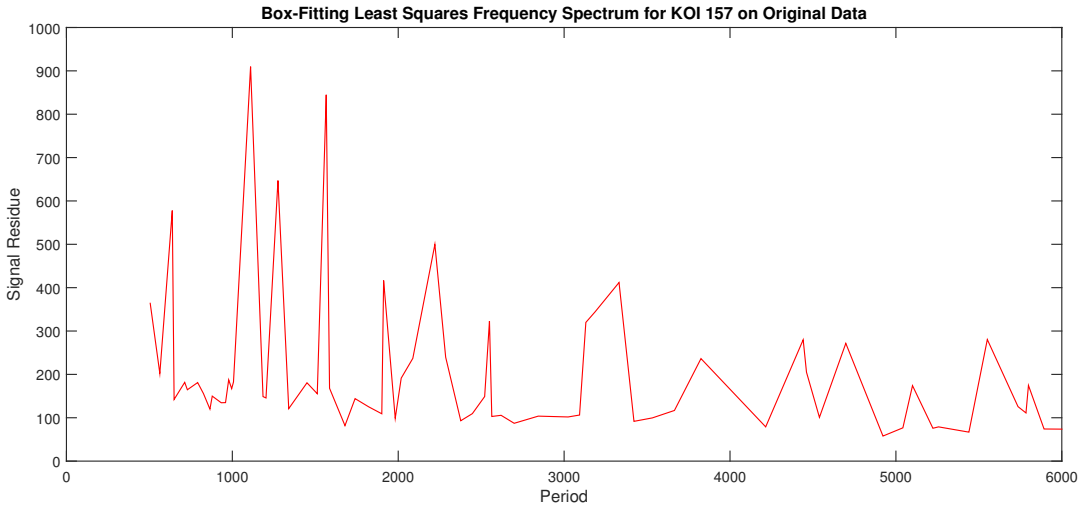


Figure 5.7: Signal Residue of KOI 157 before preprocessing using BLSAI.

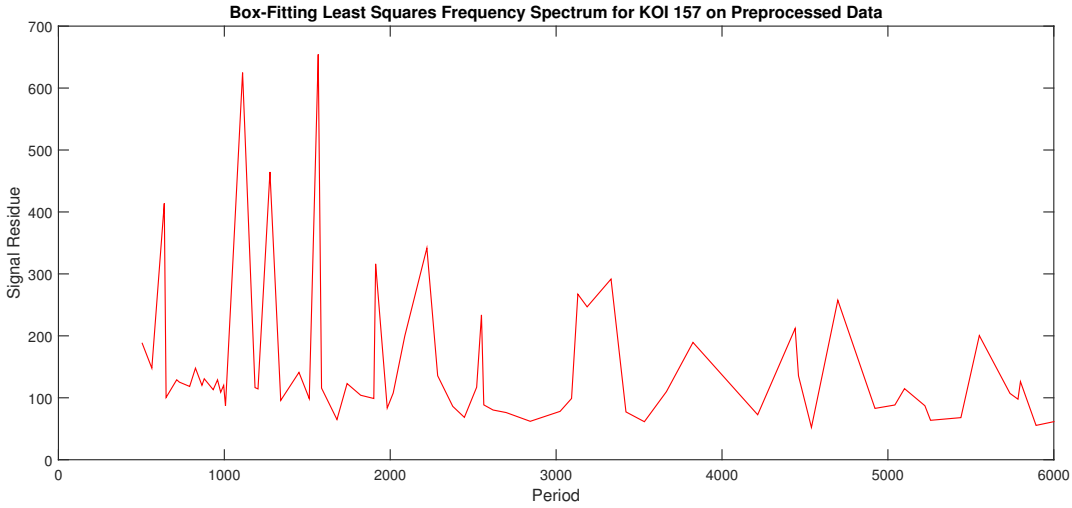


Figure 5.8: Signal Residue of KOI 157 after preprocessing using BLSAI.

Further plots of the Signal Residue demonstrate that if the original 18 quarter dataset is well behaved like for KOI 157, then the SR plots will be relatively unchanged after processing. The differences in the figures 5.7 and 5.8 are marginal. Note that the peaks in these plots that correspond to planets are at 638, 1110 and 1566. The peak at 1275 is just a harmonic of KOI 157.01. This is a very comforting result as it confirms that the '`gpml_pipeline`' has not skewed data which required little preprocessing prior to planet detection.

