ON TREED GAUSSIAN PROCESSES FOR MODELLING STRUCTURAL DYNAMIC SYSTEMS

A thesis submitted to the University of Sheffield for the degree of Doctor of Philosophy in the faculty of Engineering.

Tianwei Zhang

Department of Mechanical Engineering

February 2018

University of Sheffield

Abstract

The aim of this PhD project is to develop an approach to particle damper modelling based on Gaussian process regression trees (GPRTs). Developing validated particle damper models is a prerequisite for being able to optimise their performance when applied, for example, in the damping of various components and assemblies for a range of frequencies. The objective of the model is to make a credible prediction of damper behaviour for a given set of design parameters and input excitations.

Particle damper modelling may be approached in one of three ways: analytically from first principles; numerically using finite or discrete element methods; or experimentally, with the structure and parameters of a reduced-order model inferred from measured data. This project focuses on the third of these approaches, with low-order, physics-based models being combined with experimental training data. The aim is to both identify the model structure that most accurately captures the behaviour of the damper and to infer the model parameters. The model can then be used to predict damper behaviour for design parameters and excitation conditions that have not been tested experimentally. This is made more challenging by the fact that particle dampers will in general, display nonlinear dynamic behaviour which can lead, for example, to switching behaviour in the observed responses.

Gaussian process (GP) regression has emerged as a powerful and adaptable approach for modelling experimental data. The basic GP is, however, best suited to modelling outputs that vary smoothly as a function of input values - something that is emphatically not the case for switching behaviour. An alternative is to develop a treed model which partitions the input space into regions that display smooth behaviour and to infer a separate GP model within each partition. This project will focus on the application of these approaches to particle damper modelling, including investigation of the contributing statistical techniques; impact of experimental design; and validation of developed models for previously unseen conditions. In addition to the works associated with the particle damping, in the later chapters of the thesis, the developed statistical model has also served an application to Structure Health Monitoring (SHM) problems, via which the generality of such a model has been demonstrated.

Acknowledgements

Is it the beginning of a thesis, or the end of all these days' intros and exits, in a drama made of 4 acts of years? Or is it the first note to this overture of science whose momentum is crescendoing with the instruments of technologies performing in this golden age; or the last brick to this architecture of dissertation whose foundation is still trembling with the hasted time dodging the falling of the deadline. Oh, in either way, bitter accomplishment or sweet endurance, let these gratitudes and appreciations, who are neutral for their movements are not toward me but to others, shall reconcile such civil war waged by my own heart.

GOD, thou art omnipresent and omnipotent, and thou art my saving grace when hope had once closed his last light for me during those time when my unfastened stage was drifting on the billows of diffidence, panic and despair. Now they all have gone, as thou granted'st me with your mightiest love.

Keith, some time my supervisor, some time my wise life teacher, some time my financial supporter and some time my friend, then let me be all time respectful, all time studious, all time obliged, and all time truthful to him.

Robert, my equally respectful supervisor, your assiduousness, meticulousness, and responsiveness are always those fertile lands where my gratitudes shall thrive and bloom.

Jem, my supervisor once back in time, you with your continuous supports that travelled along with these passed years, does make a salvo of my salutations in my memory.

Grandmother, my beloved and most missed, may your soul Rest In Peace. In my thesis, your place will never be ceded to time's oblivions.

Father and Mother, one ensures me security in finance, and the other one in confidence; one fights like my hero in his business, the other one shines like my sun in her kindness.

Pastor Chen, my Christian lighthouse, when I was drifting, you spotted me with the light of the LORD. Your lighthouse shall continue to beam in my thesis.

Terry and Gill, you with your opulence of ages have generously provided me with the ripe fruits of your benevolence and kindness with which my weak days softened and thawed away.

Xiao, my best friend, 3-year companionship with you has finally joined us together in the same sail in this thesis, and what sails here is our united friendship.

Tingting and Meini, you are both miracles to me, sweet, lovely and beautiful like always; and both are my heroines. Let me wish all those disagreements between you and me shall finally vanish.

David, Leslie, Jack, Philip, Geoff, Haichen, Ning, Suzhou, Emi, Sharafiz and all my other good friends, your help and supports have formed the panoply of the grand fireworks in my heart.

After I've spent up my wealth of gratitude, I now feel that nothing can measure my treasure's altitude.

Table of Contents

Acronym and Nomenclature

AID	Automatic Interaction Detector
ARD	Automatic Relevance Determination
BCART	Bayesian Classification and Regression Trees
CART	Classification and Regression Trees
CI	Confidence Interval
CTGP	Chipman-based Treed Gaussian Process
COF	Coefficient Of Friction
COR	Coefficient Of Restitution
CV	Cross-Validation
CGM	Conjugate Gradient Method
DEM	Discrete Element Method
FEM	Finite Element Method
FRF	Frequency Response Function
GP	Gaussian Process
GPML	Gaussian Process Maximum Likelihood
GTGP	Gramacy's Treed Gaussian Processes

iid	Independent and Identically Distributed
LLM	Limiting Linear Model
MAP	Maximum A Posteriori
MCMC	Markov Chain Monte Carlo
MHA	Metropolis-Hastings Algorithm
MLE	Maximum Likelihood Estimate
NM	Newton's Method
NOPD	Non-Obstructive-Particle Damper
NLCGM	Nonlinear Conjugate Gradient Method
PD	Particle Damping or Particle Damper
PDF	Probability Density Function
QNM	Quasi-Newton Method
RJ-MCMC	Reversible-Jump Markov Chain Monte Carlo
SDM	Steepest Descent Method
SE	Squared-Exponential kernel
SHM	Structural Health Monitoring
TGP	Treed Gaussian Processes
THAID	THeta Automatic Interaction Detector
WC	Wolfe Conditions
μ	Vector of means of a probability distribution
σ^2	Variance of a dataset or a univariate Gaussian distribution
Σ	Covariance matrix of a dataset or a multivariate Gaussian distribution
w	Vector of inference parameters for parametric regressions

x	Single input variable or point of training data
y	Single output variable or point of training data
X	A vector of x, normally refers to the entire input domain
Y	A vector of y, normally refers to the entire output domain
ϵ	noise, residual or error
K	Covariance matrix
f	Predictive or modelling function
σ_f^2	Function variance hyperparameter in the Squared-Exponential kernel
l^2	Length scale hyperparameter in the Squared-Exponential kernel
σ_n^2	Noise variance hyperparameter in the Squared-Exponential kernel
N	Normal or Gaussian distribution
IG	Inverse Gamma distribution
W	Wishart distribution
θ	Hyperparameter vector

Chapter 1

Introduction

Number is the within of all Things. —Pythagoras 570BC-495BC

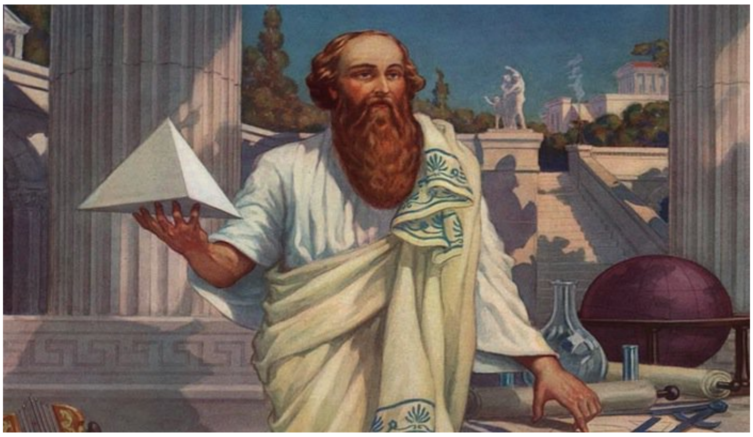


Figure 1.1: [1]Pythagoras with his pyramid.

Although not among the seven sages, nor the three greatest, as an illustrious philosopher, Pythagoras is widely recognised for his almost superstitious obsession with numbers. His idealistic realm dilapidated during his own time at the attack on irrational numbers. His belief was then largely treated with ridicule by many spectators sitting in history. If really *Number is the whithin of all things*, the number alone should be able to function as a tool that could piece the phenomenons together to unveil its hidden rules. The classic era of science didn't provide a stage for its chance to happen. But vicissitude is the eternal ingredient of time; entering the 21th century, a modern Neopythagorean trend is gathering tides in computer sci-

[1] Source: http://www.famousmathematicians.net/pythagoras/

ence. this trend is in the rising of Machine Learning (MA). Machine Learning is a branch of computer science. By its name, machine learning allows the computer to analogously learn like human beings at being given a certain form of training [1]. It has a close affinity with statistics, thus the study of machine learning is entirely data or number based with no direct involvement of real physical insights. Machine learning in the 21st Century flourishes mostly in developing artificial intelligence; its power now gradually permeates to other fields, and engineering is definitely one of them. The specific machine learning technique discussed in this thesis is a statistical regression model called Treed Gaussian Processes (TGP). Given training data, such a model can learn the hidden rules inside the data according to a series of criteria, essentially give predictions to the data space. Such a fact means that even if the data has physical meaning, such as output measurements from experiments, no physical insight is required to model the relation between the input and output in the data space. A corner of Pythagoras' realm is revealed here.

1.1 Uncertainty Analysis

The TGP is a state-of-the-art machine learning technique which is still, to some degree, in its experimental stage. It has not yet been formally applied to solving any industrial problems; while, all its applications are mostly subject to academical analysis to test its performance. As a derivation from the well applied Gaussian Process, the TGP is theoretically capable of dealing with any type of problems that a conventional GP is adapted to. The GP is a classic statistical model dealing with uncertainty in terms of either regression or classification. In the envelop of the main interest of this project, the regression, in most of the scenarios, the uncertainty as generally manifested as in a form of noise, can be accommodated properly by the conventional GP, if the noise level maintains uniform throughout the data. However, real world problems are not always benign in this respect; in fact, non-uniform noise propagation is a commonplace. The TGP offers the capability in dealing with the type of uncertainty whose property is subject to variation throughout the data space, which in statistical parlance, it is called non-stationary regression. In the broad context of structural dynamics, the rise of non-uniform uncertainty is often contributed by the change of physical properties inside the system. In this project, such a statement will be illustrated by two major examples, where the effectiveness of the TGP will be tested and studied.

1.2 Motivation from the Particle Dampers

Damping is a conventional engineering property describing the ability to attenuate vibrations in the system. As vibration represents a transfer of energy, the damping can be equivalently regarded as the ability of dissipating the energy involved in the vibration (potential energy and kinetic energy). In the real world, structural deformation will always incur energy loss due to friction between molecules at the microscopical scale, thus all materials do possess a certain level of damping. If a component is specifically designed for attenuating vibration, based on such application, in this way it is called a damper. The application of dampers is pervasive. One of the typical examples is the dashpot shown in Figure 1.2, which is a classic simple linear damper. It can often be seen in the shock absorber (Figure 1.3), one of the indispensable components in vehicles. The shock absorber is a classic paradigm for traditional spring-damper system design (the shock absorber is a bilinear spring-damper system).

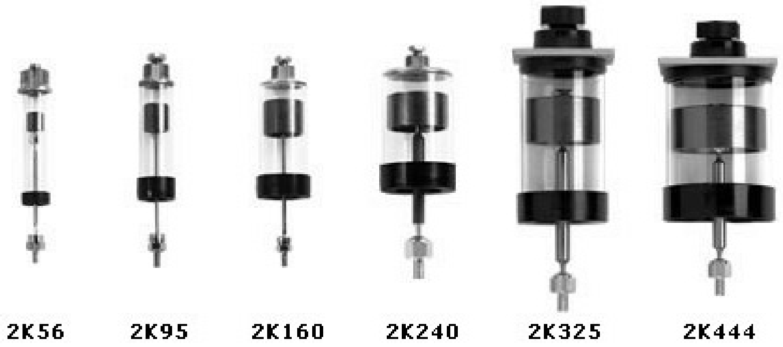


Figure 1.2: [2]Classic dash pots.

In contrast, a particle damper (PD) (Figure 1.4) comprises granular solids encapsulated in a container which can be attached separately or embedded in the vibrating structure. Because the microstructure of the PD can occur in many different configurations, the damping behaviour of the PD exhibits high levels of uncertainty. The particle damper has been extensively studied by researchers in recent decades because it is highly effective if applied correctly but is difficult to optimise. Different from conventional polymeric damping materials, the particle damper energy dissipating mechanism relies on particle-to-particle and particle-to-wall friction and inelastic impact [2]. The major advantage of the particle dampers compared to traditional damping methods is that the performance of the particle damper is insensitive

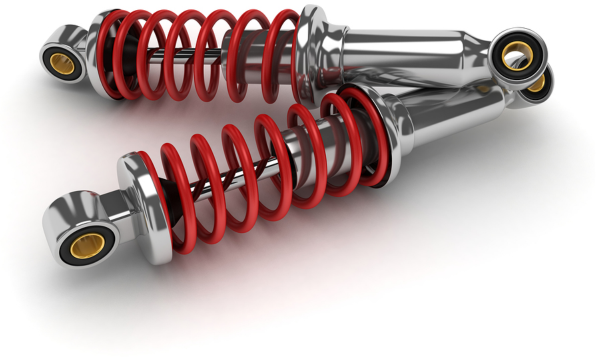


Figure 1.3: [3]Classic shock absorber.

to temperature, which makes the PD a better fit to harsh working conditions.

For example, the PD has been applied to an electronic package installed on a spacecraft in the Indian Space Research Organisation(ISRO) satellite centre, so as to mitigate the vibration during the launching process where high temperature is definitely a considerable issue [3]. Another industrial application is that the PD has been used in steering control system in the automotive industrial to reduce the noise and vibration of the electric motor [4].

From the industrial point of view, the advantage in resisting harsh conditions is doubtlessly a practical virtue. However, PDs are not in common use in engineering applications; indeed their use even in specific applications that appear aligned to their properties is limited. The main imputation goes to their highly nonlinear and uncertain behaviours which make them difficult to model and control.

Because the physics inside the PD is an entangling complication of multiple physical interactions, including particle collisions, inter-particle friction etc; to study the true detailed physics at a microscopical scale is unattainable in theory, considering the behaviour is rather chaotic and implicated with many factors. Using Finite Element based methods will also be too excessively demanding in computational power, considering the size of the particles needing to be modelled. To study at a macroscopic scale through equivalently treating the PD as a simplified system with the aid of appropriate assumptions will lose generality and reliability. Therefore, instead of

[2] Source: http://airpot.com/product-category/product-lines/dashpots-shock-absorbers/airpot-precision-dashpots/

[3] Source: http://www.roshfrans.com/cambia-el-alternador-de-tu-vehiculo/

dealing with all the difficulties in the physics, the machine learning technique is used to bypass such problems.

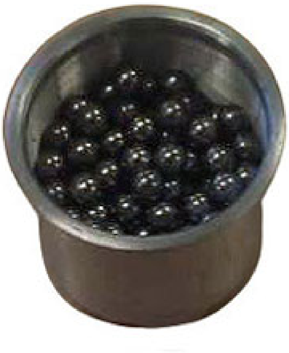


Figure 1.4: [4]A particle impact damper.

1.3 The selection of the Treed Gaussian Processes

There are numerous machine learning techniques that can be employed to model the PD. To select the statistical modelling tool, the Treed Gaussian Processes (TGP), is motivated by the characteristics of the behaviour of the PD during vibration. The PD exhibits three general phases at different levels of vibration amplitudes and frequencies as pointed out by Saluena et al. [5] (more details in Chapter Literature review). Such a fact indicates that the damping from the PD should hold piecewise characteristics. The TGP stands out, because it is built on a decision tree framework which, in the end, can generate regions in the data space, and these regions might be able to associate relations with the piecewise behaviour of the PD. Ideally the TGP model can identify these piecewise behaviours by producing corresponding regions through partitioning the data space. The TGP is a state-of-art model that has not yet been applied to the PD data, thus it is also worthwhile to explore its effectiveness. Moreover, the TGP used in this project is a new developed one of its type by the author. Therefore, this project is not merely about borrowing present statistical tools to apply to novel cases, but more importantly about the development of a new statistical tool which has its own novelty and uniqueness. It can be expected that the TGP model can apply to other problems in the broad engineering circle. In this thesis, the TGP model will also be applied to solving problems in Structural Health Monitoring.

[4] https://www.etronixab.se/2017/03/30/pid%C2%A0particle-impact-damper-34665129

1.4 Objectives

The TGP itself was first introduced by Robert Gramacy in 2004, since when concept of the TGP is almost patented under his name. However, the functions of the TGP can be achieved through different mathematical approaches. In this project, the chief objective of this thesis is to develop a brand new Treed Gaussian Process which has never been studied before, in contrast with one developed by other researchers (Gramacy), and then demonstrate its performance in applications to real engineering problems. Then such a brand new TGP will be applied to studying the behaviour of the particle damping, where no TGP has ever been used. To fulfil these purposes the following sub-objectives have been set:

- Fully absorb and understand the present TGP.

  This will involve a comprehensive study of the present TGP, followed by implementations of this in MATLAB.

- Develop a new TGP model that will function with advantages over the present TGP.

  This will involve a comprehensive study of decision tree models, Bayesian analysis etc.

- Design and conduct an experiment on particle damping.

  The aim of designing and conducting the particle damping experiment is to acquire data encoding typical particle damping behaviours, so that it could be used as a general example to demonstrate the TGP's effectiveness.

- Apply the TGP on the data measured

- Extra applications of the TGP in other engineering fields to demonstrate generality.

  In order to demonstrate the generality of the PD, more data are needed. In this case some Structure Health Monitoring data will serve the purpose.

1.5 Summary of Chapters

Chapter 2

Chapter 2 contains a literature review focusing on particle damping and the development and application of treed Gaussian processes. The review on the particle damping generally focuses on the traditional modelling methods, mainly to highlight their drawbacks. The review on the TGP focuses on its history of development and its applications.

Chapter 3

Chapter 3 establishes the extensive mathematical background to the TGP task. It builds from ground to top as intending to show how the new TGP evolves from the old. This chapter also addresses inks on some side mathematics whenever necessary.

Chapter 4

It is a benchmarking chapter for the new TGP against the present one. The benchmarking will be done in two respects, the performance and the computational cost. These two items are quite universal on any algorithm-based benchmarking.

Chapter 5

The design and process of the particle damper experiment will be shown here. It includes the design of the method of measurement, the choice of materials and components as well as their parameters. The results will also be shown here and discussed on the grounds of general physical insights.

Chapter 6

The TGP will performed on the PD data, the result will be discussed in terms of both mathematics and physics. Some efforts will be made to link the pure mathematical results with the physical insights to show that the TGP has a potential to aid the physical exploration of the PD rather than completely overwriting it.

Chapter 7

Presents a new case study on bridge data. The primary purpose is to demonstrate the general applicability of the TGP algorithm. This study is within the scope of

Structure Health Monitoring, thus a completely different role for the TGP to adapt to.

Chapter 8

Conclusions for all chapters.