Comparison of Methods

A comparison can now be made between the vanilla NASA BLS algorithm which has no prior information over the periods with one which uses the peaks of the autocorrelation as the input periods. This has been done using the same test data which was used to plot figure 5.4. In the following figures 5.9 and 5.10, the red circles are the results of the current BLS algorithm and the blue dots correspond to incorporating the autocorrelation.

When looking at figure 5.9, the BLS with autocorrelation inputs has correctly identified the period of the planet in the synthetic data after a depth[2] of 1.4. This includes any classification of harmonics of the input period as a successful result as it has still detected the planet. Therefore all the blue dots in this figure fall on one of the harmonic dashed lines after a depth of 1.4 (apart from one point that falls on the fifth harmonic at 1.55). This result corroborates with that of the contour plot of figure 5.4 when moving along the line of constant period of 1,000. The depth of 1.4 just lies outside the SNR region greater than 2. This verifies that this contour plot gives a good estimation of the area that it is possible to detect planets.

[2]This is relative to independent and identically distributed Gaussian noise with a variance of 1.

Comparing the blue dots of the BLSAI method with NASA's, represented by the red circles, shows that they both struggle at the lower depths. It could be argued that the original grid search method is better at correctly classifying the actual period, as the BLSAI occasionally tends to predict harmonics of the input period instead of the actual one. This is to be expected as the autocorrelation of a transit signal with white noise will pick up these harmonics as its peaks and feed them into the BLS algorithm for the BLSAI. However detecting harmonics is still a successful result, although it would require a bit more data analysis. The tendency of the grid search of the periods to miss transits in NASA's version is evident by the two red circles that completely falsely predict the periods to be around 1,500 at depths of 1.75 and 1.85. Whereas the BLSAI algorithm performs well enough at this depth to pick up the period or a harmonic.

A further plot of comparing the two methods can be seen in figure 5.10 which looks at how they perform at a constant depth of 3. The blue dots and red circles have the same denotation as before and any points falling on the dashed lines correspond to a detection. Due to the need to set a lower bound on the period values used in NASA's BLS, BLSAI outperforms at lower periods. The lower periods are in fact where the BLSAI technique should perform at its best, indicated by the SNR contour plot. As the input period is increased, both algorithms are successful until upon reaching a period of around 6,000. After this, where the SNR is low, the algorithms are both shown to struggle to make correct detections. This is especially noticeable for the BLSAI's predictions which drop to a level of around 600. This is because of the way in which the code cuts off periods lower than 500 which causes the SR to be high at this cut off period when the SNR is particularly low. This cut off value can be experimented with to slightly improve the result for higher periods. This would then sacrifice the superior performance at lower periods which may not be desirable considering this is currently an advantage that this method has over NASA's original one.

Summary Results

Tables 5.2 and 5.3 summarise the results of six experiments to compare the original BLS method with the BLSAI method. These include the results from figures 5.9 and 5.10. Table 5.2 shows the percentage of successful detections at given depths over the range of periods 50 to 30,000 data points. Whereas table 5.3 shows the percentage of successful detections at given periods over the range of depths 0.5 to 3.5.

The percentages listed in the two tables show that the BLSAI method is superior for lower transiting periods, noting that for a period of 500, BLSAI had a success rate of 93% in comparison to 79% for the original BLS method. Therefore for lower periods and high transit depths the BLSAI algorithm

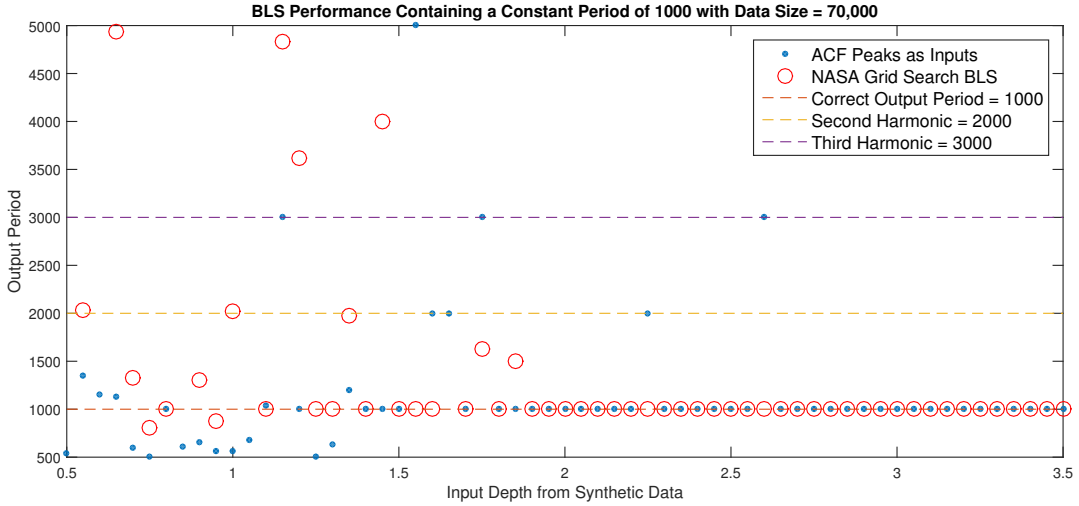


Figure 5.9: Application of both the BLSAI and the original BLS method to synthetic data of size 70,000 with a constant period of 1,000. The vertical axis corresponds to the predicted output period by the algorithms. Any values falling on the dashed lines are successful detections, where the lines correspond to the first, second and third harmonics of the period. The blue dots and the red circles are the outputs from the BLSAI method and NASA's BLS method respectively.

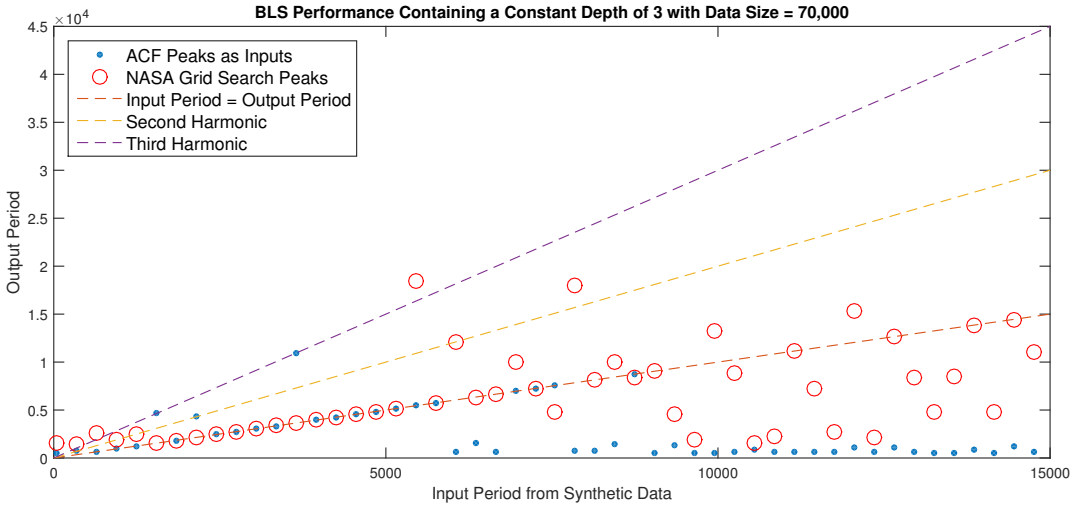


Figure 5.10: Application of both the BLSAI and the original BLS method to synthetic data of size 70,000 with a constant depth of 3. The vertical axis corresponds to the predicted output period by the algorithms. Any values falling on the dashed lines are successful detections, where the lines correspond to the first, second and third harmonics of the period. The blue dots and the red circles are the outputs from the BLSAI method and NASA's BLS method respectively.

Method	Transit Depth		
	1.5	3.0	5.0
BLSAI	8%	22%	40%
BLS	6%	26%	55%

Table 5.2: Percentage of successful detections for given depths, spanning from periods 50 to 30,000 data points.