Chapter 2

Literature Review

2.1 Chapter Overview

In this chapter, the review of the past literature will be segregated into primarily two parts, namely the review on the particle damping (PD), and on the relevant machine learning (ML). As the raw initial motivation, the reviews on the particle damping will focus on its modelling history with methods from various different departments of engineering, this will lead eventually to the avant-garde innovative thoughts brought by the machine learning from which the Gaussian process stems. The review of the TGP relevant content will focus on its predecessors, considering the Treed Gaussian Processes is a comparatively state-of-the-art statistical modelling technique. Some reviews on its industrial and engineering applications will be also addressed.

2.2 Particle Damping

While the damping behaviour of particle dampers has been studied intensively during the last two decades, the first reported use of a particle damper traces back over 70 years, to when Lieber and Jensen were attempting to suppress aircraft flutter related vibration by introducing a single particle damper [6]. However, note that sand bags have been used to reduce the shock from artillery for much longer. It is simply that nobody considered this as a particle damper - although that is exactly how they were working. Initial development of particle dampers focused on the design of a separate supplementary component to the primary vibrating structure. In 1989, Panossian [7] first introduced the concept of the Non-Obstructive-Particle-Damper (NOPD) which is a category of built-in damping component of the vibrating structure by filling particles in structure holes or cavities at desired locations.

2.2.1 Analytical Modelling Methods

For the purpose of applying particle dampers pragmatically, it is required that the nature of particle dampers is investigated, so that the performance of the particle damper can be controllable and adjustable. Against the inconvenience brought by high nonlinearity, researchers have exploited many different models to characterise particle dynamics experimentally, theoretically and numerically. Using the fact that a standard particle damper is the derivative of single particle impact damper, Masri in 1970 introduced an equivalent model for particle dampers by considering the particle bed as a single equivalent moving particle subjected to two equi-spaced symmetric impacts [8]. Though his early theory was refuted by Bapat (1983) due to inapplicability in a gravity field, many researchers have conducted subsequent investigations based on the heuristic of the single equivalent impact damper idea [9] [10] [11]. Liu et al. (2005) attempted to model the nonlinear PD damping system as an equivalent viscous damper through taking linear snapshots at different levels of excitation when studying a disk geometry particle damper [2]. Most of these analyses of particle dampers have provided good agreement with experiment results. However, the application of the models is rather limited to prerequisite conditions such as certain frequency ranges, free vibration, and particular types of input excitation. Besides, the damping performance of particle dampers shows at least three highly different types of behaviour with correspondingly different levels of damping; this implies that none of these theoretical equivalent models is universal[5]. Moreover, many studies of the equivalent particle damper only consider the friction and impact between the particle and container walls with no consideration of the internal energy transfer between particles, indicating a non-compatibility with the Three Phase states of particle dampers.

Apart from the equivalent impact damping models, many authors have addressed other solutions to characterise particle dampers. Chen et al [12] developed a technique which used a restoring force surface for the characterisation of particle dampers, and their results agree with some of the findings discovered by other researchers.

Wu et al. [13] proposed a theoretical model of the granular particle damping in transient vibration-based on the multiphase flow theory of gas particles. The theory originates from the work of Fan and Zhu [14], where they illustrated that the granular particles contained in a cavity of a vibrating structure can be regarded as a multiphase flow of gas particles with low Reynolds number. According to their theory, the momentum transfer between particles is governed by the pseudo-shear stress and the viscosity of particle interactions. The theory is validated by reasonable agreement with experimental data acquired from a cantilever beam attached with a particle damper. However, the functionality of this model is restricted by a number of assumptions (e.g. moderate particle damping ratio).

Martin et al. [15] managed to associate the particle damping system with fluid mechanics by introducing the hydrodynamic model for a vibrofluidised granular bed. The model is an analogous process to relate the dynamic properties of a granular system to the thermodynamic properties of gas. By such an analogy, the mean fluctuation kinetic energy of the particles is also known as the granular temperature. The model is validated by the comparison of granular temperature with the experimental data. However, as a prerequisite, the use of a hydrodynamic model is restricted to simulate the fluid phase of particle damping systems, and a number of assumptions such as a Maxwell-Boltzmann distribution also determine the area of applicability.

2.2.2 Discrete Element Methods

For the purpose of obtaining full scale understanding of the particle damper, a reasonable modelling technique is more likely to take place at the particle scale level. During recent decades, making use of developments in computational techniques, the Discrete Element Method (DEM) has been considered as an adequate solution to the modelling of particle damper systems. The pioneering work of DEM was proposed by Cundall [16] in the 1970s for considering the study of rocks. As an explicit-based process, the DEM computes resultant forces subjected to each particle in the system with small time iterations. Generally, the DEM can be classified into two categories, which are known as soft sphere and hard sphere models. Hard sphere models allow no overlap between particles and therefore do not model the complex contact/interaction mechanisms. The simulating computation only relies on the coefficient of restitution along with particle and container dimensions. As each of the variable time steps indicates a new collision, the hard sphere DEM is also named as an Event-Driven simulation method [17]. Apparently, the computation executes relatively faster by assuming no deformation mechanism, but such simplification could lead to a reduction in the accuracy of the simulation results as well as a lack of meaningful understanding of energy dissipation by inelastic deformation of particles during collisions. McNamara and Young [18] reported a phenomenon associated with inelastic particle collision modelling (mostly in the hard sphere model), which is known as inelastic collapse. Such a phenomenon can often be observed when the number of collisions among a group of particles tends to approach infinity during a finite amount of time (finite time singularity). The inelastic collapse can cause the relative particle velocities to approach zero exponentially during their collision. In their subsequent work [19], they proposed a solution to the inelastic collapse problem called the quasi-elastic limit which sets limit values for the coefficient of restitution and the number of particles.

For soft spheres, the contact mechanism specifying the normal and tangential interactions needs to be discussed in the models [20]. Normally, the contact concerned with particles and walls will be modelled by springs and friction interface. Since there are more specifications in the soft sphere model to simulate, in conjunction with the large number of particles, it is necessary for one to make a decent trade-off between computational cost and accuracy.

The DEM is highly computationally consuming. Though both methods provide computation over millions of particles in an individual simulation, it is still debatable to consider such a system as macroscopic. Some authors such as Vermeer et al. [21] and Poschel et al. [22] carried out efforts to develop a micro-macro transition method so that the relative microscopic simulation can be used to predict and study the true macroscopic dynamic system within the framework of a so-called macroscopic continuum theory. Benefiting from the use of the DEM method, researchers have made significant progresses in characterising the particle dynamic system.

Saluena et al. [5] in 1999 published a paper (widely cited since then) where they expounded the three phases encountered by the particles at different levels of excitation amplitude and frequency. With the aid of molecular-dynamics simulation, they arrived at a final summarising diagram known as the particle damper phase diagram (the three phase states), and the diagram is shown in Figure 2.1.

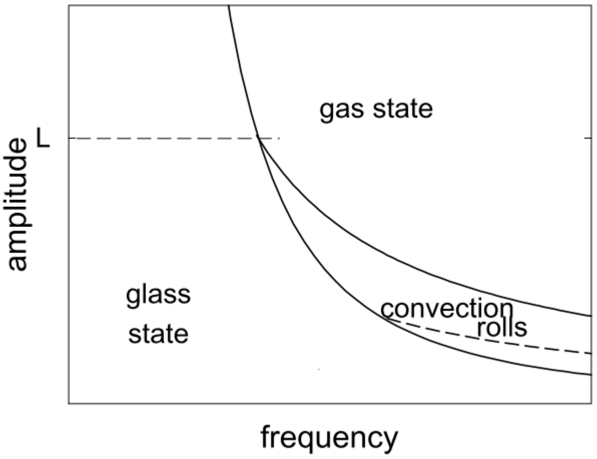


Figure 2.1: Particle damping phase diagram, see [5].

As indicated in the diagram, the state of the particle damper during vibration could be classified into three phases known as the Glass state, Liquid State (visually as convection rolls in particle view, so the name convection rolls, see [5] for more information) and Gas state. The presence of the exact state depends on both excitation amplitude and frequency. As one can imagine, for a constant frequency, the particles in the container will stay in contact with each other at low amplitude excitation, and gradually start to bounce around with the increase of amplitude indicating a gas-like state.

In terms of the damping performance at different phases, they also defined a parameter b referring to the ratio between the averaged dissipated power per cycle and the mean translational kinetic energy of the system. By plotting the effective acceleration $\Gamma = A\omega^2/g$ against the damping parameter b, one can arrive at Figure 2.2. (A is the excitation displacement amplitude, ω is the angular frequency, g is the gravity constant, and b is a ratio between the cyclic energy dissipation rate and the translational kinetic energy in the system, for more detail, see [5])

The different symbols in the diagram correspond to different frequencies. The glass transition point is defined at $\Gamma = 1$, below which the parameter b varies significantly with less correspondence to the Γ. In the glass state phase, the energy dissipation is strongly governed by the packing configuration of the particles. During this phase, since almost all the particles remain *in situ*, the inter-friction between particles and between particle and wall will dominate the dissipation of energy. At the fluidised

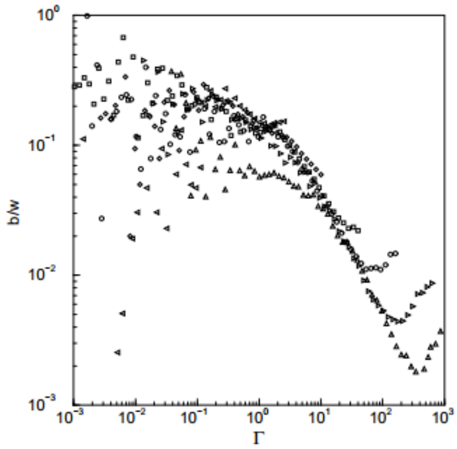


Figure 2.2: Effective damping vs effective acceleration, see [5].

states (convection rolls and gas), the energy dissipation is governed by a combination of collision and friction. Peculiarly, in the gas state, where the collision (or impact)-based dissipation overwhelms the friction-based dissipation, the total energy dissipation is perceived to be proportional to the collision frequency of unbounded particles.

Equipped with DEM, Mao et al. [23] made a successful characterisation of particle dampers with comparisons with both dry-friction damping and impact damping models. The specific damping capacity with a maximum instantaneous value of near 50% can be reached by the corresponding particle damper accordingly. A typical vertically-vibrated closed particle damping system was studied by Sǻnchez and Carlevaro with the implementation of DEM [24]. The study was analytically investigated in a manner of chaotic dynamics. Both Poincǻre maps and maximum Lyapunov exponents were employed to distinguish regular from chaotic orbits for both the primary system and granular bed. Through the simulation and analysis, they showed that the particle grains gradually switch from a periodic motion to chaotic motion with the increase of excitation frequency. Damping was found to be small at low frequency excitation due to lack of relative motion between particles. Chaotic behaviour has been confirmed in high frequency excitation, which shows good agreement with Saluena's work. There are signs indicating the chaotic transition from quasi-periodicity. The optimum granular damping is achieved at frequencies close to the resonance of the primary system as well as revealing a window of periodicity according to the Poincare map. The decline of damping performance when entering the chaotic state does comply with Saluena's work.

The influences of a number of system properties on the particle damping performance have also been studied using DEM. Saeki in 2001 [25] investigated the granular damping in a horizontally-vibrated system. The agreement between DEM and experiment shows that the increase of mass ratio will enhance the damping performance and high values of particle radius and mass ratio will lead to an increase in the optimum cavity length.

Wong et al. (2007) [26] set out to improve characterisation for the particle damper by carrying out both experiment and DEM simulation on a vertically-vibrating poly methyl methacrylate (PMMA) column filled with granular materials. They focused on studying particle damping at different values of coefficient of restitution, coefficient of friction and particle stiffness. Though results have shown that the major part of the primary system energy is dissipated by friction, the actual coefficient of friction does not seem to have a significant effect on the energy dissipation in most cases. A promoted role for the coefficient of friction could be observed when the coefficient of restitution is greater than 0.9. Large changes of contact stiffness were also confirmed to be a factor affecting the energy dissipation.

2.3 Treed Gaussian Processes —the stochastic Modelling

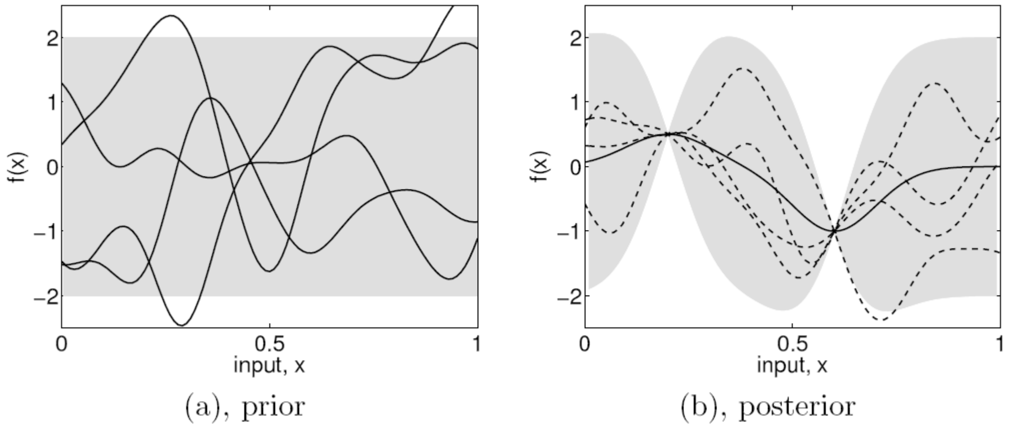
Particle dampers may exhibit different dynamic behaviour during operation due to the Three Phase States. Such characteristics of particle dynamics has made most of the predicting models fail or be restricted to a certain limited applicability. However, recently researchers came up with idea to detour the inherent physical complexity of particle dampers by predicting their behaviour utilising machine learning techniques. Possible data acquisition experiments could be done on particle dampers to collect necessary data which is used as training data sets for predicting the untested data of interest (eg. amplitude & frequency).

The advancement of computational capacity in recent decades has housed the practicability of many numerically-based simulations. Not only DEMs or their close correlative, the Finite Element Methods (FEMs), have been extensively used and voluminously published on. Somehow those old ideas regarding sampling-based speculating derived from old gambling tables have gained their strength for much more extensive applications. Statistical modelling, or more precisely the statistical inference, is what is being spoken here. Statistical inference is a purely data-based mathematical model, that by itself, contains no direct physical insight to its application. It could be said, given a set of data, the statistical model is used to infer out the statistical property (e.g. p-value, mean and variance) of that data system based on some certain rule. For example, given a set of data, through a proper statistical modelling, one could be able to give predictions to the behaviour of the data through providing a predictive curve characterised by the mean and variance. On the gambling table it is like given a number of trials of dice casting, and make a prediction on its next incident. To the particle damping modelling, one could picture the measured data are actually observations (such as numbers on a dice) to a hidden unknown rule (behaviour of the data such as trend), and the statistical modelling focuses at constructing that rule with the mean and variance.

The two main porticoes supporting the statistical inference are the classical Frequentist statistics, and Bayesian statistics [27]. Frequentist statistics is not the subject of study in this thesis, despite its dominance in the last century. Its debate with the Bayesian statistics has formed a series of remarks in statistical history [27], a further discussion on it will deviate the thesis away from its main theme. However, Bayesian statistics once lived in a long shadow cast by the frequentist statistics. Its rehabilitation generally started from 1980s as attributing to its coalescence with the Markov Chain Monte Carlo (MCMC) and the growth of the computational power which in general allow the samples to be drawn faster and more efficiently. In 1990s, the Bayesian statistics came in confluence with the rising trend of machine learning techniques, and discovered its crucial place there [28]. Many approaches could be associated with the Bayesian machine learning which is built on the framework of Bayesian inference.

One of them is the Gaussian Process Regression Tree (GPRT), also known as the Treed Gaussian Process (TGP). The TGP is a recently produced mathematical model developed by Gramacy [29]. The essence of the TGP is the amalgamation of the Gaussian Processes (GP) and the Classification and Regression Trees (CART). Tersely speaking (more details in the chapter on theory), the GP is a statistical regression model used for making predictions on a provided data space, while the CART model is a binary tree generation process which is used for dividing the data space into a group of sub-spaces.

The GP is not a recently invented mathematical method, the first application of the GP refers to Kriging (1951) [30]. However, it is only a recent event for the GP to be used in the scope of machine learning. As a stochastic process, the Gaussian Process is a collection of random variables, any finite number of which have a joint Gaussian distribution [31]. The Gaussian process is specified by its mean function and covariance function. In a sense of simplification, any function can be perceived as a very long vector assembled by data points, at each data point; there is an input and output pair (X, Y). The Gaussian Process defines that at every such point there is a Gaussian distribution over output Y with a mean function $m(x)$ and covariance function $k(x, x')$ which establishes relations to other points in the function. In the field of supervised machine learning, analogous to the traditional Bayesian study, the prediction of function (regression) is determined by the posterior which is derived via Bayes Rule (for more information see [31], briefly, the posterior is proportional to a prior times a likelihood) with specified Gaussian Prior as well as the provided training data information. This can be roughly explained in a pictorial way:



(a), prior (b), posterior

Figure 2.3: (a) prior sampling (b) posterior distribution, see [31].

Figure 2.3 depicts a naive but comprehensive way to explain the prediction of function via stochastic machine learning. Imagine that there are initial prior beliefs in the form of the function under prediction (e.g. Gaussian), then sample such functions with such belief in the prior space. If some of the data points of the function are known (Figure 3(b) shows 2 given data points), with the correlation between points specified by the covariance function, the posterior space will be conditioned and produce the confidence interval. Clearly, the increase in the number of known data points will further condition the posterior with more fixed points and a more concentrated confidence region.

Gaussian Process Regression has proved to be a reliable solution in machine learning, but limitations show that if the relations between input and output are governed by multiple functions, the performance of GP will deteriorate. Considering the Three Phase behaviour of particle dampers, one is certainly expecting to partition the input space into regions for better use of GP. Such expectation sets a journey to the application of the Classification and Regression Tree (CART).

The CART model recursively partitions the predictor space into regions (leaves) to form a binary tree structure whose leaves contain more homogeneous datasets. Several generations of the CART have been developed in the past. The earliest conceptual formation of the prototype of its type dates back to 1959, when Belson [32] stated the biological matching process could be equivalently treated as a prediction process, and he illustrated such with a tree in Figure 2.4. In 1963, Morgan and Sonquist introduced the Automatic Interaction Detector (AID) algorithm for growing a binary tree [33], which gained contemporary popularity. The AID is a binary tree developed for piecewise constant regression. Unlike the later more robust tree models, the AID can only progressively make splits rather than removing splits in the process. The AID had been richly applied during that period to a number of problems in practice; most of those applications are in the field of social science which is the common natural fit for statistical experiments. As one of the exception, Cellard et al. (1967) pioneeringly extended its application into engineering where they used the AID to study the effects of environmental and technical factors on the gripping of locomotive engines. Messenger and Mandell (1972) introduced the first classification tree known as THAID based on the AID [34].