Method	Planet Period		
	500	1000	2000
BLSAI	93%	75%	66%
BLS	79%	70%	70%

Table 5.3: Percentage of successful detections for given periods, spanning from depths 0.5 to 3.5.

is better at detecting exoplanets as the autocorrelation provides a strong prior. However when the SNR is low, the the original BLS method outperforms BLSAI. This is shown in table 5.2, where the range for a given depth spans up to a period of 30,000. For a depth of 3, approximately 80% of the transit signals along this range have an SNR of less than 1, resulting in BLS having a 26% success rate compared to BLSAI's 22%.

5.4 Summary

The detection of extrasolar planets in the 18 quarters of the Kepler Mission data has been successful when planetary signals are present. Initially, taking the autocorrelation of the preprocessed flux gave a very good indication of the periods as shown by developing table 5.1.

Then the BLS algorithm currently used by NASA was applied to KOI 87 to show that the preprocessing pipeline has led to the successful detection of KOI 87.01, which was not previously possible on the original dataset. It was then proposed to to take advantage of the autocorrelation by using the peaks to give prior information on the input periods for the BLS. This was with the aim of improving on the limitation of using a simple grid search to pick periods.

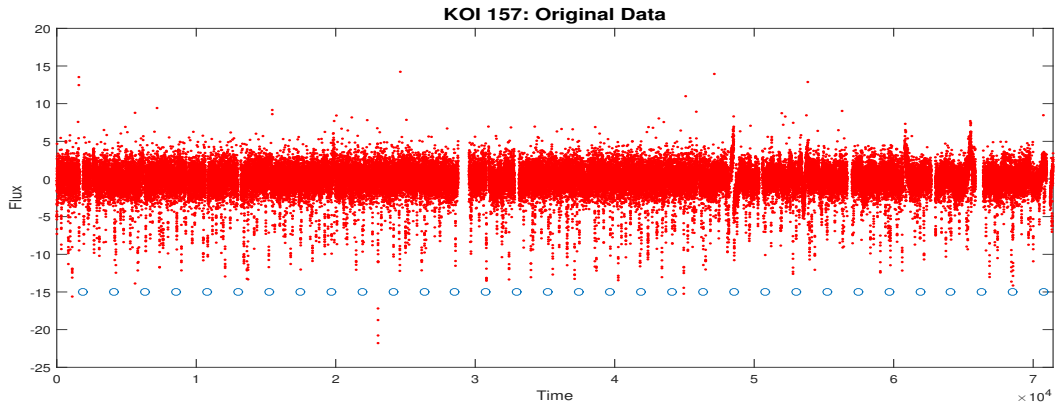
Both algorithms were applied to the same test dataset and a comparison was made. When the signal to noise ratio of the synthetic data was high, the BLSAI performed better than NASA's algorithm. However BLSAI did misidentify some planet periods as harmonics. To benefit from both methods, an algorithm that utilises the prior information from the autocorrelation along with the well established grid search algorithm should be developed. This could take a finer period grid search in regions that the autocorrelation suggests a planetary signal and apply a coarser general search to the rest of the data in a way similar to NASA's current algorithm.

Finally the Signal Residues of two KOIs were investigated to show the BLSAI in action and demonstrate that the '`gpml_pipeline`' performs well on both badly behaved data such as KOI 87 and on reasonably well behaved data such as KOI 157.

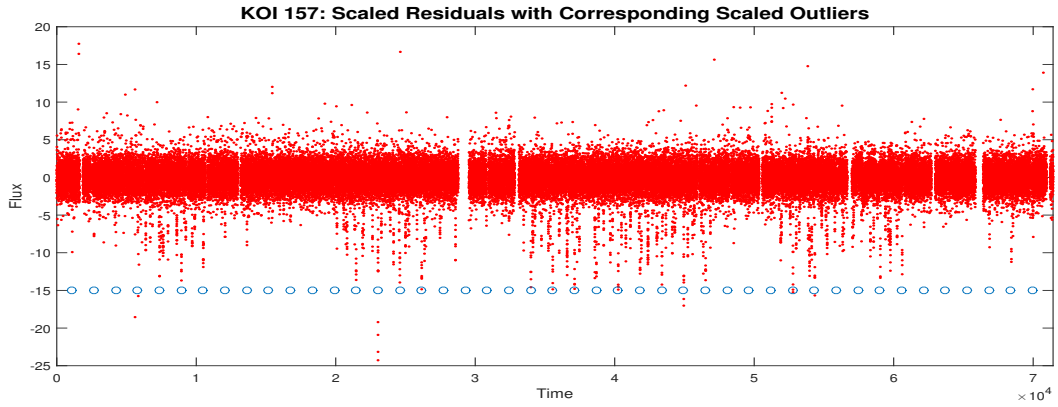
Chapter 6

Results

This section contains further results obtained through using the methods detailed in this report on different datasets. Figures 6.1a and 6.1b show the results of applying the BLS method to KOI 157, both before and after the pipeline. Figures 6.2a, 6.2b, 6.2c and 6.2d display the other four plots containing the results that produced tables 5.2 and 5.3.

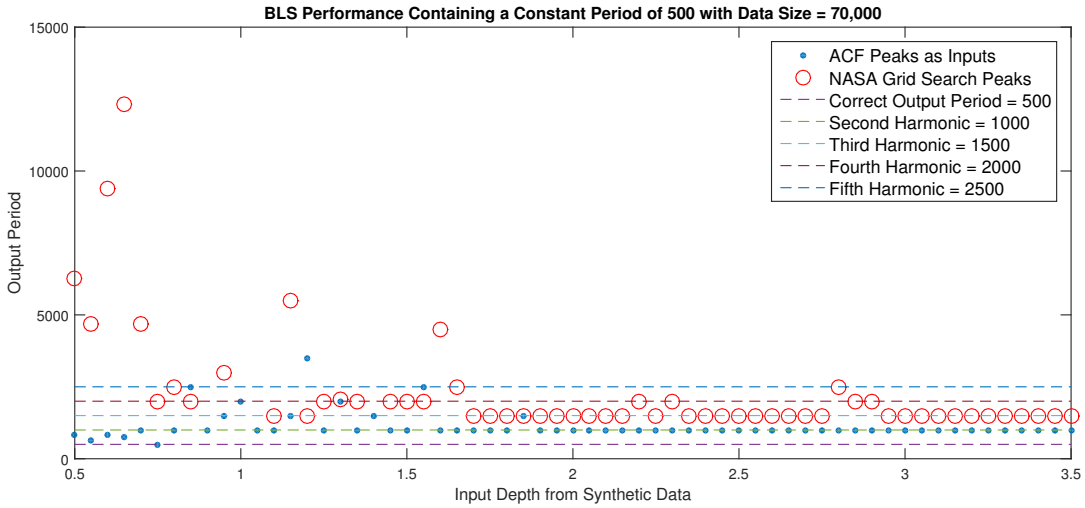


(a) KOI 157 flux before processing.

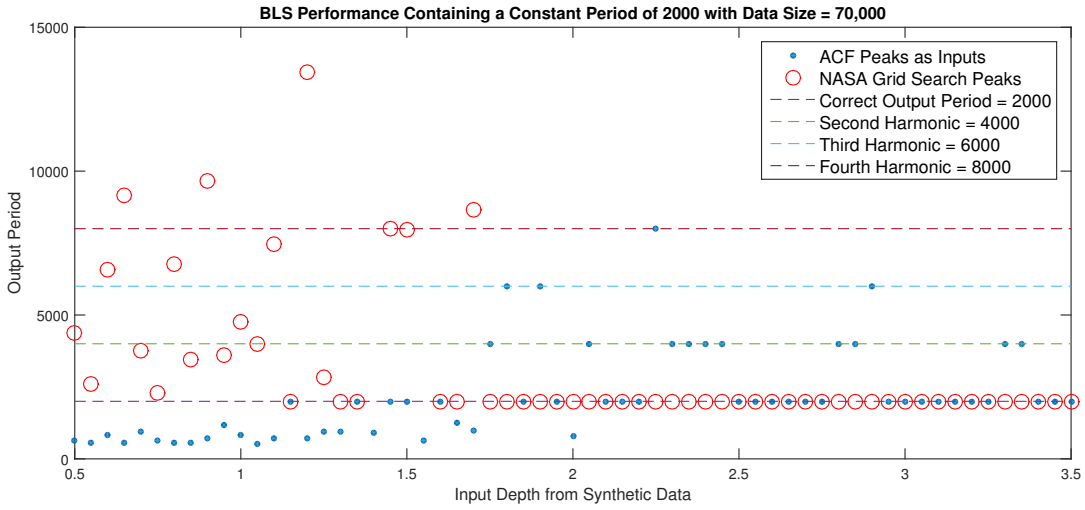


(b) KOI 157 flux after processing.