Breiman et al. (1984) first introduced the *Prune* operation into the construction of the binary tree in place of the stopping criterion in the AID and THAID [35]. This is considered as a big leap in the development of the binary tree models, because the introduction of the *Prune* substantially improved the robustness of constructing the tree. This was proved beneficial in the later stochastic treatment of the construction of the tree [36]. In their revolutionary paper, they also first introduced randomness into the tree model that the linear splits could be obtained by random search.

Quinlan et al. (1992) [37] first developed the tree for stepwise linear regression, which greatly inspired the later researchers like Torgo (1997) [38] to install function models in tree leaves.

Chipman et al. (1998) [36] applied the Bayesian framework to stochastically construct the tree. Such an approach was later known as the Bayesian CART (BCART).

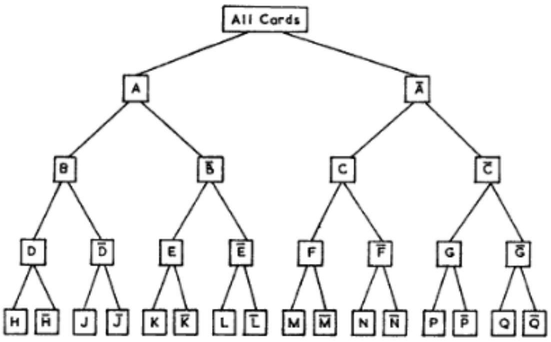


Figure 2.4: Belson's tree for Biological classification, see [32].

This BCART is also the foundation for the TGP developed in this project. The BCART is generally comprised of two parts: prior specification and stochastic search. Basically, Bayesian CART intend to achieve more promising CART models via a stochastic search guided by the posterior distribution. At the end of the search, a number of selections could be made upon these tree models based on a variety of criteria like maximum likelihood etc. In contrast to the traditional greedy algorithm used by other authors [18], the search strategy involved in the Bayesian CART is a fully sampling based one that requires a much higher computational power to implement in practice.

Breiman et al. (2001) made another striking effort in this area by embedding the idea of random forest into the tree model. The random forest allows large set of CART trees to be constructed from bootstrap samples [39](Bootstrapping is a classic estimating method for systematic statistical properties based on random sampling with replacement).

Kim and Loh (2003) [40] innovatively introduced the idea of multiple sub-nodes for classification trees, which allows more than two sub-nodes to be generated from one node through growing the tree.

The sprouting of these new, or once new, ideas in the development of the tree model has indeed made it more applicable and adaptive to various different problems, and naturally demonstrated its capability and reliability. Later in the union of the GP and the tree model by Gramacy, the potential of the GP regression has been more effectively exploited with the aid of the decision tree-based model which provides better data space for the GP to exert its power.

Currently, there is a lack of applications of the TGP in practice. Its reliability in dealing with real world problems has not yet been demonstrated. But it is highly promising to expectantly see a good performance from an industrial application of the TGP, since the decision tree model has plenty of practical experiences with the combination with other regression models, such as linear regression, logistic regression etc.

Here are some examples. Yusuf et al. (2007) used the classic CART model in Zoology to study the weaning weight in farm animals [41]. Druedahl and Munk-Nielsen(2017) did a study on the Danish national income data with a regression tree equipped with a generalised linear model, and they did arrive at a satisfying modelling of the income's uncertain behaviour exhibiting high-order dynamics [42]. Hecker and Kurner (2007) published a paper on applying a linear regression tree to predicting the mobile terminated call for the study of the cellular traffic in the area of Braunschweig and Hannover. They latter found the use of the CART improved the overall predictive accuracy compared to the more traditional approach of Linear Multiple Regression Analysis [43].

Likewise, the decision tree models have made a wide presence to many research fields. It could also be sensed in these researches that the primary heuristic for using the decision tree-based models is the existence of characteristic changes in the data space. Therefore in the same vein, the three phase based characteristic change in the PD has raised enough enticement to the application of the decision tree based models.

2.4 Conclusion

The modelling of the particle damping is traditionally based on analytical approximation and computational discrete element methods. The analytical modelling has been voluminously studied from the middle of the 20th century, during which numerous assumptions, equivalent treatments and relevant theories have made their efforts of attempting to decipher the hidden physical rules hidden inside the particle damping system. However, no matter these methods made their presence in a sense of succession with improvements or innovation with amendments, all failed to serve their original purpose, being a general approach to model the PD. The later coming discrete element method unveiled its power in realistically modelling the real world PD system, but is limited heavily by the computational power. The recent exploding spreading of the machine learning methodology has introduced a new dimension for the researchers to jump over the aforementioned predicaments in modelling the PD. The TGP is one of the approaches offered, whose mathematical property offers a natural solution to modelling the PD. However, to apply the TGP in the context of particle damping is an engineering area currently not dabbled in by the researchers. In fact, even stepping back to an even broader range, the application of a regression tree model on modelling engineering uncertainty has rather limited presence under the scope of either academics or industries. Despite the lack of abundance in relevant researches, both the GP and the decision tree based models have been under research for decades; their capabilities are doubtless. Also considering the recent resurgence of Bayesian statistics over the last two decades, one could take a positive outlook on the future of machine learning. To study the TGP on the PD is definitely a small spot in the research area, which no one dabbled in before, but also a strengthened one at the edge of the development, pushing and expanding.

Chapter 3

Theory —Treed models and Gaussian Processes

3.1 Chapter Overview

It is not rare to observe physical systems that exhibit piecewise behaviour where the quantification or qualification of system factors could be modelled in a discrete fashion as allocating regions.

When it comes to the case of particle damping or some SHM scenarios, behaviours on both the physical or pure mathematical account, can feature a sense of shifting or switching of phase. For example, as introduced in the literature review, particle damping has been reported to behave in analogy to a form of Gas, Liquid or Solid. There is obvious switching of phase in terms of physical observations. In SHM, for example, the Z24 bridge has one switching dependence on the environmental temperature as a result of the stiffening of asphalt [44].

In the cases mentioned above, the treed model offers a natural solution derived from its structure which could be equivalently viewed as a process of allocating regions. In order to allocate regions with reasoning based on the case, the treed model has to be implanted with a 'brain' which defines its functionality, criteria, principle etc. For the particular concern of this project, the treed model is intentionally developed for the very purpose of regression-based analysis. Therefore, the treed model will be comprised of mainly two parts, namely the tree structure and the regression sub-model.

As subject to both parts, there are a series of versatile options to select for both the purpose of better satisfying the requirements from the real case. For the tree structure, the essential idea shines light on its dynamical manipulation of the structure so to find the very structure fitting the data space best. This manipulating process could be executed in either a probabilistic or deterministic way. This option of being probabilistic or deterministic also remains as an availability for the regression sub-model as well. Eventually, this leads to the Gaussian Process as the main consideration for this project, and this nomination is discussed briefly in the coming paragraph. More detailed discussion will be seen in other sections of this chapter.

A Gaussian Process (GP) is a nonparametric regression model (NPRM). Nonparametric regression models are inherently well-suited for modeling real world scenarios in which unknown nonlinearity dominates the associated physics. The literal expressive terminology 'nonparametric', to some degree, is a bit misleading, considering that a nonparametric model still has to contain modelling parameters to guide the regression. Differing from a parametric model, the parameters within non-parametric model could be specified as informative of the change of characteristic within the data (more details later).

The Gaussian Process (GP) as a branch of NPRMs can provide generality, robustness and reliability to cases with different degrees of nonlinearity. The GP bases its inference on a full Bayesian framework. By specifying different covariance functions, the predictions could be made vastly versatile.

In this chapter, a rather intuitive progress will be taken to unfold the development of a TGP through a simple deterministic tree. As to form the completeness of the argument, any related side mathematical theories will also be addressed and expounded in detail to a certain extent.

3.2 Decision Trees

The aforementioned tree structure, for the use on this particular project, refers to a Binary Decision Tree (BDT). A decision tree is a logical mapping process through which the elements of a given input space will be assigned into different groups represented by leaves of the tree based on a series of criteria [45]. A binary tree basically means the criteria take the form of a simple choice of YES/NO, thus branching the underlying space into two sub groups. Through repeated application of this unit process, a tree structure will be established. Figure 3.1 shows a typical BDT. In brevity, the BDT may be thought of as a tangible representative framework for partitioning a set of data. From the pictorial presentation of the tree, the vertex on the top is the root of the tree (decision trees grow from the top towards the bottom). All the inequalities are known as the splitting rules which specify whereabout to place the split or partition in the data space. Excluding the root, those nodes without being marked with letters are internal nodes of the tree which is unobservable in the data space. At last, the nodes marked with letters (A,B,C,D,E) are leaf nodes (or just leaves or external nodes), they are the characteristic regions as intended by the purpose of the decision tree.

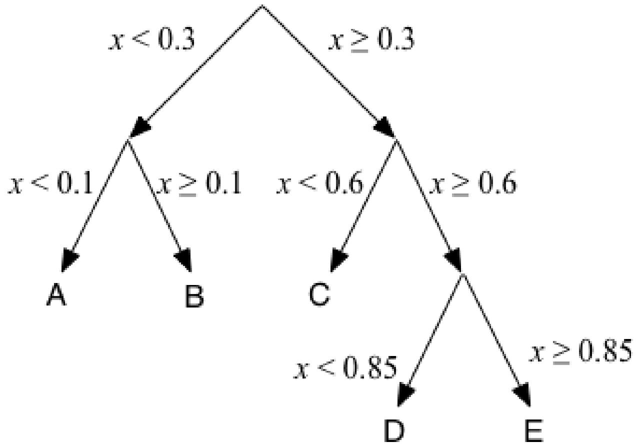


Figure 3.1: Typical Binary Decision Tree.

Because of the possession of an intrinsic binary mapping, the BDT naturally requires a logical sequential construction of the tree structure. This sequential construction is schemed in a rather fixed form, in which splits are supplied through proposing YES/NO questions to all the leaves in the current tree. Generally in two ways, the structure of the treed framework could be established, namely: deterministic and probabilistic. For a deterministic treed framework, mostly referred to as the Greedy Algorithm (GA), the algorithm will set forth the Y/N questions to each existent leaf by the order of its node number. On the contrary, a probabilistic tree structure generates splits in a random manner, it could either add splits or remove splits or alter the existent splits. Using which execution to perform on the tree is a pure stochastic matter. One thing worthy of notice is that whether the tree structure is deterministic or probabilistic, it does not necessarily mean the same thing to the whole treed model. The definition of the treed model will be addressed later.

The construction of the tree cannot accomplish without the proper settings on the splitting rule. Basically the rule takes its format in a way of whereabout to place the split within the data space. However, to deploy such a split requires a certain set of splitting criteria for the sake of logical reasoning. The criteria here are basically a global metric to evaluate the quality of the tree, and it is completely akin to the regression sub-model in use. Or in other word, the regression sub-model decides the type of criteria. Depending on the type of regression sub-model chosen, the treed model could be further classified into 3 categories, namely: fully-deterministic, semi-probabilistic, and fully-probabilistic.

Fully-deterministic refers to the treed model with deterministic treed framework incorporated with a deterministic splitting criterion. The corresponding criteria could be, for example, Mean Square Error (MSE).

A semi-probabilistic treed model establishes its framework on the probabilistic basis, but reasoning the partitioning of the space with a deterministic criterion.

A fully-probabilistic model basically encodes the whole treed model in a probabilistic manner, in which the treed framework and the splitting criterion will be arranged as probabilistic and remain mutually influential.

In the following sections of this chapter, with respect to all the three types of treed models, exemplary models are subject to detailed investigations for the purpose to compare and contrast, and more importantly, presents a logical tour for the evolutionary process of building the current TGP model.

3.3 Deterministic Trees

The treed model, at its very simplest, could explore the splits in the data space via a deterministic way. The primary principle inside a deterministic tree is the Greedy Algorithm. The terminology Greedy Algorithm is decently descriptive to its character whose greediness aims at reaching a global optimum through a series of steps of satisfying the local optimum. The quantity to be optimised in this particular circumstance for a simple parametric regression usually refers to error measurement or sum of Bayesian posterior probability of the tree. For this particular project, the deterministic tree developed in pertinence to arriving at the final TGP, is a simple linear regression tree featuring a greedy algorithm and a Mean Square Error (MSE) splitting criterion.

A MSE metric for measuring the quality of a linear curve fitting is almost the most conventional curve fitting strategy. From the mathematical perspective, the MSE is the sum of all the squared vertical deflections of each data points from the fitted curve. The general expressive form is,

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y_i} - Y_i)^2 \tag{3.1}$$

where

n = number of training data points;

$\hat{Y_i}$ = predictive Y_i due to curve fitting;

The MSE is the second moment of the error, and judging from its expression, it is intuitive that such a metric reflects both the influence of predictive mean and the variance. As qualified to be a decent choice for a treed model, the MSE stands out for its neatness and simplicity. More specifically, the essential part is that the local error at each training data point will not lose its generality across leaves, thus is eligible as a standard metric to evaluate the prediction from a treed model.

The specific mechanism of the linear deterministic regression tree could be epitomised into the following procedures:

1. Grow the selected leaf of the tree by placing the split to the corresponding data space (root corresponds to the entire data space);

2. To place the split, the algorithm will sweep through all possible splitting locations, and at each of these locations, it will place down a pseudo-split;

3. Evaluate the sum of MSE from the generated left and right leaves under that pseudo-split, record this MSE in an array [MSE];

4. Repeat the evaluation and record for all the pseudo-splits at all the possible splitting locations in the selected leaf;

5. Find the lowest MSE from array [MSE], and identify its splitting location, and this splitting location will be the splitting rule for that leaf;

6. Repeat the procedures one to five for the next leaf whose selection is conveyed in ascending order in terms of node number in the tree;

7. A leaf will cease to split, if the lowest MSE is not less than the previous MSE if the leaf remains unsplit;

8. The entire process will stop when all leaves cannot be further split.

By the large, a deterministic tree gains its effectiveness from believing the global optimal could be approached through a series of local optimal steps. Under this presumed premise, encoding its core with a simple deterministic splitting criterion can substantially reduce the computational cost, though essentially it is wired with deliberate redundancy in exhaustive evaluation. However, still as a non-sampling based method, it can provide great accuracy, and extricate one from issues such as bad convergence rate etc.

The defective aspects of the deterministic tree are considered to be mostly inevitable. As it has been already addressed that, the greediness of the encroaching way to reach the global optimum, at its truest sense is entwined with 'Discreditability' and 'Profligacy'. 'Discreditability' basically refers to the false or inaccurate ultimate global optimum it eventually obtained. The theoretical viability of the Greedy Algorithm is established on the theoretical belief that any discontinuity on the general trend of the data will induce a conspicuous rise in the MSE. This theory holds its own in most scenarios with a decent tolerance on the accuracy of the exact placement of splits. However, there are some cases where the Algorithm completely fails its fulfillment by giving splits at apparent faulty locations.

'Profligacy' refers to the fact that since every possible splitting locations will be evaluated, this fact could be considered as an implication of great loss in the efficiency. And also because of this inefficiency, a deterministic tree will not be practically feasible to collaborate with splitting criteria encoded with probabilistic inference, or equally speaking, it is best to be applied for parametric regression where analytical solutions or simple numerical solutions to the regression function do exist.

3.4 Probabilistic trees

To start with the probabilistic tree, the first model to be set under scope will be something preeminent for its achievements in pioneering and expanding the field. This particular model is the Bayesian Classification and Regression Tree (BCART) for constant regression introduced by Chipman et al. in 1998 [36].

3.4.1 Bayesian Classification and Regression Tree (BCART)

The BCART essentially is a semi-probabilistic model, as a matter of the fact that the splitting criterion is deterministic conveyed through a probabilistic optimisation process. On the behalf of the formation of the tree structure, the process is completely probabilistic by taking its form in a Markov Chain Monte Carlo (MCMC)-based random walk.

In the context of statistical theories, a Markov chain, formally debuted in the early 20th century, is a stochastic process to simulate the random walk among states whose inter-relations are presented as the probability of commuting from one state to another. The following picture Figure 3.2 shows a typical Markov Chain. The circles represent the state in the Markov space, and the Capital P_{ij} next to the arrows indicate the transition probability.

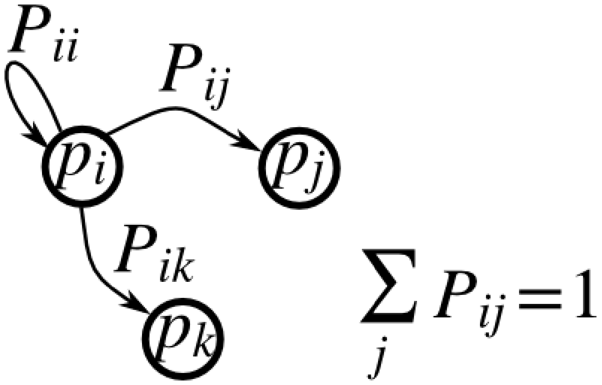


Figure 3.2: [1]Typical Markov chain.

In the current BCART model, the tree structure along with its splitting rule (basically this combination is referred to as the tree) will be represented as a state in the Markov chain. Hence, the construction process of the treed model is, in a

[1] Source: http://www.scipy-lectures.org/intro/numpy/exercises.html

certain sense, tantamount to the random walk in the Markov space, and the chief objective for the random walk is to expediently discover the states satisfying the splitting criterion (in a global sense) best. Accordingly, the random walk will perform in a guided way which embodies in two aspects: 1. The alteration of the tree structure, which is equivalent to the pace of the walk in the Markov space. 2. The Metropolis-Hastings Algorithm.

For the first point, to represent the entire tree as a Markov state is somewhat a bit abstract and intricate, because any nuance made to a tree (e.g. same structure, different splitting location) will give birth to a complete different tree requiring to be treated as a different Markov state. In this sense, the Markov space will be populated with an enormous number of states. It could be further demonstrated through this: if a one dimensional data space contains n data points, there could potentially be $n-1$ splits at its maximum. If one split is put down, there are $n-1$ different partitioned space; if two splits are presented, there are $(n-1)(n-2)/2$, which is equivalent to $(n-1)!/(2!(n-3)!)$; for 3 splits, the number will be $(n-1)!/(3!(n-4)!)$; hence for $n-1$ splits, it is $(n-1)!/(n-1)!(0!) = 1$. And each partitioned space with m splits will have $m!$ trees possible to be fitted to it, considering the difference in the splitting order. Thus the total number of possible states will be,

$$Number\ of\ possible\ states = (n-1)!\sum_{i=1}^{n-1}\frac{1}{(n-1-i)!} \tag{3.2}$$

Then according to convergence property of the sum of the reciprocal series of factorial,

$$\sum_{n=0}^{\inf}\frac{1}{n!} = e \tag{3.3}$$

Hence if the n is large enough(eg.$n > 20$) ,the number of possible states will be nigh $(n-1)!e$, which leads to great difficulty in exploring the Markov space. Therefore the pace of the random walk must be, to some degree, efficient enough to travel around the Markov space within a reasonable amount of steps, and it also should be partially trackable, as the newly reached state as a tree, should be a derivation or modification based on the previous state. In association with these requirements, Chipman gives the following tree alteration executions (or jumping criterion in the Markov space).