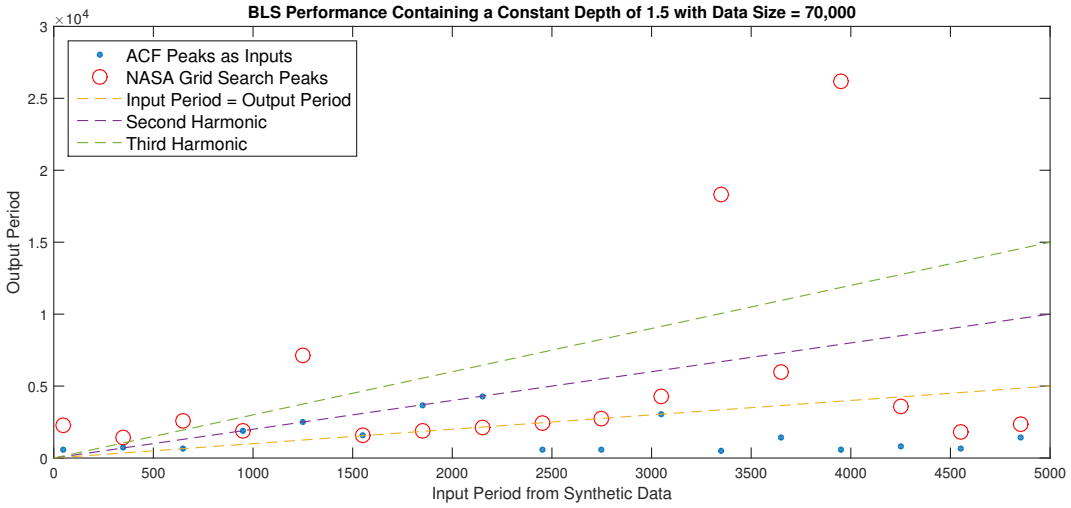
Figure 6.1: The successful detection of KOI 157.03 in the preprocessed data is denoted by the blue circles in figure 6.1b. This is compared to the incorrectly assigned transits by the BLS method in figure 6.1a. This verifies preprocessing in the pipeline before using the BLS algorithm.



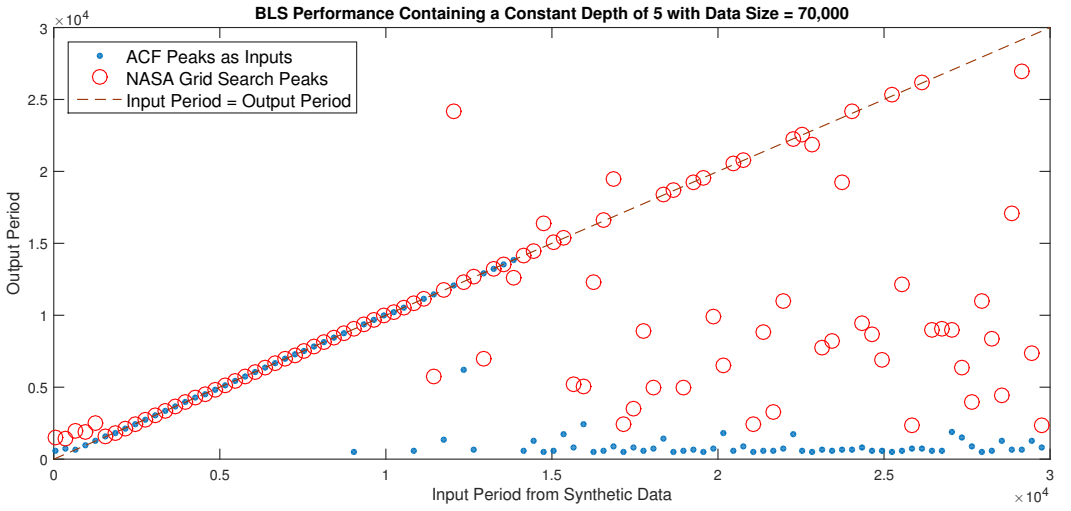
(a) Synthetic data of size 70,000 with a constant period of 500.