- *Grow*: Add one partition by splitting one leaf node of the tree
- *Prune*: Remove one partition by joining two sibling child nodes
- *Change*: Relocate an existent partition by changing splitting rule in the tree
- *Swap*: Find two internal nodes who are mutually parent and child, and swap their splitting rules

Grow and *Prune* are basically the pair of executions to determine the optimal number of splits. *Change* is set for finding the optimal location of the partitions, and *Swap* is intended to find the most promising tree structure based on the depth of the tree. Hence each execution has its own distinguishable character, which preserves a certain uniqueness in the context of the alteration of the tree.

Guiding the pace of the random walk will improve the agility of the MCMC walk, but not yet the efficiency of it in terms of moving towards the desired region in the Markov space. Therefore, more guidance targeting at the walking direction needs to be set. In the statistical community, this guidance could be led by the application of a Metropolis-Hastings Algorithm (MHA).

An MHA is a sampling strategy dedicated to sampling from probability distribution whose profile unknown or conforms to no known PDFs which allows a comparatively direct way for sampling (e.g. Box-Muller Transformation based sampling for the normal distribution with accessible uniform sampler [46]).Even though the type of sampling probability distribution can be directly sampled from (eg. Normal distribution) , the loss of information such as normalising constant (eg. marginal likelihood constant in Bayesian inference) can still cause troubles to sampling process. Bridging with the BCART model, each state in the Markov chain has a related posterior as inferred from Bayesian inference, and the entire collection of all the posteriors forms a discrete probability distribution of the tree. Apparently, it is more than the non-quantitativeness of the tree can inflict predicament on establishing a regular probabilistic distribution for the posterior. The MHA offers a route to simulate the sampling through a series of acceptance or rejection decisions on suspending states sampled from simple distributions.

The mechanism operates in the way below:

1. Randomly choose an execution option from *Grow*,*Prune*,*Change*,and *Swap*;

2. Walk to a new tree state by altering the tree according to the selected execution, and infer out the optimum posterior for that tree, record this posterior as P^*;

3. Then evaluate the Metropolis-Hastings ratio A;

4. Compare a uniformly sampled random number a in [0,1] with A, if $a < A$, accept this new tree state. Otherwise, abort the alteration.

In this 4-step process, the Metropolis-Hastings criterion has undertaken a crucial responsibility for the decision making on the acceptance of the walked new state. This criterion has the following mathematical expression,

$$A = \frac{P(T')}{P(T^*)} \frac{Q(T^*;T')}{Q(T';T^*)} \tag{3.4}$$

Where T' denotes the current state of tree.

Q is the transition probability between states, and P is the posterior probability for the corresponding state.

The transition probability for any pair of tree states of the BCART model, if mutually accessible, is defined by the specific execution chosen to commute between both states. As the execution is uni-directional, which only defines the probability of transferring towards the next state, but the backward travel does not share the same probability. For example, assume there are two tree states A and B. If A to B is conducted by *Grow*, thus B to A is *Prune* to form a counterpart. In notational form,

$$\begin{gathered} Q(A,B) = Q(\mathit{Grow}); Q(B,A) = Q(\mathit{Prune}) \\ Q(A,B) = Q(\mathit{Prune}); Q(B,A) = Q(\mathit{Grow}) \\ Q(A,B) = Q(\mathit{Change}); Q(B,A) = Q(\mathit{Change}) \\ Q(A,B) = Q(\mathit{Swap}); Q(B,A) = Q(\mathit{Swap}) \end{gathered} \tag{3.5}$$

Then further details on the relationship between the transition probability and the executions are explained by the following equations,

$$
\begin{aligned}
Q(Grow) &= \frac{1}{4}p(terminal\ node\ of\ grow)\\
Q(Prune) &= \frac{1}{4}p(internal\ node\ of\ prune)\\
Q(Change) &= \frac{1}{4}p(internal\ node\ of\ change)\\
Q(Swap) &= \frac{1}{4}p(internal\ node\ of\ swap)
\end{aligned}
\tag{3.6}
$$

For *Grow* and *Prune*, they are mutually reversed executions to each other, thus the choice of either will result in the numerator and denominator of Q becoming a pair of Q(*Grow*) and Q(*Prune*). And Q(*Grow*) refers from number of of total terminal nodes as a contrast to the number of internal nodes for the *Prune*. This contrast will nullify the cancellation of the Q ratio, which is valid for *Change* and *Swap* where both executions are reversed executions to themselves.

The posterior probability of each sampled tree is basically the product of the posteriors from each of the leaves in that tree. This probability is the key quantity and metric to reflect intuitively how likely the tree could be rated as the tree producing promising predictions to the data space. Its source of reliability stems from the traditional Bayesian analysis. The central part of the Bayesian analysis is the well-known Bayes rule taking in the form,

$$
posterior = \frac{likelihood \times prior}{marginal\ likelihood} \tag{3.7}
$$

In terms of a treed model, the performance of the prediction on the data space is purely dependent on the specific partitions presented by the tree, if other factors, such as the type of regression, are considered to remain fixed during the entire inference process. Consequently as the only global variable, the tree will be of the chief interest to be inferred out its highest posterior. However, as a non-quantitative factor, its prior specification has to associate with other features. Particularly for a geometric structure, the natural choice will be the depth of the tree. Therefore a decent assumption on the prior of the tree could be that this specification shall set preferences favouring the tree with less complexity or less depth. The following prior specification set forth by Chipman, serves this assumed purpose well,

$$prior = p(T) = \alpha(1+d)^{-\beta} \tag{3.8}$$

Where α defines the initial base acceptance probability and β defines the decay rate with the increase of the depth of the tree; d refers to the depth of the tree.

This neat and succinct specification of the prior of the tree does encode the initial preference over the trees with less depth. Once the prior has been set, the likelihood in a form of $p(\theta|T)$ could be treated as a conditional posterior on the present tree. Therefore, this leads to a sub-Bayesian inference centred at the predictions on the data in each leaf,

$$p(\theta_n|D_n, T) = \frac{p(D_n|\theta_n, T)p(\theta_n|T)}{p(D_n|T)} \tag{3.9}$$

Where D_n= $(X,Y)_n$ refers to the data in each leaf n, and θ_n is the regression parameter in leaf n. In an equivalent expression,

$$p(\theta_n|X_n, Y_n, T) = \frac{p(Y_n|X_n, \theta_n, T)p(\theta_n|T)}{\int p(Y_n|X_n, \theta_n, T)p(\theta_n|T)d\theta_n} \tag{3.10}$$

In evaluating the posterior from a Bayesian analysis, the fact of reasoned speculation requires an initial specified type of probability distribution on both the likelihood and prior. The marginal likelihood on the denominator is a constant completely determined by the likelihood and prior. However, not every pair of specifications on the prior and likelihood can collaterally yield an analytically obtainable marginal likelihood, as owing to the common predicaments in analytically evaluating intricate integrals. Thus, a convenient way to resolve this problem is to use a conjugate prior. A conjugate prior is a relation-wise specification of the prior and likelihood pair, so that the resulted posterior distribution possesses the same form of the distribution of the prior. This type of special conjugate relation entitles the arrival of the posterior without the computation of the marginal likelihood term on the denominator. It is somewhat a bit analogous to using trial method to solve the Ordinary Differential Equations such like $ay'' + by' + c = 0$, where the special form of $y = C_1e^{r_1x} + C_2e^{r_2x}$ suffices as the general form of the solution to y.

In the BCART model, for the purpose of simplicity in both Bayesian deduction and generality, the choice of the conjugate prior will be a Normal Inverse Gamma (NIG) distribution coupled with a Gaussian likelihood. In the spirit of a model for constant regression, the conjugate prior and likelihood pair could be specified in two ways as in compliance to the users preference, namely a Mean shift model (MSM) and a mean-variance shift model (MVSM).

The prior part of the mean shift model below, in the spirit of an NIG distribution, is essentially a separate specification of a Normal and an Inverse Gamma distribution. The NIG distribution is formally defined by the multiplication of the Normal and Inverse Gamma distribution.

$$\mu_1, \mu_2 ..., \mu_b | \sigma, T \quad i.i.d. \quad \sim N(\bar{\mu}, \sigma^2/a) \tag{3.11}$$

$$\sigma^2 | T \quad \sim IG(v/2, v\lambda/2)(\leftrightarrow v\lambda/\sigma^2 \sim \chi^2_v) \tag{3.12}$$

where as a constant regression, μ is the predictive constant; v and λ are the IG parameters which are treated as known; The conversion sign in the bracket shows $v\lambda/\sigma^2$ could be sampled from the v Degree of Freedom (DOF) Chi-square distribution.

Of the part of likelihood, the specification below encodes that the y value in each leaf is identically distributed in a manner of Gaussian distribution with a conditional mean and fixed variance across all leaves.

$$y_{i1}, y_{i2} ..., y_{in_i} | \theta_i \quad i.i.d. \quad \sim N(\mu_i, \sigma^2) \qquad i = 1, 2, ..., b \tag{3.13}$$

The Mean variance shift model is a minor modification of the mean shift model in terms of the expressive mathematical form,

$$\mu_i | \sigma_i \quad \sim \quad N(\bar{\mu}, \sigma_i^2/a) \tag{3.14}$$

$$\sigma_i^2 \quad \sim \quad IG(v/2, v\lambda/2) \tag{3.15}$$

In comparison of the conjugate prior of both MSM and MVSM, the indication of the distinction has been completely undertaken by the subscript of the σ. This subscripted denotation manifests that the MVSM, in contrast to the MSM which unifies the σ across leaves, allows each leaf to retain its own distinguishable character in the σ. This individual independence brings the advantageous impartation of robustness and generality of a certain degree to the algorithm. And correspondingly, the likelihood will be specified as,

$$y_{i1}, y_{i2}..., y_{in_i}|\theta_i \quad i.i.d. \quad \sim N(\mu_i, \sigma_i^2) \qquad i = 1, 2, ..., b \tag{3.16}$$

The conjugate specification above, according to the conjugate property, will consequence a posterior analytically derivable. For NIG conjugate prior to Normal likelihood, the posterior for the MSM and MVSM are given below,

$$p(Y|X,T) = \frac{ca^{b/2}}{\prod_{i=1}^{b}(n_i + a)^{1/2}} \times (\sum_{i=1}^{b}(s_i + t_i) + v\lambda)^{-(n+v)/2} \tag{3.17}$$

$$p(Y|X,T) = \prod_{i=1}^{b} \pi^{-n_i/2}(\lambda v)^{v/2} \frac{\sqrt{a}}{\sqrt{n_i + a}} \frac{\Gamma((n_i + v)/2)}{\Gamma(v/2)} \times (s_i + t_i + v\lambda)^{-(n_i+v)/2} \tag{3.18}$$

where c is the generated constant from the deduction, which is independent on T; s_i=(n_i-1)var(Y_i); $t_i = [n_i a/(n_i + a)](\bar{y}_i - \bar{\mu})^2$.

The set of parameters(v,λ,$\bar{\mu}$,a) are treated as known prior to the implementation of the algorithm. The choice of values for this parameter set could be led by reasons with the knowledge of observed Y values as the guidance and trace. In depth, for the MSM, the knowledge of the observed Y values is equivalently the same as knowing the standard deviation of the data space(denote as $s*$). Since the presence of a Treed model is intended to better fit the data, thus its standard deviation σ should be less than the natural dispersion of the data as quantified by $s*$. Then in consideration of the deterministic tree in the upper section, its exhaustive searching behavior, in many cases, will consequence a partition space prone to over-fitting which features a standard deviation s_* lower than majority of the σ as related to other partition patterns in the data space. Thus it is reasonable to arrive at the presumption of the inclusion in the interval $[s_*, s*]$ for the σ. Accordingly, v and λ could be tuned to accommodate the majority of the PDF of σ in this interval. Further guidance on $\bar{\mu}$ and a could be achieved subsequent to the choice of v and λ, as to ensure that the prior of μ will cover the whole range of Y values within a reasonable probability level.

For the MVSM, this guidance preserves its effectiveness, except with a little adjustment in terms of the guiding direction for σ. Different from the MSM, the MVSM has an inclination to over explain the variance in the data space. Because each of the leaves preserves a complete freedom in possessing its own σ, this could basically be translated as that the MVSM has the potential of fitting each leaf with extremely high resolution, which sometimes could be unrealistic. As interpreted in mathematics, the σ tends to approach the left end of the interval $[s_*, s*]$. To deploy against this potential unrealistic fitting, the v and λ are tuned in such way that the σ has a concentration of probability spanning the centre range of the interval.

For a broad consideration, the choice of either model should entitle the dependence of the selection of v and λ on the complexity of the tree, as the depth of the tree does lead to finer partitions where over-explanation of variance is more common to occur.

The existence of analytical evaluation of the curve fitting posterior, with the fully established prior model for the tree structure, forms the basis for Bayesian inference for the entire tree,

$$p(T|X,Y) \propto p(Y|X,T)p(T) \tag{3.19}$$

Since there is no conjugate relation traversed between the likelihood and prior, the evaluation of the marginal likelihood constant cannot be detoured, and of course cannot afford simple integration methods. Thus the foregoing MH algorithm is used here to reveal the posterior probability of each Markov state in a tentative fashion whose spirit of guided experimental process will encourage the arrival at the high posterior trees.

3.4.2 Gaussian processes

Before entering the discussion of the Treed Gaussian Processes, it is necessary to expand the details of the Gaussian processes. As having already been addressed briefly in the chapter overview, the Gaussian process is a nonparametric model built on the foundation of a full Bayesian framework.

Hence, the best starting point for examining the GP is the traditional Bayesian linear regression. Since the fundamental element for any Bayes-based inference is the Bayes Rule as described in equation (3.7). The Bayesian linear regression also will belong to the tier of parametric regression model. Therefore the predictive curve must preserve a closed expressive form which contains geometrically interpretable parameters to settle down the type of regression. In the linear regression, the Bayesian analysis begins by defining the type of regression. This naturally yields the following predictive form,

$$f(x) = x^T w \qquad y = f(x) + \epsilon \tag{3.20}$$

where x and y are the input and output vector sets; w is the linear parametric vector set and ϵ is the noise.

It is rather convenient and reasonable to consider the noise to be modelled essentially by a Gaussian distribution whose form accommodates the universal noise.

$$\epsilon \sim N(0, \sigma_n^2) \tag{3.21}$$

Since the regression performance is fully decided by the choice of the w, thus the Bayesian inference assigns a prior over it. For the purpose of convenience in analysis and conservativeness in generality, again a Gaussian distribution will be assigned to it.

$$w \sim N(0, \Sigma_p) \tag{3.22}$$

where the Σ_p is a $p \times p$ covariance matrix.

Both the specifications of the noise and prior are conditioned on a broad sense (a priori according to human experience), logically plausible, but observationally less evidenced. The specification of the likelihood will intensify the strength of reasoning from the Bayesian framework by encoding the observational data. For a linear model, if the prior at each input point is Gaussian distributed, to reflect how likely a certain selected set w could yield a curve that fits the training data space well, the likelihood will also be nominated as Gaussian distributed over the function y with the parametrisation of w. This parametrisation sets the mean of the function response y at its corresponding given location x on the predictive curve $X^T w$ (X is the vector set of all x locations), implying that higher likeliness occurs when the predictive curve could effectively be treated as an averaged interpretation of the training data space. The general mathematical expression goes as,

$$p(y|X, w) = \frac{1}{(2\pi\sigma_n^2)^{n/2}} \exp(-\frac{1}{2\sigma_n^2}|y - X^T w|^2) = N(X^T w, \sigma_n^2 I) \tag{3.23}$$

Both the likelihood and the prior are subject to active definition by the user. The marginal likelihood is a passive factor fully defined by the likelihood and the prior. Not in every circumstance will the marginal likelihood will be attached with a closed form, and in fact in most scenarios, the evaluation of the marginal likelihood can raise substantial nuisances. More details will be addressed in the later content of this chapter. At least for simple Bayesian linear regression, the knowledge of the marginal likelihood is not peremptory. Through correct mathematical operations, the posterior will be presented as,

$$p(w|X, y) \sim N(\bar{w} = \frac{1}{\sigma_n^2} A^{-1} X y, A^{-1}) \tag{3.24}$$

Where $A = \sigma_n^{-2} X X^T + \Sigma_p^{-1}$.

Though the posterior of the w is derived, it is still required to integrate out the dependence on the w to give out the prediction, thus,

$$p(f_*|x_*, X, y) = \int p(f_*|x_*, w) p(w|X, y) dw = N(\frac{1}{\sigma_n^2} x_*^T A^{-1} X y, x_*^T A^{-1} x_*) \tag{3.25}$$

As a parametric regression model, the Bayesian linear regression bears comparatively higher precision (statisically say, precision is not a rigorous term) in producing the linear predictive curve fitting than the more basic and standard least squares method by accounting for other possible choices of w. In addition to providing the predictive fitting, the Bayesian based inference also conveys the benefits of being less absolute at the presence of confidence intervals.

However in the real world, the compliance to pure linearity is scarcely to be seen.

And also in its most commonness, the scenario would either be the absence of knowledge of the exact closed form for regression or the absence of the existence of a closed form. This predicament gives rise to multiple countering solutions, such as high-order polynomial fitting, linear approximate fitting, nonparametric regression etc.

The Bayesian linear regression, although being parametric, its rudimentary basis, the Bayes rule has enough potential to frame an inference scheme for the nonparametric regression. The Bayesian linear regression is parametric, it is because the regression model splits the entire inference processes into two parts. The first part involves a Bayesian inference targeting at the linear regression parameter set w, and to marginalise the prediction conditioned by the w will characterise the second part. This could be comprehensively encapsulated as giving predictions to a data space via a prediction on a parameter set based on the information from the data space. In this sense, it appears like this process suffers a certain degree of redundancy, if the prediction is possible to be directly given based on the data space.

The Gaussian Process perfectly fits in this crack, because the GP essentially is a Bayesian inference directly performed on the function itself whose posterior distribution is entirely and directly conditioned on the training dataset. The term 'function' is used to describe the immanent relations between the input and output among the data space. This relation is not necessarily accessible for mathematical expressions with closed forms. In the GP, the prior will directly be incarnated in the function as a multivariate Gaussian distribution over all the data in the space. It is unlike the Bayesian linear regression whose prior specification is assigned to w.

The prior of the GP goes,

$$f(x) \sim N_p(m(x), k(x, x')) \tag{3.26}$$

where p is the total number of data points for both the training and testing dataset. $m(x)$ is the multivariate mean and $k(x, x')$ is the covariance.

To further explore this specification, it is conducive to conducting a well clarified analysis through separating the training and testing dataset. Then the prior bears the form below,

$$\begin{bmatrix} f \\ f_* \end{bmatrix} = N(\mathbf{0}, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix})$$

Where N stands for the multivariate normal distribution, X is the training input data and X_* is the test input data, $K(.)$ is the covariance matrix.