(b) Synthetic data of size 70,000 with a constant period of 2,000.



(c) Synthetic data of size 70,000 with a constant depth of 1.5.



(d) Synthetic data of size 70,000 with a constant depth of 5.

Figure 6.2: Application of both the BLSAI and the original BLS method to synthetic data of size 70,000. The vertical axis corresponds to the predicted output period by the algorithms. Any values falling on the dashed lines are successful detections, where the lines correspond to the harmonics of the period. The blue dots and the red circles are the outputs from the BLSAI method and NASA's BLS method respectively. These four experiments combined with figures 5.9 and 5.10 produced the results for tables 5.2 and 5.3.

Chapter 7

Conclusions

A new pipeline for planet detection in large astronomical datasets has been presented in this report. The use of the Gaussian process has allowed the behaviour of stellar variations and unwanted systematic trends of stars to be modelled with little prior information. This is the huge advantage of non-parametric modelling. Using the autocorrelation as a prior for the input scale length has been particularly successful in alleviating the problem of getting trapped in local minima when optimising the log marginal likelihood of the GP.

This pipeline has been successfully automated so that it does not require any input from the user once the data has been fed into the pipeline. Applying this algorithm to KOIs has decomposed the signal into Gaussian noise and potential planetary signals. This has achieved the subsidiary objective outlined in the original introduction of preprocessing large Kepler datasets along with stitching the quarters together.

Finally this data was then put through the current BLS algorithm. The results of which have demonstrated the success of removing trends from the data before applying the planet detection algorithm. This is a very promising outcome that was demonstrated to perform particularly well when the initial PDC flux behaved particularly badly as illustrated through KOI 87. The BLS algorithm, which has normally been applied to a much smaller set of data, was unable to detect KOI 87.01 in the original PDC flux but it was successful for the preprocessed flux.

The autocorrelation also has been shown to perform well on the preprocessed data in detecting planets and it was therefore investigated if it could be incorporated with the BLS method to improve on the basic grid search method that is currently in place. The prior information available from the autocorrelation led to encouraging results and it is suggested that more research should be put into this area to form an improved planetary detection algorithm, leading to many more exoplanet discoveries in the future.

Chapter 8

Further Work

The results from this report have successfully shown the that it is capable to detect transiting signals in the presence of stellar variability and other systematics in large datasets. Therefore it follows that the next stage would be to apply these techniques to K2 Mission data as the unwanted trends are particularly prevalent for this data and would therefore benefit from preprocessing before applying a planet detection algorithm.

Furthermore there are certain areas of the pipeline that could be improved through further research.

The logical continuation of this report would be to carry on investigating the incorporation of the BLSAI method with the vanilla BLS. As was briefly suggested at the end of chapter 5, a good starting point would be to edit the original code to include the prior information of the autocorrelation, along with using the standard grid search method for picking periods in regions where the ACF has no strong predictions of a particular period.

Further work could also be done to ensure that the outlier removal algorithms pick out all transiting signals. Making this more robust would perhaps lead to detecting more exoplanets as, at the moment, although there has been a high level of success, some planetary signals may be missed by the outlier removal algorithms.

Finally it is suggested that additional work could be put into looking at what should be a suitable threshold level for a transiting signal. This would be in order to completely automate the pipeline from processing the data to getting a result that indicates if a planet is present and a confidence on this outcome. This could potentially come from looking at the magnitude of the peaks of the autocorrelation and picking a suitable threshold level.

Bibliography

[1] Michel Mayor and Didier Queloz. A Jupiter-mass companion to a solar-type star. *Nature*, 378(6555):355–359, 1995.

[2] NASA. Table of confirmed planets. `http://kepler.nasa.gov/Mission/discoveries/`, 2015 (Accessed: 9 January 2015).

[3] NASA. Kepler Launch. `http://www.nasa.gov/mission_pages/kepler/launch/index.html#.VQrAWfmsVqU`. (Accessed: 22 March 2015).

[4] C.A. Haswell. *Transiting Exoplanets*. Cambridge University Press, 2010.

[5] Michael Perryman. *The exoplanet handbook*. Cambridge university press, 2011.

[6] GitHub. Introduction to Differential Photometry. `https://github.com/OSCAAR/OSCAAR/wiki/Introduction-to-Differential-Photometry`. (Accessed: 22 March 2015).

[7] Michael R Haas, Natalie M Batalha, Steve T Bryson, Douglas A Caldwell, Jessie L Dotson, Jennifer Hall, Jon M Jenkins, Todd C Klaus, David G Koch, Jeffrey Kolodziejczak, et al. Kepler science operations. *The Astrophysical Journal Letters*, 713(2):L115, 2010.

[8] Jon M Jenkins, Douglas A Caldwell, Hema Chandrasekaran, Joseph D Twicken, Stephen T Bryson, Elisa V Quintana, Bruce D Clarke, Jie Li, Christopher Allen, Peter Tenenbaum, et al. Overview of the Kepler science processing pipeline. *The Astrophysical Journal Letters*, 713(2):L87, 2010.

[9] D Foreman-Mackey. kplr, A Python interface to the Kepler data. `http://dan.iel.fm/kplr/`. (Accessed: 29 December 2014).

[10] Shawn Seader, Jon M Jenkins, Peter Tenenbaum, Joseph D Twicken, Jeffrey C Smith, Rob Morris, Joseph Catanzarite, Bruce D Clarke, Jie Li, Miles T Cote, et al. Detection of Potential Transit Signals in 17 Quarters of Kepler Mission Data. *arXiv preprint arXiv:1501.03586*, 2015.

[11] S. Aigrain, S. T. Hodgkin, M. J. Irwin, J. R. Lewis, and S. J. Roberts. Precise time series photometry for the Kepler-2.0 mission. *Monthly Notices of the Royal Astronomical Society*, 447:2880–2893, March 2015.

[12] Carl Edward Rasmussen. *Gaussian processes for machine learning*. Citeseer, 2006.

[13] S Roberts, M Osborne, M Ebden, S Reece, N Gibson, and S Aigrain. Gaussian processes for time-series modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984):20110550, 2013.

[14] N. P. Gibson, S. Aigrain, S. Roberts, T. M. Evans, M. Osborne, and F. Pont. A Gaussian process framework for modelling instrumental systematics: application to transmission spectroscopy. *Monthly Notices of the Royal Astronomical Society*, 419(3):2683–2694, 2012.

[15] Carl Edward Rasmussen and Hannes Nickisch. Documentation for gpml matlab code version 3.5. `http://www.gaussianprocess.org/gpml/code/matlab/doc/`. (Accessed: 15 October 2014.

[16] Michael Osborne. C24 Advanced Probability Theory. Gaussian processes. University of Oxford `http://www.robots.ox.ac.uk/~mosb/teaching/C24/2_slides_Gaussian_processes.pdf`, (Accessed: 24 March 2015).

[17] Scott E Maxwell and Harold D Delaney. *Designing experiments and analyzing data: A model comparison perspective*, volume 1. Psychology Press, 2004.

[18] ORACLE. Outlier Detection Methods. `http://docs.oracle.com/cd/E17236_01/epm.1112/cb_statistical/frameset.htm?ch07s02s10s01.html`. (Accessed: 22 March 2015).

[19] MathWorks. Confidence intervals for sample autocorrelation. `http://uk.mathworks.com/help/signal/ug/confidence-intervals-for-sample-autocorrelation.html`. (Accessed: 9 January 2015).

[20] M. R. Haas, N. M. Batalha, S. T. Bryson, D. A. Caldwell, J. L. Dotson, J. Hall, J. M. Jenkins, T. C. Klaus, D. G. Koch, J. Kolodziejczak, C. Middour, M. Smith, C. K. Sobeck, J. Stober, R. S. Thompson, and J. E. Van Cleve. Kepler Science Operations. 713:L115–L119, April 2010.

[21] Geza Kovács, Shay Zucker, and Tsevi Mazeh. A box-fitting algorithm in the search for periodic transits. *Astronomy & Astrophysics*, 391(1):369–377, 2002.