The GP is a special version of Bayesian inference whose prior specifies not only the initial preference on the output y ($y = [f, f_*]$) at each input entry, but also encodes the mutual correlations among each pair of data points via the presence of a Covariance matrix. The covariance matrix, whose entries are output from a covariance function pre-selected by the user, does command the GP likelihood as it determines the predictive function at the given training data space. From the perspective of mathematical neatness and simplicity, the GP is extremely advantageous for analytical derivation of the posterior, simply because the prior in such a matrix form allows the inference of the posterior through the matrix operations without the knowledge of the expressive form of both likelihood and marginal likelihood (detailed derivation see appendix),

$$\begin{aligned} & f_*|X, y, X_* \sim N(\bar{f}_*, cov(f_*)), \ \ where \\ & \qquad \bar{f}_* \triangleq \mathbb{E}[f_*|X, y, X_*] = K(X_*, X)[K(X,X) + \sigma_n^2 I]^{-1} y, \\ & \qquad cov(f_*) = K(X_*, X_*) - K(X*, X)[K(X,X) + \sigma_n^2 I]^{-1} K(X, X_*). \end{aligned} \tag{3.27}$$

By definition the covariance matrix is constructed to reflect the statistical variation at each input entry as a result of resultant covariance with other data points in the space. Because the influence between points is reciprocal, thus the covariance matrix is completely symmetrical matrix whose elements are all scalar values. Below shows a typical covariance matrix,

$$COV = \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1n} \\ k_{21} & k_{22} & \cdots & k_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ k_{n1} & k_{n2} & \cdots & k_{nn} \end{bmatrix} \tag{3.28}$$

where the k_{ij} as an abbreviated form of $k(x_i, x_j)$, represents the covariance between two points x_i and x_j.

In the covariance matrix above, it could be observed that each row or each column describes the variance interaction between one point (i^{th} or j^{th} if it is the row or column being picked) with other points (including itself) in the dataset.

The parametric model has a fixed pattern to reason the prediction through the human a priori assumption of the predictive form. As a nonparametric model, the GP specifies no fixed form to accommodate the prediction. Its own basic reasoning relies on the generating criterion of the covariance matrix, which is the covariance function as mentioned formerly. In the broad sense, the GP is so powerful for that it could, to some degree, imitate the recognition system of human-beings through specifying the covariance function. When a person is giving predictions to an input entry in a certain dataset, the recognition system instinctively will estimate the prediction roughly through the data points around that entry regardless other distant data points unless there exists obvious signs from the general trend. In this case, specifying a covariance in association with the distance will definitely be highly reasonable and beneficial to encode the logic of inference. This distance based covariance specification is known as the distance covariance function. For the GP, the standard and most commonly used distance covariance functions are the Squared Exponential (SE) covariance function (or alternatively the SE kernel in other name) and the Matérn family of covariance functions. The corresponding expressions are listed below,

$$k_{SE}(r) = \exp(-\frac{r^2}{2l^2}) \tag{3.29}$$

$$k_{Matern}(r) = \frac{2^{1-v}}{\Gamma(v)}(\frac{\sqrt{2v}r}{l})^v K_v(\frac{\sqrt{2v}r}{l}) \tag{3.30}$$

where r is the distance between two mutually influential data points, l is the distance influence weighing parameter, and K_v is the modified Bessel function [47]. All the parameters in the covariance functions are known as the hyper-parameters.

Both the SE and Matérn class kernels are designed to convert the idea of short-distance reasoning into mathematical relations. Therefore, they both share the similarity of assigning lower influential factor to more distant points from the under-influenced point. The SE kernel is considered to be more common than the Matérn class in terms of the usage. The SE kernel is more academically desirable, because its configuration is rather simple and easy to adjust and track its properties. In addition, the SE covariance function is infinitely differentiable that its presentation in a form of predictive curve will be perfectly smooth. The Matérn class adds more versatilities in the smoothness of the final curve fitting, and essentially it is considered to be more realistically representative for real world scenarios where an extremely smooth curve might not be the greatest idea. In the current thesis, all the simulations with either the TGP or GP will be based on the use of the SE kernel, because computational-wisely the SE kernel is more parsimonious. To set scope on the Matérn kernel, as well as other kernels, will be an object in the future work.

The GP is also versatile as the covariance function does not have to be distance covariances. For example, the GP also allows the covariance function to be set in relation to the axial coordinates as to plug in a fixed curve form for the regression fitting. Hence the GP could also be applied as an alternative approach to conduct linear regression etc. From a more general angle, the covariance function could be specified in some way to conduct non-stationary regression as well. However, the analysis on such GP will be accompanied with various mathematical difficulties.

Looking back to the inference of the posterior of the GP, it could be perceived that just the same as any Bayesian inference, the final stage of the prediction is to select the most suitable prediction from the posterior. For the Bayesian linear regression, it is extremely simple, as firstly the linear curve parameter w has been integrated out leaving a posterior predictive distribution that accounts for all possible predictions. Thus to select the best fitting is just the selection of the fitting corresponding to the highest probability from the posterior predictive distribution. Even if the w cannot be integrated out simply, a reasonable selection could still be made under the Maximum A Posteriori (MAP) criterion which selects the prediction by evaluating the highest probability of the posterior. In the GP, the analytical posterior predictive is conceptually the same as its counterpart in Bayesian linear regression, and it is commonly addressed with the name 'Gaussian Process Marginal Likelihood (GPML)'. The GPML describes the likelihood of prediction with the accounts for all possible predictions as weighed by their corresponding probabilities. Since given the training data and the covariance function, the GPML is a measure of the reliability of the prediction w.r.t a set of pre-selected hyperparameters. The Maximum Likelihood (ML) criterion selects the predictions parametrised by the hyperparameter values which maximise the GPML as the appropriate interpretation of the data space. The

ML analysis suffers from the exasperated analytical and computational difficulties compared to the MAP of the Bayesian linear regression due to the complexity of the GPML function where its function profile as related to the hyperparameters is vacant to the probe of direct differentiation analysis.

In fact, the statement of the problem is rather simple and clear that given a function, the objective is to find its global extremity. This particular type of problem has a long cast in the history of mathematics. Its internal concept is rather coherent to any simple problems such as finding the extremity of a parabola. However its external expression varies and is much more complicated and intricate. In the GP, by the ML criterion, the objective function is the logarithmic GPML,

$$\log p(y|X,\theta) = -\frac{1}{2}y^T K_y^{-1} y - \frac{1}{2}\log|K_y| - \frac{n}{2}\log 2\pi, \tag{3.31}$$

where all the hyper-parameters are contained in the covariance K_y.

Given a new data space, the actual graphical profile of the equation above is mostly unavailable unless the data space is well organised, expressing an extremely obvious behaviour. To search for its maximum, there are a number of difficulties to encounter. First at different selections of the covariance function, the number and the type of the hyper-parameters could be radically different. Thus this gives rise to the difficulties in constructing general analytical models for the optimisation. The next problem is that the presence of the K_y also implies the whole dataset will act as a dynamic influential factor imposed on the equation, thus leading to stacked complexity in operating matrices, especially matrix inversions. The impediments in the computational and operational cost are already very demanding to the mathematical manipulations as well as computational efficiencies, but the problem of multiple local extrema is even worse for trapping the optimisation away from the global extremity. Both the academies of algebraic and numerical mathematics do offer solutions to carry out this optimisation, and they will be explained in depth in the later sections of this chapter.

3.4.3 Gramacy's TGPs

General overview

The BCART model is considered to be the first successful and mature attempt to wire the idea of decision trees with the Bayesian learning for the purpose of statistical classification and regression. The Treed Gaussian Processes introduced by Robert Gramacy is built on the raw idea of Chipman's CART, but its theory and application have its own sense of origin.

Before the full course of detailed explanations of the Gramacy's TGPs (GTGP), it is worthwhile to address its internal relation with the BCART in terms of genealogy and variation. As having been explained in the BCART, the amalgamation of the binary decision tree and the Bayesian based inference will lead to the algorithm being compartmented into two parts: the alteration of the tree structure and the corresponding inference under that structure. This character in general will hold firmly for all types of treed Bayesian based inferences. The GTGP complies to this character with a certain degree of variation. In the first place the GTGP has borrowed the whole fraction of the tree structure alteration from the BCART, and made its adaptation on it. This hereditary sense of borrowing remains as the closest similarity between the BCART and the GTGP. For the inference part, the difference is considerable and drastic due to the totally unique specifications of the prior. Because of this difference, the GTGP allows a dual-sampling system in which both the tree structure and the parameter space could be altogether included in the same stochastic sampling space where a Gibbs sampling is used for exploration. For the purpose of a better compare and contrast, recalling that in the BCART, at each sampled tree structure, the hyper-parameters will be optimised to find the MAP of that particular tree structure. However, in the GTGP, the parameter space, instead of being explored separately on a leaf basis, are sampled along with the sampling of the tree structure in the joint stochastic space, where the high joint posterior of all sampling participants is what requires the significant attention. Conceptually, the BCART could be properly considered as a statical MAP optimisation process of the trees, whose sampling space is effectively a local sampling of the tree structure, whereas the GTGP is more dynamic, whose sampling space has rather a global sense across over various different quantities.

Hierarchical prior

The foundational distinction between the BCART and GTGP comes from the specification of the priors. In fact, it is not quite an equitable comparison between BCART and GTGP on this ground, because BCART is used for constant function parametric regression, while the GTGP is designed for nonparametric regression. The GTGP of course will surely be superiorly more complicated. The GTGP specifies the prior in a hierarchical structure.

$$\begin{array}{c}
Z_v|\beta_v,\sigma_v^2,\boldsymbol{K}_v \sim N_{n_v}(\boldsymbol{F}_v\beta_v,\sigma_v^2\boldsymbol{K}_v)\\
\hdashline
\beta_v|\sigma_v^2,\tau_v^2,\boldsymbol{W},\beta_0 \sim N_{m_X}(\beta_0,\sigma_v^2\tau_v^2\boldsymbol{W})\\
\sigma_v^2 \sim IG(\alpha_\sigma/2,q_\sigma/2)\\
\hdashline
\beta_0 \sim N_{m_X}(\boldsymbol{\mu},\boldsymbol{B})\\
\tau_v^2 \sim IG(\alpha_\tau/2,q_\tau/2)\\
\boldsymbol{W}^{-1} \sim W((\rho\boldsymbol{V})^{-1},\rho)
\end{array} \tag{3.32}$$

where any subscript v denotes the vth leaf.
Z_v is the function value set (the response variable of the function, or training output)
$\boldsymbol{F}_v$ is the design matrix in conventional linear regression model
β_v is the linear regression characteristic parameter set
$\boldsymbol{K}_v$ is the covariance matrix
σ_v is the function variance
τ_v is the function variance for β
$\boldsymbol{W}$ is the covariance matrix for β
ρ and V are the parameters of Wishart distribution

The hierarchical framework is comprised of three layers, on the foundational layer lies the direct prior specification on the function itself, which is of the tradition to the Gaussian Process, where the prior is imposed directly on the function itself. The parameters of this particular association is known as the hyperparameters. The second layer is an expansion in the sense of the coverage of the prior belief. Basically the parameters in the base layer are allowed for sampling as well. Thus rather than making speculations on the Z directly, the sampling of its parametrization intends to shift the greater uncertainty in guessing Z into a guessing with comparatively lower uncertainty. The parameters parametrising the hyperparameters are recognised as hyper-hyperparameters. Then the last layer is where the hyper-hyperparameters are defined as probability distributions. The distributions of the hyper-hyperparameters are treated known in completeness.

By analysing this hierarchical prior structure, from the first layer, it could be apparently seen that the algorithm holds the initial belief of a preferred compliance to a linear manner for the fitting curve. This presumed compliance serves a pragmatic industrial standard where simplicity is always of the priority at a reasonable concession of the accuracy. And this deliberate setting of a linear mean has a more deliberate purpose for the introduction of the limited linear model by Gramacy, where the algorithm automatically selects whether to perform a traditional GP fitting or a linear fitting. In all the parametrisations, the parameters could be primarily classified into two groups:

1. Local parameters: $\theta_v = [\beta_v, \sigma_v^2, \boldsymbol{K}_v, \tau_v^2]$ which is entirely associated with the local region (or leaf). The first three parameters in the set are also known as the GP parameters as in the parametrisation of the GP prior.

2. Global parameters: $\theta_0 = [\beta_0, \boldsymbol{W}]$ which has the property to influence the sampling in each leaf across the entire data span. These two parameters plus the last one in the local parameters are also known as hierarchical parameters.

The presence of fully defined distributions for the hyper-hyperparameters is necessary considering the sampling of the hyperparameter β_v. Albeit the posterior analysis could exist without the specifications in the third layer of the hierarchical priors. To assign probability distributions to the hyper-hyperparameters has its own significance and considerations as designated at the performance of the prediction by the algorithm. It will be expounded later in this section.

The foregoing context addressed the compare and contrast between the BCART and the GTGP. The most significant advantage that the GTGP carries is the dual-sampling scheme where the tree structure and the parameters are jointly sampled in the same sampling space. Now denote the tree structure as T and the total volume of parametrisations as $\theta = [\theta_v, \theta_0]$, applying basic Bayesian analysis on T and θ individually, the following relations arrive,

$$\begin{gathered} p(T|\theta) \propto p(\boldsymbol{K}_v|T)p(T) \\ p(\theta|T) \propto p(T|\theta)p(\theta) \end{gathered} \tag{3.33}$$

The best performance of the fitting combined with partitions by the GTGP is decided by the Maximum A Posteriori criterion. The posterior space macroscopically is bipartite of T and θ, thus the joint posterior $p(T, \theta)$ is where the MAP criterion is imposed. As conditioned that mutual conditional posteriors are analytically accessible as shown above, a Gibbs sampling strategy could be utilised to simulate the draws from the joint sampling space by sampling iteratively from the conditional posterior distributions $p(T|\theta)$ and $p(\theta|T)$.

On the part of sampling the posterior of the tree structure T as conditioned on the θ, the prior is defined in a way same as in the BCART model as equation (3.8).

For its likelihood part, the fact as a conditional PDF on the T allows an equivalent view of the T likelihood as the quality of prediction under that tree across all its leaves. To reflect such quality, one could use the classic GP posterior as equation (3.27) for MAP evaluation, or alternatively the *log* GPML equation (3.31) for ML evaluation. In terms of the GP performance evaluation, the *log* GPML (whether should be logarithmic depends on whether all other derivations are conducted in the logarithmic environment) weighs more preference over the direct GP posterior for being less bias by taking into account all possible predictions. Hereby because the covariance $\boldsymbol{K}_v$, as part of the GPML, is the predominant factor to its output. The GP from its traditional sense is a process, where given the data the covariance is well set and given, to produce a fitting on the data according to the setting of covariance (more specifically the setting of the covariance hyperparameters). The backward search for the optimal covariance setting leads to the optimsation of the GPML. In the GTGP, since the sampling space wires everything together, and everything if whose joint distribution with others is not mathematically available in closed form or whose closed form of possible existence but is impossible to sample directly from, it should be able to fit into a Bayesian inference framework for deriving the conditional PDF condistioned on each other, thereby the Gibbs sampling strategy could simulate the joint sampling of the random space. For the fullness of establishing the Gibbs sampling and arriving at conditional distributions in closed form for all the parameters, here the GPML has to pair with the prior of $\boldsymbol{K}_v$ to make the $p(\boldsymbol{K}_v|T)$ effectively the posterior of the $\boldsymbol{K}_v$ (one should distinguish between the posterior of $\boldsymbol{K}_v$ and the posterior of the GP).

For the sampling of the conditional posterior for parameters, the multi-dimensional posterior sampling space is entangled in sufficient complexity to impede simple sampling strategies. Again, the Gibbs sampling scheme is utilised to break down the complexity into multiple simple localised uni-variant sampling with mutual conditions. Through a series of rigorous mathematical operations, the conditional posteriors for all the parameters could be obtained as below:

For the β_v,

$$\begin{gathered}\beta_v|rest \sim N(\tilde{\beta_v}, \sigma_v^2\boldsymbol{V}_{\tilde{\beta_v}})\\ \boldsymbol{V}_{\tilde{\beta_v}} = (\boldsymbol{F}_v^T\boldsymbol{K}_v^{-1}\boldsymbol{F}_v^T + \boldsymbol{W}^{-1}/\tau_v^2)^{-1},\\ \tilde{\beta_v} = \boldsymbol{V}_{\tilde{\beta_v}}(\boldsymbol{F}_v^T\boldsymbol{K}_v^{-1}\boldsymbol{Z}_v^T + \boldsymbol{W}^{-1}\beta_0/\tau_v^2)\end{gathered} \tag{3.34}$$

Similarly β_0,

$$\begin{gathered}\beta_0|rest \sim N(\tilde{\beta_0}, \boldsymbol{V}_{\tilde{\beta_0}});\\ \boldsymbol{V}_{\tilde{\beta_0}} = (\boldsymbol{B}^{-1} + \boldsymbol{W}^{-1}\sum_{i=0}^{r}(\sigma_v\tau_v)^{-2})^{-1},\\ \tilde{\beta_0} = \boldsymbol{V}_{\tilde{\beta_0}}(\boldsymbol{B}^{-1}\mu + \boldsymbol{W}^{-1}\sum_{i=1}^{r}\beta_v(\sigma_v\tau_v)^{-2}).\end{gathered} \tag{3.35}$$

Then τ,

$$\begin{aligned} \tau_v^2 | rest &\sim IG((\alpha_\tau + m)/2, (q_\tau + b_v)/2) \\ b_v &= (\beta_v - \beta_0)^T \boldsymbol{W}^{-1} (\beta_v - \beta_0)/\sigma_v^2. \end{aligned} \tag{3.36}$$

And Wishart covariance matrix,

$$\begin{aligned} \boldsymbol{W}^{-1} | rest &\sim W(\rho \boldsymbol{V} + \boldsymbol{V}_{\tilde{T}}, \rho + r) \\ \boldsymbol{V}_{\tilde{T}} &= \sum_{i=1}^{r} \frac{1}{(\sigma_v \tau_v)^2} (\beta_v - \beta_0)(\beta_v - \beta_0)^T. \end{aligned} \tag{3.37}$$

And σ_v^2,

$$\sigma_v^2 | d_v, g, \beta_0, \boldsymbol{W} \sim IG((\alpha_\sigma + n_v)/2, (q_\sigma + \Phi_v)/2) \tag{3.38}$$

At last the covariance $\boldsymbol{K}_v$ is obtained,

$$\begin{aligned} & p(\boldsymbol{K}_v | t_v, \beta_0, \boldsymbol{W}, \tau_v^2) = \\ & \quad (\frac{|\boldsymbol{V}_{\tilde{\beta}_v}|}{(2\pi)^{n_v} |\boldsymbol{K}_v| |\boldsymbol{W}| \tau_v^{2m}})^{1/2} \frac{(q_\sigma/2)^{\alpha_\sigma/2}}{[(q_\sigma + \Phi_v)/2]^{(\alpha_\sigma + n_v)/2}} \frac{\Gamma[(\alpha_\sigma + n_v)/2]}{\Gamma[\alpha_\sigma/2]} p(\boldsymbol{K}_v) \\ & \quad \Phi_v = \boldsymbol{Z}_v^T \boldsymbol{K}_v^{-1} \boldsymbol{Z}_v + \beta_0^T \boldsymbol{W}^{-1} \beta_0 / \tau^2 - \tilde{\beta}_v^T \boldsymbol{V}_{\tilde{\beta}_v}^{-1} \tilde{\beta}_v. \end{aligned} \tag{3.39}$$

As seen from the equations above, it is observable that, excluding the $\boldsymbol{K}_v$ and σ_v^2, all the other parameters do hold the full conditions. Because for the purpose of efficient sampling, it is possible to achieve further marginalisation of parameters through mathematical manipulations for $\boldsymbol{K}_v$ and σ_v^2. Excluding $\boldsymbol{K}_v$, all the other parameters are available for sampling from integrated samplers provided by most of the programming softwares. The $\boldsymbol{K}_v$ is beyond any common means of sampling in a sense of direct availability, since its closed form is too much analytically complicated.

Analysis on the $\boldsymbol{K}_v$

In the expression, there includes a prior term for $\boldsymbol{K}_v$ which has not been addressed above. Because the covariance matrix is completely parametrised by the hyperparameters in the covariance function acting on a fixed dataset, to specify the prior for $\boldsymbol{K}_v$ is identical to specify the prior for the covariance hyperparameters. The GTGP uses a classic SE kernel to explain the hidden relations in the data space. With the infliction of the noise, the SE kernel will in addition contain a hyperparameter g (alternatively known as nugget) representing the noise variance. Thus,

$$K_{SE}(i,j) = -\frac{(x_i - x_j)^2}{2d} + g\delta_{ij} \tag{3.40}$$

where d is the distance hyperparameter and g is the noise variance hyperparameter; δ_{ij} is the delta function whose value is 1 at $i = j$ and elsewhere $\delta_{ij} = 0$.

For these two covariance hyperparameters, there are various ways to define their probability distributions. The definitional domain for both d and g stretches from 0 to $+\infty$, and drastically different choices of the d or g yields different interpretations of the data space. Therefore the scale of the values for the covariance hyperparameters does not necessarily imply the quality of the prediction out from this particular hyperparameter setting. For example, a small choice for the value of p results in a prediction that tends to go through all the data points, while the large value choice explains the whole data volume as noise and generates a neutral predictive curve passing through the middle of the entire dataset bed. Therefore the user needs to pay extra attentions to the preference specified on the prior distributions particularly of d and prudently of g. From a conventional perspective towards the prior specification of g, a meaningful, pragmatic and informative fitting requires in general that the data space should be presumably adequately self-evidenced that the prediction produced could be considered as a decent interpretation of the entangling relations hidden in the data rather than roughly consider everything results from an event of noise. Thus it would be reasonable to assign a exponential distribution to the g,

$$g \sim EXP(-p) \qquad p > 0 \tag{3.41}$$

This prior specification for the g has a marginal limit of 1 towards the left end of the g on the coordinate ($g \to 0$), which prevents extremely unrealistic overfitting; and towards the right end ($+\infty$) the prior assigns infinitely low probability to discredit the heavily noisy model.

The setting on the d prior can raise potential arguments, because priorly plainly given the dataset, it still remains obscure which approximate choice of the d will be appropriate to interpret the dataset well without incurring overfitting. And more perplexing is that the setting of d does affect the influences of the g to the prediction, despite it does not necessarily oblige that d and g are correlated. The choice of d should avoid being too small. Because any data acquisition process, though could be recorded in a continuous fashion (e.g. record vinyl CDs), is always computationally processed in a certain level of precision which is absolute discrete, in this condition extremely small d will eliminate the presence of realistic noise and mathematically causing matrix singularity as well. Normally the d hyperparameter is specified in a way of Gamma distribution or a mixture of multiple Gamma distributions. Because Gamma distribution has 0 probability for very small d and also 0 value for very large d, plus that the mixture of Gammas allows multiple fitting curve characters to be encoded in the prior. For example the dual-mixture with some setting can consequence a wavy or smooth fitting out from the GP. The prior for d is given like,

$$d \sim \frac{1}{n}\sum_{i=1}^{n}[Gamma(\alpha_n, \beta_n)] \tag{3.42}$$

The joint prior distribution for d and g is naturally Bayesian related to the corresponding individual distributions of d or g,

$$p(d, g) = p(d|g)p(g) = p(g|d)p(d) \tag{3.43}$$

For the purpose of simplicity and efficiency, it is technically sensible and practically desirable to assume d and g are independent for their own prior specifications, so to make the joint prior above reduces to,

$$p(d, g) = p(d)p(g) \tag{3.44}$$

The coalescence of the prior $\boldsymbol{K}_v$ with the tedious mathematical expressions at front, resorted the thinking on iterative sampling methods whose performance is at great reliance on the artificial computational power. Here again, the Metroplis-Hastings method is introduced for sampling from analytically unidentifiable probability distributions. To establish full Gibbs sampling for the holistic parameter space, theoretically it is required for the MH sampling of the $p(\boldsymbol{K}_v| \sim)$ to reach the statistically stable state through enough iterations. Here one cheat could be trumped as is allowed based on the true purpose of sampling of the parameter space. Though the Gibbs sampling here is commissioned to simulate the joint sampling of all participants in the sampling space, the ultimate goal that brings its meaning is to search (or reach in another sense) for the highest probable tree in the Markov space. Therefore it means any method that could accelerate this searching process should be taken into consideration regardless it potentially could breach the strict Gibbs sampling. Because in the MH sampling, especially when it is done in a logarithmic scale, the large contrast between the proposed and current state will almost ensure any accepted state will be probabilistically superior. Meanwhile the other parameters are not at the most optimal values, the iterative completeness in the MH process for the $p(\boldsymbol{K}_v| \sim)$ will not mean too much as compared with just one iteration. Overall the single step MH sampling for $p(\boldsymbol{K}_v| \sim)$ will generally be sufficient to make global convergence toward the high posterior region in the Markov space, plus not mention that by doing so could substantially reduce the wasteful computations.

Summarising the GTGP

Then look back to the three-layer hierarchical prior structure, by associating it with the set of conditional posteriors, the necessity of the priors for the hyper-hyperparameters could be revealed. They actually act as grease in the sampling parameter space to glue the tree structure information with the individual parameter sampling in each leaf.

In general the process of the GTGP could be summarised as the follows:

- Propose a new treed structure by randomly choosing one from the four executions:*Grow*, *Prune*, *Change*, *Rotate*;
- Based on this proposition, make single-iteration MH sampling for all the $\boldsymbol{K}_v$;
- sum up all the logarithm $\boldsymbol{K}_v$ and add the tree prior (3.8) to obtain the proposed global logarithm posterior;
- compare the proposed and current global logarithm posterior using the MH criterion;
- make decisions to the acceptance of the proposition and record necessary information tagged as current;
- Make draws for the global parameters based on their conditional posterior distribution;
- Make draws for all the leaf parameters except $\boldsymbol{K}_v$ from their corresponding conditional posterior distributions;
- return to step one.

The performance of the GTGP will be discussed in details with benchmarking against the Chipman based TGP in the later discussion chapters.

3.4.4 Chipman-based TGP

The central concern of this thesis is more toward the side of Chipman-based TGP (CTGP) than anything else. The Chipman-based TGP suggests a TGP model completely springs from the raw ideas of Chipman's BCART model (Chipman himself hasn't done any extended work for TGP model, his work stopped at Bayesian CART). The GTGP is by any means a true innovative model out from the BCART, but this is not the main focus. The CTGP compared with the GTGP is more orthodox in the respect of pedigree to BCART. Chipman et.al. are not the pioneers of developing any TGP models. Therefore Gramacy is indubitably recognised as the founder of the TGP. However the GTGP made massive modifications on the statistical mechanism inside, thus to some extent, deviates itself from the BCART conceptually from its idiosyncratic full stochastic properties. The CTGP discussed here is a true semi-stochastic model resembling the BCART. Because each proposed tree will strictly be subject to an optimisation process to evaluate the true MAP of the tree, while the dual-sampling scheme in the GTGP allows everything being sampled in an ensemble fashion, hence that the MAP evaluated at each step of the GTGP is conditioned on transient parameter values which are performing their own random walking in the Markov space in a guided fashion, therefore such conditional MAP is iteratively true but holistically uncertain.

The basics of the CTGP is extremely straightforward in conveying the spirit of the BCART. As the same as the GTGP, the CTGP also inherits the entire sampling strategy for the tree structure by using the 4 tree structure alteration executions in conjunction with the MH algorithm. In details, the CTGP borrows the complete scheme from GTGP which differs from the BCART by replacing the *Swap* in the BCART with the *Rotate*. This has not been addressed in the content above. The *Rotate* execution as introduced by Gramacy is an amelioration for the *Swap* in the BCART. The *Swap* occasionally generates leaves with zero elements, while the *Rotate* only changes the order of placing the existent partitions in the data space so that no damage is made to the leaf elements. In the CTGP, there is no scheme such like dual-sampling scheme to weave a complete stochasticity, instead the MAP evaluation principle is a mixture of simplicity and inflexibility that each proposed tree structure is subject to a GP hyperparameter optimisation process as to find the exact ML (or MAP if hyperparameter priors are assumed) of that tree. This optimisation process could be achieved in either stochastic or nonstochastic way, but its spirit of deterministicity is inexorable. Whatever approach to take does comply with the general inference pattern:

$$\begin{aligned} p(T|X,Y) &\propto p(Y|X,T)p(T) \\ p(Y|X,T) &= \prod_{i=1}^{b} p(Y_i|X_i) \end{aligned} \tag{3.45}$$

Where T is the tree, (X, Y) are the training input and output, and b is the number of leaves in the tree.

From the equations above, it could be seen that the CTGP breaks down the overall

GP of the entire data space into b mutually isolated sub-GPs. Each of these sub-GPs does uphold the traditions of the classical GP. The second equation above tells the likelihood of the tree $p(Y|X,T)$ is effectively the total product of all the GPMLs from all individual leaves. The track naturally goes to the optimisation of these GPMLs. To apply either of the two general aforementioned optimisation methods to the GPML function equation (3.31) yields two different types of CTGP, namely: the CTGP with stochastically based optimisation and the CTGP with numerically based optimisation.

The comparison between both CTGPs yields no obvious inclination towards either. The numerically based CTGP apparently possesses the advantages of fast convergence rate in search of the maxima of the GPML, where the entire searching process is schematically tractable. The defective aspect of this type of optimisation results from its incapability of dealing with the complexity characterised by multiple local extrema (here is the maxima for GPML). Because the optimisation of the GPML belongs to the category of non-convex optimisation where the local minimum is not necessarily the equivalence to the global minimum. As the numerically based optimisation is mostly gradient-based, whose searching mechanism relies on detecting the geographical information (gradient and derivative of gradient) of the function at a given point, that thereby a reasonable approaching to the target point could be made accordingly. If multiple extrema do exist, it is nearly impossible for the search to escape from the effective region of a non-global local minimum once the search is trapped there. On the same ground, the stochastically based optimisation offers a different solution to counter the multiple maxima of the GPML through randomly visiting function coordinates according to certain specified criteria. The fact of random visits suffers much less constraints in its iterative update of the searching point than as being a numerical optimisation tracing the gradient information which makes its update rather deterministically confined in the effective region of the local minima. However, trade-offs on the computational cost have to be compromised for more uncertainties in the searching process as well as the larger searching span possibly reachable through the random walk. The following two sections expounds the detailed insights on both optimisation methods.

Stochastically-based Optimisation

The GPML at equation (3.31) obtains its mathematical form based on a multi-dimensional parametrisation whose actual dimensionality attributes to the selected GP kernel. At the choice of the standard SE kernel as equation (3.29), the optimisation will deal with 3 dimensions $\theta = [\sigma_f^2, l, \sigma_n^2]$, namely [function variance, influential distance, noise variance].

$$k(x_p, x_q) = \sigma_f^2 \exp(-\frac{(x_p - x_q)^2}{2l^2}) + \sigma_n^2 \delta_{pq} \tag{3.46}$$

The σ_f^2 and σ_n^2 are not the inherent parameters in the classical SE kernel, because either of them does not encode anything into the covariant relation in the data space. However they are indispensable for the outcome of the inference. The function variance σ_f^2 is a tuning mass to the covariant intensity as implied by the mathematical expression. Its graphical interpretation in the data space is manifested by the vertical offset of the predictive curve to the data cluster. Thus it is more like a calibration factor for the inference. The noise variance σ_n^2 only models the local variance as a result of noise in complete segregation against other information in the data space. The inclusion of the noise variance is twofold essential. In the first place, noise is ubiquitously objectively existent, thus to model noise is natural and indubitable. On the second part, the presence of σ_f^2 in the matrix computation act as a lumped uncertain mass in each diagonal entry of the covariance matrix to prevent the covariance matrix from being singular. Therefore the covariance will be positively definite.

In this three dimensional function space, a multidimensional modified uniform sampling strategy is employed to simulate the sampling from the marginal likelihood. The name suggests a modification of the traditional uniform sampling. What is not superimposed with the traditional uniform sampling here is something could term as 'arbitrary window'. The arbitrary window basically means the uniform sampling will be implemented within a certain bounded range whose size is essentially alterable. There are chiefly three window sizes to account for three types of searching, namely: refined window, normal window and large window. The specific size for each depends on the given training data, as the complexity of the data space proportionally influences the complexity of the optimisation.

There are quite a number of selections of sampling strategies could be applied at this particular instance. Sampling strategies like multidimensional Metropolis-Hastings algorithm should also suffice the requirements. The reasons for the preference goes toward modified uniform sampling hold a sense of idiosyncrasy specially dedicated to the particularity characterised by the Treed model. Uniform sampling, as a means of optimisation or stochastic simulation for unknown distributions, is criticised heavily for being highly inefficient, especially in dealing with high dimensionality. Because in those cases requiring high dimensional simulations, according to mathematical analysis, the number of samples demanded to hit the target proximity (e.g. high GPML region) could be egregiously large, which practically put the method into severe discouragement. However this inherent deficiency of uniform sampling is not generalisable in the treed model. Because one of the essential benefits from the treed model is dissembling the complexity of the input space through partitioning the data into multiple regions where the behaviour of the data could be modelled in a much simpler fashion. Premised in this, it could be assumed with fine reason that of each partitioned leaf, the GPML function surface in terms of its hyperparameters would not suffer too much from badly skewed and distorted profile. It could also be further assumed the profile of the function at each dimension preserves a decent degree of parabolic property where the curvature does not change drastically and abruptly.

In general, using a modified uniform sampling at this case gains two advantageous points over the MH algorithm:

First for pragmatic reasons, given an unknown function in many occasions, there is no obvious sign (eg. monotonousness) to indicate whereabout of the global extremity in a domain protracting to both ends of infinity. What is most likely known by the user is a bounded broad range within which there exists with high chance a plausible global extremity whose global extremeness is not absolute but practically acceptable. In the GP, this kind of scenario is of a common consideration. To exhaustively explore the infinitely large hyperparameter domain of the GPML seems probably inefficient and practically impossible. Thus the best expedient plan is to constrain the search within a certain set of boundaries.

The second merit is that if even though a fully unbounded continuous proposition PDF is applied to convey the MH sampling, for example a simple multi-variant normal distribution, to set the parameters for such PDF can still lead to arguments. Because the parameter setting for the proposition PDF determines the pace of the search. However to decide this pace is not as easy as it seems to be, because the correlative relation among the hyperparameter space is not explicit, plus the matching of the pace with the terrain of the function's profile is also problematic to conduct. In this sense, simple bounded uniform sampling can heavily reduce such complexity.

Numerically-based Optimisation

Numerically-based optimisations generally use line search methods (LSM) to find the target extremity of the function. When it comes to the line search method, the selection of choices is quite diverse. There appear to be three commonly-used methods to carry out such a task. As an LSM, there is no extrication from the use of knowledge about the gradient of the target function, as the gradient of function generally characterises the function extrema. In many LSMs, the matrix of the 2^{nd} derivative of the function (Hessian matrix) is also required for computation. The knowledge of the Hessian forms the main differentiating feature while distinguishing among the three methods. Before introducing any of the three methods, it is worthwhile to advance the general line search criterion first. Briefly a line search method is to approach one unknown extremity of the function in multiple iterative steps by visiting and evaluating multiple locations of the function. The general iterative criterion goes as,

$$x_{k+1} = x_k + t_k P_k \tag{3.47}$$

where t_k is the search step at the k^{th} iteration, P_k is the search direction at the k^{th} iteration. The update of the search location x from the equation above envisages an approaching towards the target extrema. The choices of the t and P characterise different line search methods.

The first applicable line search method is pervasively studied and used for optimisation problems, and is prominently known as Newton's method. The theoretical foundation of Newton's method is to collect the Jacobian (gradient vector) and Hessian matrix information through differentiating the target function, then use this information to guide the search of the extremity. The idea springs from setting the derivative of the Taylor expansion to zero, where the general form of Newton's method can be obtained. Such form is shown below for the kth iteration:

$$x_{k+1} = x_k + H_k^{-1} \triangledown f(x_k) \tag{3.48}$$

where H is the Hessian matrix and $\triangledown f(x_k)$ is the gradient vector of f at x_k.

The major advantage signified by Newton's method is its comparatively fast convergence rate. Because Newton's method has access to the closed form of the Hessian matrix, it means the Newton's method is internally wired with the mechanism to adjust the search step size automatically based on the gradient and the change rate of the gradient; thus if convergence is ensured, it should occur with less iterations. However the downside is that Newton's method naturally demands analytically differentiating the target function twice, which for some rather complex functions is hard to do, and also for some functions the Hessian computation in terms of computational expenditure is rather burdensome, for the inverse of Hessian matrix has to be performed for each iteration. Thus, Newton's method is advantageously parsimonious of iterations, but it does not necessarily mean it is overall computationally faster. Another problem associated with the Newton's method is the attraction of saddle points, which has been addressed by many scholars in numerous articles. In high-dimensional optimisation problems, a saddle point is a location on the function profile where there is both a maximum and a minimum jointly with respect to each dimension of the function. In a two dimensional space, the saddle point imitates a saddle (figure 3.3) which makes the coinage of the name. The problem with saddle point is a subject commonly to be encountered in high dimension non-convex optimisations [48]. The basic Newton's method has no riposte to a saddle point, but the evaluation of Hessian could act as an indicator of the wrongful approach toward a saddle point, as mathematically the determinant of the Hessian at a saddle point is negative.

The gradient and Hessian evaluation shows the Gaussian Process Marginal Likelihood (GPML) has no exemption from the nuisance of the presence of saddle points. To what extent the GPML suffers from this requires further study. Aside from this, to apply Newton's method to the GPML also suffers the pain from the tedious derivation of the Hessian matrix. Thanks to Mardia and Marshall [49], based on their derivations, one could arrive at the following key quantities by denoting the logarithmic GPML as L,

[2] Source: https://en.wikipedia.org/wiki/Saddle_point

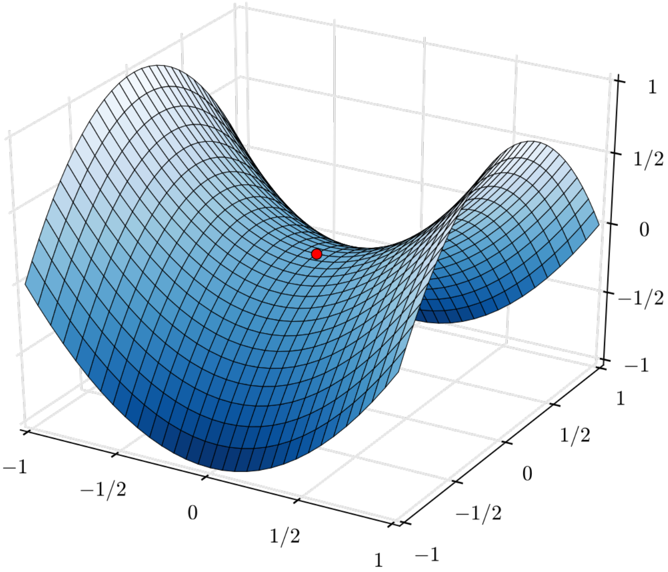


Figure 3.3: ^{2}Saddle point in two dimensional space.

$$\nabla_i L(\theta) = -\frac{1}{2} tr(K^{-1}\frac{\partial K}{\partial \theta_i}) - \frac{1}{2} Y' \frac{\partial K^{-1}}{\partial \theta_i} Y \tag{3.49}$$

where ∇_i is the gradient on the i^{th} dimension in a v-dimensional hyperparameter space, $theta_i$ is the i^{th} hyperparameter from the covariance function defining each entry of the K

For finding the inverse derivative one can use the following relation,

$$\frac{\partial K^{-1}}{\partial \theta_i} = \frac{\partial K^{-1}}{\partial \theta_i}\frac{dK}{dK} = -K^{-1}\frac{\partial K}{\partial \theta_i}K^{-1} \tag{3.50}$$

The closed form of the GPML Hessian is as follows,

$$H = \begin{bmatrix} \nabla_1\nabla_1 L & \nabla_1\nabla_2 L & \cdots & \nabla_1\nabla_v L \\ \nabla_2\nabla_1 L & \nabla_2\nabla_2 L & \cdots & \nabla_2\nabla_v L \\ \vdots & \vdots & \ddots & \vdots \\ \nabla_v\nabla_1 L & \nabla_v\nabla_2 L & \cdots & \nabla_v\nabla_v L \end{bmatrix} \tag{3.51}$$

where $\nabla_i\nabla_j L = \frac{\partial^2 L}{\partial\theta_i\partial\theta_j}$

Then,

$$\nabla_i \nabla_j L = -\frac{1}{2}\{tr(K^{-1}\frac{\partial^2 K}{\partial\theta_i\partial\theta_j} + \frac{\partial K^{-1}}{\partial\theta_i}\frac{\partial K}{\partial\theta_j}) + Y'\frac{\partial^2 K^{-1}}{\partial\theta_i\partial\theta_j}Y\} \tag{3.52}$$

Again, there is an associated relation between the K^{-1} based derivative and the K based derivative,

$$\frac{\partial^2 K^{-1}}{\partial\theta_i\partial\theta_j} = K^{-1}(\frac{\partial K}{\partial\theta_i}K^{-1}\frac{\partial K}{\partial\theta_j} + \frac{\partial K}{\partial\theta_j}K^{-1}\frac{\partial K}{\partial\theta_i} - \frac{\partial^2 K}{\partial\theta_i\partial\theta_j})K^{-1} \tag{3.53}$$

The access to the full set of function, gradient and Hessian allows the Newton's method in equation (3.48) to be enforced.

The second viable solution to the optimisation is the Quasi-Newton method (QNM). Generally the QNM is a complementary method to Newton's method. It is used where the Hessian of the function is not explicit or too costly to compute. The QNM still requires a Hessian in the line searching process; however this Hessian is an approximation whose derivation relies not on the analytical analysis of the function but on an iterative update through a certain criterion [50]. The QNM bears heavy resemblance to Newton's method. In equation (3.48), The second term on the RHS, which is the search direction, goes as,

$$H_k h_k = -\nabla f(x_k) \tag{3.54}$$

The Newton's method computes the inverse Hessian to infer the search direction h_k for the k^{th} iteration. The QNM just directly uses an approximate Hessian B_k (strictly say an approximate inverse Hessian) to make:

$$h_k = -B_k \nabla f(x_k) \tag{3.55}$$

The update of B_k forms the key feature of the QNM. In general the search rule of the QNM could be summarised as (for searching the minimum as a custom):

1. Make initial guess for both x_0 and B_0.

2. Compute the gradient $\nabla f(x_k)$ and set the searching direction as $h_k = -B_k \nabla f(x_k)$.

3. Update the x by $x_{k+1} = x_k + t^* h_k$, where $t*$ minimises $f(x_k + th_k)$.

4. Update B by setting $B_{k+1} = B_k + U_k$, where U_k is an updating matrix based on a certain criterion.

The update of the Hessian bears a structure-wise strictness to regularise the approximate Hessian to be absolute positive definite, and of course, symmetric. There are quite a few updating functions could be selected. The most recent trending is at the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update [51],

$$U_k = (1 + \frac{\Delta g_k^T B_k \Delta g_k}{\Delta x_k^T \Delta g_k})\frac{\Delta x_k \Delta x_k^T}{\Delta x_k^T \Delta g_k} - \frac{B_k \Delta g_k \Delta x_k^T + (B_k \Delta g_k \Delta x_k^T)^T}{\Delta x_k^T \Delta g_k} \tag{3.56}$$

where $\Delta g_k = g_{k+1} - g_k$, $\Delta x_k = x_{k+1} - x_k$.

From the updating method, one can also observe that the approximation of the Hessian is based on the information of the function and gradient before and after a line search step.

Apart from the updating matrix, there is another technical issue to solve, which is the search for the t^* value. The t^* value is the t which maximises the $f(x_k + th_k)$. The available knowledge of the closed form of the target function enables a evaluation of the derivative function for finding the t^*. However, there are practical issues; the evaluation of the derivative function is subject to severe mathematical difficulty. The evaluation of the derivative function of the GPML serves as a typical example, where the operations regarding the K (e.g. compute the determinant of K) introduce great complexity to the solution of the t. One could briefly picture that for an Square Exponential (SE) kernel-based GPML, the final expression for the GPML derivative is a mixed entangled combination of exponential and polynomial functions. Because for an $n \times n$ matrix, its corresponding determinant is a $n!$ term polynomial expression where each term is a n-element product of the selected entries from the matrix. Therefore it is nearly impossible to obtain direct solutions to t analytically, whose solutions are encrypted in such complicated fashion. Fortunately, aside from making direct analytical solutions, it is viable to seek an expediency to indirectly accommodate the problem by not directly finding the t^* but, compromisingly, an inexact t^*, notated as t_* which ensures an approaching to the target extrema and also ensures an approach to zero for the gradient. The very method for inexact line search here is known as Wolfe conditions [52]. The principle of Wolfe conditions is rather simple. Principally, there are two conditions constraining the search based on the function and the function gradient respectively.

If a minimum is to be searched (Wolfe conditions normally apply to search for the minimum). The first condition ensures the t_* at least descends the search in the function space for a certain amount:

$$f(x_k + t_k P_k) \leq f(x_k) + c_1 t_k P_k^T \triangledown f(x_k) \tag{3.57}$$

where P_k is the search direction, and c_1 is a constant manually set based on the user's preference.

The second condition guarantees that the gradient will decrease at least for a certain amount after making the step t_*,

$$-P_k^T \triangledown f(x_k + t_k P_k) \leq -c_2 P_k^T \triangledown f(x_k) \tag{3.58}$$

where c_2 is another constant defined by the user; but it should be that $0 < c_1 < c_2 < 1$.

Normally for QNM integrated with Wolfe conditions, Wright and Nocedal [53] give out reference values for $c_1 = 10^{-4}$ and $c_2 = 0.9$. The Wolfe conditions are not exclusive to QNM. They have multiple applications to almost all line search methods. Despite the wide applicability, Wolfe conditions are much disadvantaged for two major internally correlated reasons. One is that using Wolfe conditions generally in another way means increasing the iterative steps to the target minimum. The other is that t is actively chosen to check its validity to the Wolfe conditions, thus a loop for t is also required to update t to the satisfying value. Such an updating process could be extremely unstable. Because at different function locations, the required t could be massively different, which on the trivial side, increases the number of loops required, on the harsh side, drives the loop into endless cycles. This issue will be discussed through multiple case studies in this thesis.

The last numerical optimisation method discussed here is another extensively-used line search method known as the Non-linear conjugate gradient method (NLCGM). There is an obvious derivation of the NLCGM from the Linear Conjugate Gradient Method (LCGM).The heuristic basis of the CGM stems from the ground of what is known as the steepest descent method (SDM). Both the SDM and the LCGM are designed for quadratic optimisation. The NLCGM is generalised for a broader application. To serve in any GP related fields, mainly both the LCGM and NLCGM have roles to fulfil. Their applications are aiming differently, but are affinitive to each

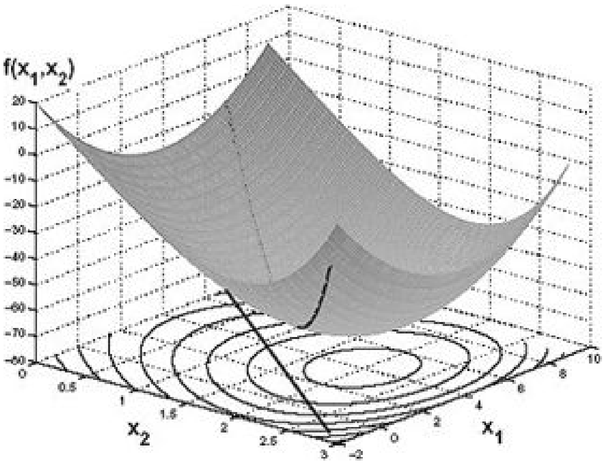


Figure 3.4: slice of the 2d quadratic gives a 1d quadratic.

other inherently, and sometimes are somewhat confounding incidentally. Briefly, the LCGM is used for approximating the maximum of the GPML, and the NLCGM is used for the GPML optimisation. Deeper content will be included in the later section.

Starting from the SDM, If given a quadratic function in a form $f(x) = \frac{1}{2}x^T Ax - b^T x + c$, it is directly perceivable that the location of the minimum point could be obtained by solving the simple linear derivative $Ax = b$ of the quadratic function $f(x)$. If the characterising matrix A is positive definite, it ensures monotonicity in every direction radiating from the minimum location. It is also a property of the quadratic function that any directional slice parallel to the function axis $f(x)$ which is N-dimensional arrives at a $N-1$-dimensional quadratic function in that sliced plane whose minimum could be evaluated from the linear relation addressed before (for illustration, see equation (3.4)).

Therefore one could naturally come up with such a thought that starting from any point on an unknown quadratic function surface, randomly given a search direction (equivalently making a slice), the minimum in that direction could be found through $Ax = b$. Then one takes this minimum point as the next iterative point, by repeating this process, one should be able to arrive at the final minimum. The iterative update will take the classic updating scheme at the k^{th} iteration $x_{k+1} = x_k + t^* P_k$. One could easily notice the reappearance of the same updating procedure as in the QNM. To use Wolfe conditions again is not disputable. However, the SDM offers another routine as provided by its second crucial property, the orthogonality property, which associates tightly with the CGM to be expounded later. In the SDM, the search direction is

$P_k = -\nabla f(x_k)$, which is the negative gradient of the function. By the definition of gradient, the gradient is the direction setting the function to greatest change at a given location, which exactly shares the same meaning as the steepest descent if the function profile is descending. The gradient at the $k+1$th iteration is $\frac{df(x_{k+1})}{dt}$, by the chain rule, it gives $\frac{df(x_{k+1})}{dt} = f'(x_{k+1})^T \frac{dx_{k+1}}{dt} = f'(x_{k+1})^T P_k$. By setting the derivative to zero, thus $f'(x_{k+1})^T P_k = 0 = P_{k+1} P_k$. Hence every pair of adjacent iterations has a pair of mutually orthogonal search directions. As interpreted in a graphical way, the orthogonality between search directions indicates these directions are mutually perpendicular to each other. Using this property, the following deduction arrives,

$$\begin{gathered}
P_{k+1}^T P_k = 0 \\
(b - Ax_{k+1})^T P_k = 0 \\
(b - A(x_k + tP_k))^T P_k = 0 \\
(b - Ax_k)^T P_k = t(AP_k)^T P_k \\
P_k^T P_k = tP_k^T AP_k \\
t = \frac{P_k^T P_k}{P_k^T AP_k}
\end{gathered} \tag{3.59}$$

Because the SDM search process is orthogonally directed as shown in Figure 3.5, it could be alternatively viewed as a process of decomposing the initial displacement from the starting point to the target minimum into several segments in two general directions perpendicular to each other. This fact realises one important message that the local guidance as directed by the negative gradient direction is somewhat inefficient, as the holistic search has to incessantly adjust itself to approach the destination. If the rule of strict rectangular angled search could be crossed, might there be any chance a freer combination of search directions could result in a process with far less steps? This question irrigates the emergence of the LCGM. The LCGM is a search process resembling the SDM whose search directions are mutually A-orthogonal to each other.

Simple orthogonality is represented by $u^T v = 0$ in symbolic form, the graphical representation is two mutually perpendicular vectors. It has been proved by mathe-

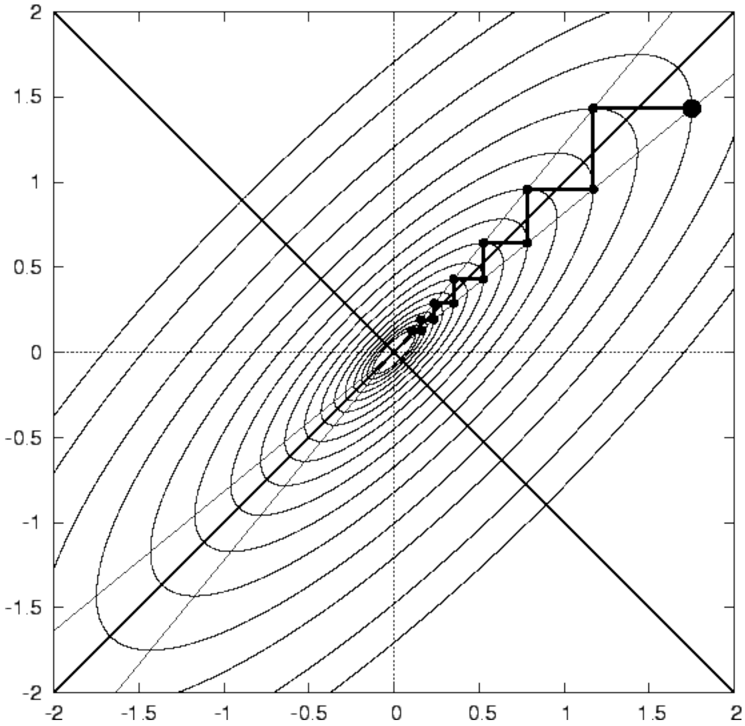


Figure 3.5: [3]steepest descent as described in graphs.

matical deduction that the adjacent two search directions are mutually orthogonal. Here is a more intuitive explanation which is forthwith more illustrative for the A-orthogonality. At the k^{th} step of the SDM search the gradient along the k^th direction at x_{k+1} has to be zero (that is how it is derived). Thus the resultant gradient (the negative of the next search direction) should decompose no component to the k^{th} direction, which in other words means that the next $(k+1)^{th}$ direction is perpendicular to the k^th. Although searching in orthogonal directions makes no collateral search in either direction due to directional decomposition, it is not necessarily indicative that the searches in a pair of orthogonal directions are completely disparate. In fact, the search in either of a pair of orthogonal directions will still cause collateral effects on the change of gradient to its orthogonal direction. To completely eliminate the collateral effect, one should make the change of gradient in the v direction orthogonal to u. The change of gradient in the v direction could be represented as $\delta_v(\nabla f(x)) = Av$. Thus the orthogonality is what could be termed as A-orthogonality,

$$u^T Av = 0 \tag{3.60}$$

Being A-orthogonal is the same as being orthogonal in the reciprocity, which means

[3] Source: http://komarix.org/ac/papers/thesis/thesis_html/node10.html

that u and v are mutually A-orthogonal to each other. This relation is also termed as u is conjugate to v (this is how the name of CGM arises).

Because the A-orthogonality ensures high efficiency on each of the search directions, the LCGM converges within far less iterations than the SDM. In fact for a perfect quadratic optimisation problem, the LCGM at most converges to the destination in n steps, where the n is the dimensionality of the characterising matrix A. The following graph illustrates the difference in the mechanism of SDM and LCGM.

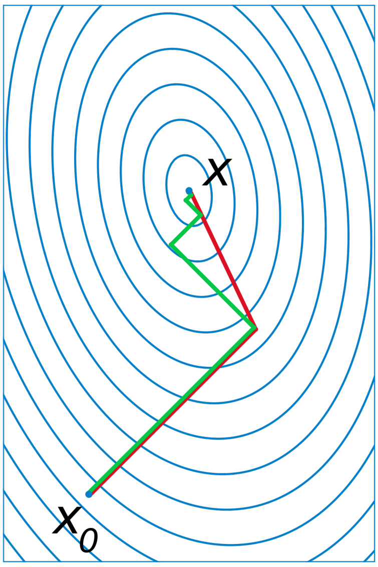


Figure 3.6: [4]LCGM vs SDM.

Unlike the SDM where the search directions are naturally $-\nabla f(x_k)$, the search directions in the LCGM require a special technique to generate. All the generated directions in the entire search process are required to hold conjugacy with the directions generated before. There is more than one way to construct such a collection of directions. The traditional LCGM uses what is known as Gram-Schmidt conjugacy, and it gives the following generative form,

$$P_k = r_k - \sum_{i<k} \frac{P_i^T A r_k}{P_i^T A P_i} P_i \tag{3.61}$$

where P_k is the search direction, and r_k is the negative gradient. However, remember that all these mathematical derivations and assumptions are based on the function in quadratic whose first derivative is a linear function and whose second derivative is a constant. The GPML does possess a quadratic form, but that quadratic

[4]Source: https://en.wikipedia.org/wiki/Conjugate_gradient_method

form is terms of prediction Y, which means the LCGM optimisation used here is to maximise the GPML w.r.t the Gaussian predictive mean of Y at a given set of hyperparameters. Hence such LCGM optimisation for the GPML is not a parallel to the Newton method and the QNM discussed in this chapter. More specifically, such kind of application of the LCGM is used for approximating the maximum of the logarithmic GPML to straightly give out the best prediction in the data space. The advantage by doing the line search for the most probable predictions over the more expressively straight matrix handling is that the LCGM offers the opportunity of finding the target prediction without the need of computing the matrix inverse [54]. The comparative benefit by doing so is proportional to the dimensionality and size of the training dataset.

The optimisation against the hyperparameters is a different story. There is no quadratic association between the logarithmic GPML with its hyperparameters. Therefore the traditional LCGM can no longer be effective here. Such fact of nullification gives rise to the NLCGM.

The major characterising difference of the NLCGM from the LCGM is that, rather than involving the A matrix in the algorithm, the NLCGM replaces A with an approximation from the gradient $\triangledown f(x)$. It's an analogous relation between the LCGM and NLCGM to the relation between Newton's method and QNM. The NLCGM still holds firmly the conjugacy property. To ensure the conjugacy, there are several ways to generate the search directions. The popular choices are the Fletcher-Reeve and Polak-Riberie conjugacy. The search direction in the NLCGM is computed by:

$$p_k = -\triangledown f(x_k) + \beta_k p_{k-1} \tag{3.62}$$

where β is the scalar to ensure conjugacy.

The update of β with Fletcher-Reeve [55]:

$$\beta_{k+1} = \frac{\triangledown f_{k+1}^T(\triangledown f_{k+1} - \triangledown fk)}{||\triangledown f_k||^2} \tag{3.63}$$

To determine the search step t in the NLCGM is not as simple as in the LCGM. There is no definite linear function to solve. One should again search the aid from Wolfe conditions in equation 3.57 and 3.58. More details will be explained in Chapter

4.

3.5 Conclusion

This chapter has taken a step-by-step process to associate inner relations among the relevant mathematical models and unravel their mechanisms accordingly. The Decision Tree forms the basis and framework, where multiple models could fit into the picture for piecewise regression. The Gaussian Process serves as the outline to lay down what specific regression picture one can imagine out from such a process. The further filling of such an outline bifurcates into the Gramacy and Chipman based Treed Gaussian Processes. Some-side related mathematical knowledge has also been added to complement the argument, such as the Markov Chain, Wolfe conditions etc. The gist of this chapter is to explain rather than to juxtapose, thus no massive content here is used for comparing which TGP is better. Standing upon the perspective of the theory, the TGP indubitably has a stout and rigorous mathematical basis. But, one should still bear in mind that the TGP is set on the Bayesian framework indicating its intrinsics of uncertainty. Using uncertain probabilistic models modelling against the practical physical uncertainty mostly is a pure solution provided by mathematics, completely devoid of any physical insights. This could be observed from the content in this chapter, where not one equation nor one symbol is related to physics.

Chapter 4

Benchmarking of the CTGP against the GTGP

4.1 Chapter Overview

The last chapter has spent some paragraphs covering the relation between the CTGP and GTGP. The GTGP developed by Gramacy has already proved itself as an effective statistical model through its applications on various datasets including both synthetic and practical data. The CTGP, stemming from the same mathematical ground as the GTGP, is proposed to be at least equally effective, or with a high chance to perform better than the GTGP, given the advantages from its dedicated exhaustive search for the local hyperparameter optimum in each leaf.

In this chapter, the popular Motorcycle Accident Data (MAD) [56] will be used for a case study to verify the effectiveness of the CTGP. The MAD is also the dataset Gramacy used for verifying the GTGP [57]. Thus it naturally suggests that the GTGP could be used here as a reference to benchmarking the CTGP in terms of a number of criteria such as prediction reliability, computational cost etc. The GTGP, as introduced by Gramacy, was programmed in and made exclusive to the language R; the current CTGP is developed in Matlab. If the benchmarking is made across the languages, it will be unfair and pointless. Therefore, a simplified version of the GTGP has been developed in the Matlab for this very purpose. The simplified GTGP in Matlab, as compared to the full version of the GTGP in R, is considered to be a strategic replication which retains all the essences and ingredients to form a GTGP, while discarding the Limiting Linear Model (LLM) introduced by Gramacy [57]. The LLM is a technical sub-algorithm which adds robustness and smartness into the GTGP model. It can automatically evaluate the necessity to perform a full GP analysis for a selected leaf. If a linear modelling of the selected leaf meets a certain performance evaluation criterion, it will be preferred over applying a full GP modelling. Therefore, in this way, the computational burden could be substantially allayed. Because the LLM is more of an auxiliary type of optimisation to the GTGP, the exclusion of the LLM in the simplified GTGP will not impose malignant damages to the structure of the GTGP. Then, through such comparison between two TGPs all together in Matlab, one could not only intuitively perceive the strength of the CTGP under each criterion, but also take a comprehensive insight of its mathematical mechanism in sampling, inference etc.

4.2 Data Explanation

In the statistical community, the MAD has a good reputation for testing the performance of nonstationary regressions [31]. It gains its popularity from its high contrast in variance in a piecewise manner. The MAD measures the acceleration of the motocyclist's head in terms of the time right after a crash [56]. The data is comprised of 94 data points, but features an explicit behaviour of heteroscedasticity.

In Figure 4.1, it is shown that the MAD approximately holds four regions based on the change of piecewise variance, or five regions by matching it with linear trends. Either way is reasonable, and both of them could find the corresponding reasoning criteria in the kernel of a GP. The recognition criterion based on the variance could be linked to the noise variance hyperparameter σ_n^2; while the later criterion based on linearity could be manifested through the length scale hyperparameter l^2. In the human recognition system, the predictions are often given based on the most evident feature observed from the data. If the observation is confounded by more than two equally weighing features such as observable linearity and contrasted variance, the brain tends to fail in coordinating both criteria to produce a decent fused prediction. Therefore the MAD offers itself as a visual perspective through which one can take a clinical view on the advantage of the TGP over pure human recognition.

Back to the data itself, to simplify the explanation on the data, the data is pre-

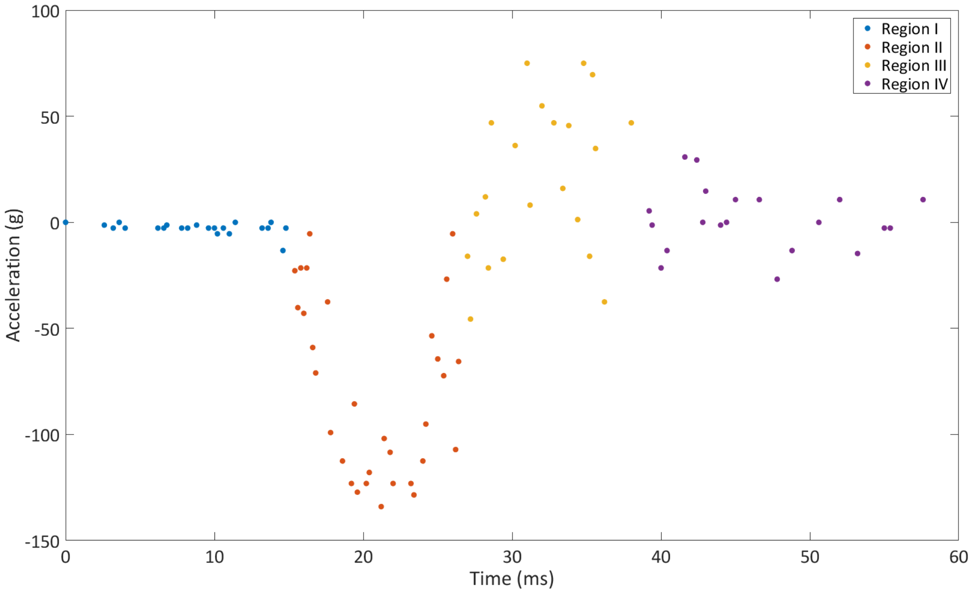


Figure 4.1: Motocycle Accident Data.

sumed to contain four regions based on the variance difference. The first region is indubitably displayed as a extremely distinguishable flat line within the interval [0,16], where the dispersion level of the data points are restrained tightly. The second region is a bit arguable to human's visual recognition. It lies across [16,30], where a consistent medium level of variance could be observed. The third region sits approximately on [30,40]. It features the most drastic variance in the whole plot. It is hard to accurately bound its range, because the consistency in the variance is violated by the gradual shrinkage of the variance in [38,43]. The last region could ostensibly be modelled as a flat line with a much greater variance than the first flat line region.

It is highly expectable to see at least 3 regions to be identified from both TGP models. The partition between the stated region II and III is a bit tricky to observe visually, whose identification can directly demonstrate the capability of the TGP.

4.3 Benchmarking on the Performance

4.3.1 MAD with the GTGP

Although both types of TGP are jointly built from the foundation of Breiman's treed model for constant regression, the GTGP with its unique dual sampling scheme can sample significantly faster than the CTGP with the more traditional mono sampling scheme. However, the faster sampling rate benefited from the dual sampling scheme trades off with the number of iterations required to search for the MAP of the tree. Through many runs of experiments, it has been found that for the MAD, it takes on average 20000 MCMC steps for the GTGP to converge to a rather stable state. In general it gives the following predictive view of the data space,

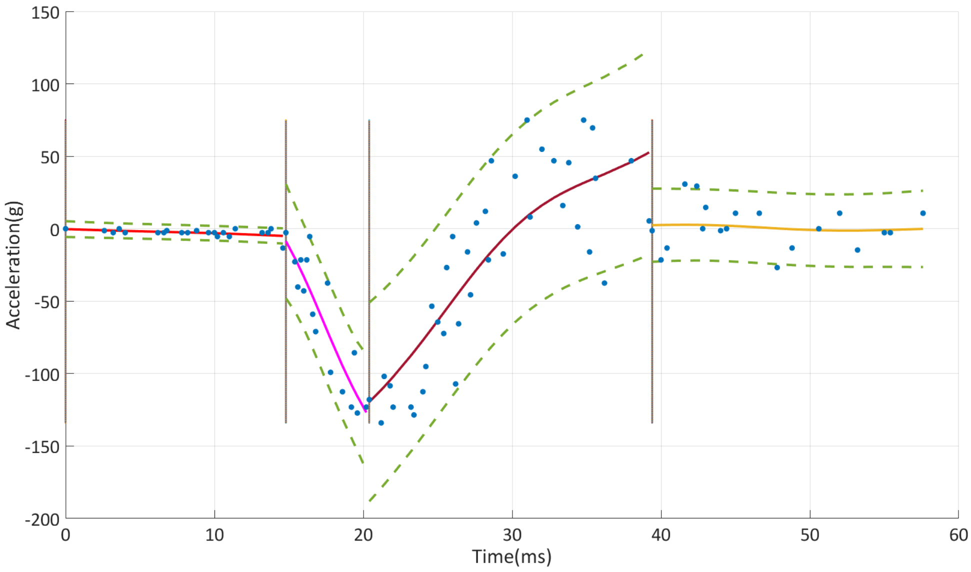


Figure 4.2: GTGP performed on the MAD (95% CI).

N.B. Regions are marked as I, II, III, IV from left to right, the same numbering order applies to all the following figures.

From Figure 4.2, it is shown that the GTGP tends to explain the data space with four regions which agrees with the four region assumption from the observation. The GTGP successfully captures the linear behaviour at the beginning of the data.

The following region II, captures a linear behaviour of the data rather than the consistency in the variance as described in the presumed region II. The prediction from the GTGP is still thoughtfully a reasonable one, because the GP makes predictions based not only on the hyperparameter σ_n^2 which describes the variance, but also the length scale l^2 which models the smoothness of the prediction. The variance is also evidently contrasted between Region II and its following region III. A smoother prediction is always preferred by the GP, which means the turning of the predictive curve will decrease the Gaussian Process Marginal Likelihood (GPML) to some degree. Hence to place a partition at 20 is still highly acceptable and reasonable.

The third region is modelled with a slightly bended curve approximating a linear trend, but it could be argued such an interpretation of the data space is a bit rough and inaccurate. As there is a relatively clear discontinuity in the variance at around point 30 in the MAD dataset. The predictive curve captured the trend well in this region, but entailed a residual trouble to its exit at point 39, where a rather sizeable discontinuity has obstructed the smooth flow from the region III to region IV. The in-depth cause of the discontinuity descends from the high noise variance hyperparameter σ_n^2 that the GTGP used to describe the region. Apparently, such a description also inappropriately explained the point at 39 as noise, which occupies the crucial position to transiently and smoothly join region III and region IV together.

The last region to the far right end produced by the GTGP shows an excellent agreement with the presumed fourth region. Apart from this figure, which only shows four typical regions given by the GTGP, it has also been frequently observed that the GTGP makes a partition at points 18 and 25 at the removal of the partition at point 20 in Figure 4.2 to highlight the joint part between region II and region III. The variance in the interval [18,25] does inflict a lumped inconsistency to the seemingly uniform variance in the presumed region II. Overall the GTGP has shown its sensitivity to both the trend and variance of the data.

As has been addressed before. the GTGP developed by the author as a replica of the raw Garamacy's TGP in R are not exactly the same. Gramacy's TGP has more additional features such as limiting linear model etc. In order to show the difference in the result between the GTGP and Gramacy's TGP in R, Figure 4.3 is borrowed from Gramacy's paper [57] to illustrate the difference. Figure 4.3 shows that Gramacy's original TGP in average makes two partitions in the data space. The GTGP also does occasionally make the same partitions, but the three-partition result is more common.

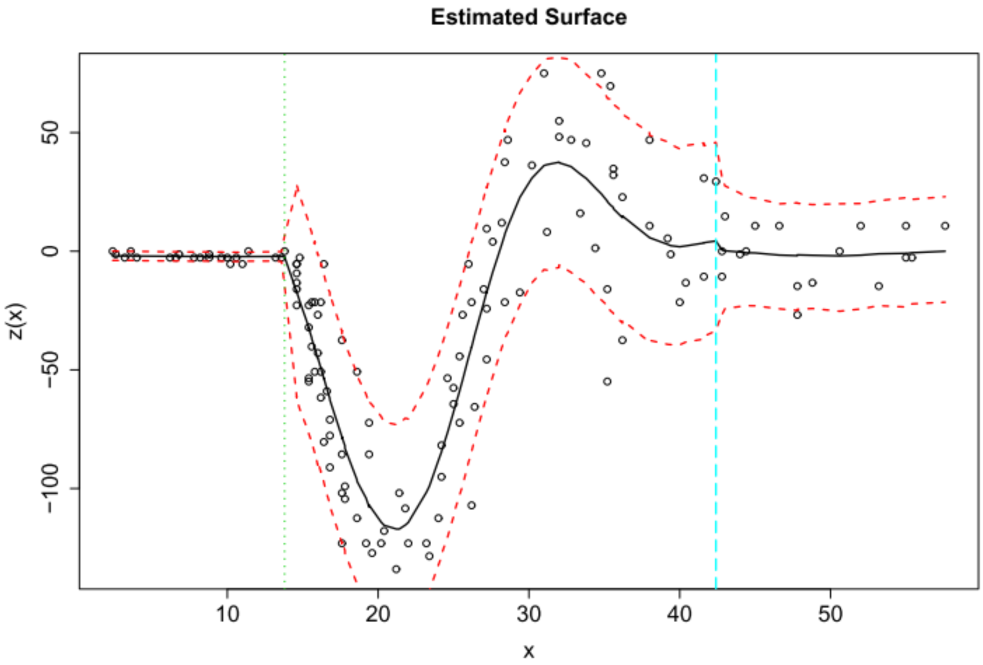


Figure 4.3: Gramacy's TGP coded in R performed on the MAD (95% CI).

4.3.2 MAD with the CTGP Optimised Stochastically

Now, applying the CTGP with stochastic optimisation to the MAD, 300 MCMC rounds (the CTGP requires much less number of MCMC rounds to converge) will give out a prediction as in Figure (4.4). The four regions produced by the CTGP also show a strong agreement with the four presumed regions described before.

The CTGP and GTGP are unanimous at putting partitions at points 16 and 39. The only difference is about the second partition which yields the different modelling in the regions II and III.

Unlike the region II by the GTGP, the region II in the CTGP, rather than putting a partition at 20 to stress on the sharp turning in the trend, modelled the turning in a sense of mildness. It could be seen there is a general consistency in the variance in this region. Although at the bottom there is a slightly discordant rise of the variance which could occasionally be captured by the GTGP, there is no strong evidence to support a region to be assigned at the trade of losing the capture of the consistent variance; besides, its presence is short lived considering the scale of the data space.

The region III produced by the CTGP complies with the presumed region III extremely well. The dispersion level there is significantly larger than the other regions. Despite assigning an equally large belief in the variance, the CTGP successfully accommodated the decreasing trend, and established a nice and natural connection with the last region. However, as has been stated before, in the interval [35,45] there features a gradually decreasing variance which suggests a continuous heteroscedasticity. Hence, where to put the last partition could be less strict. To put it at point 39 will yield the highest contrast in the variance between region III and region IV, in which case the linear behaviour in region IV will extend at its maximum to the left.

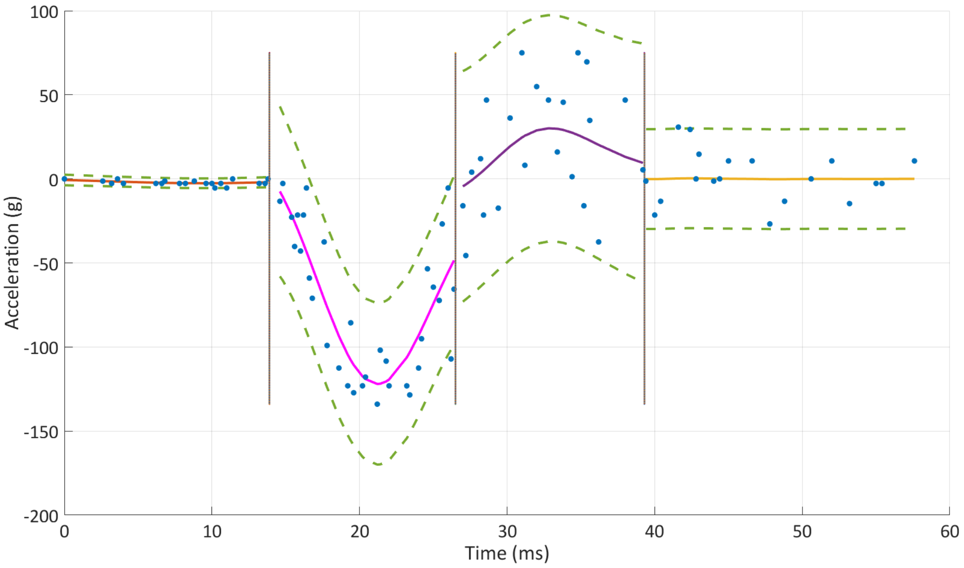


Figure 4.4: CTGP performed on the MAD.

In general, through comparing the GTGP and the CTGP in terms of the quantitative analysis based on the Mean square error (MSE), it has been found that the CTGP with a stochastic optimisation has an overall MSE value of 1513, while the GTGP has a larger value of 1665. The MSE for the middle two regions in the interval [14,40] given by the GTGP is 1466, while the CTGP gives a value of 1322. In the remaining two regions, the MSE levels are both similar for the CTGP and GTGP. The Figure 4.5 shows the details of the squared error of both the GTGP and CTGP at each data point. It shows clearly that the higher predictive error produced by the GTGP accumulates from the two regional joints at points 20 and 39. It also could be easily noticed the CTGP has a greater error around point 36, which is reflected in the prediction as an underestimation at that place.

It could be sensed that the GTGP is more sensitive to the change of the direction of the trend, while the CTGP is more sensitive to the difference in the variance. Besides, the GTGP has a general taste in favour of interpreting the data in a linear fashion, which could be partially derived from its sensitivity to turning points. Since both models are built on the same Bayesian framework as well as the same GP inference, such a difference is highly likely to result from the difference in their sampling schemes.

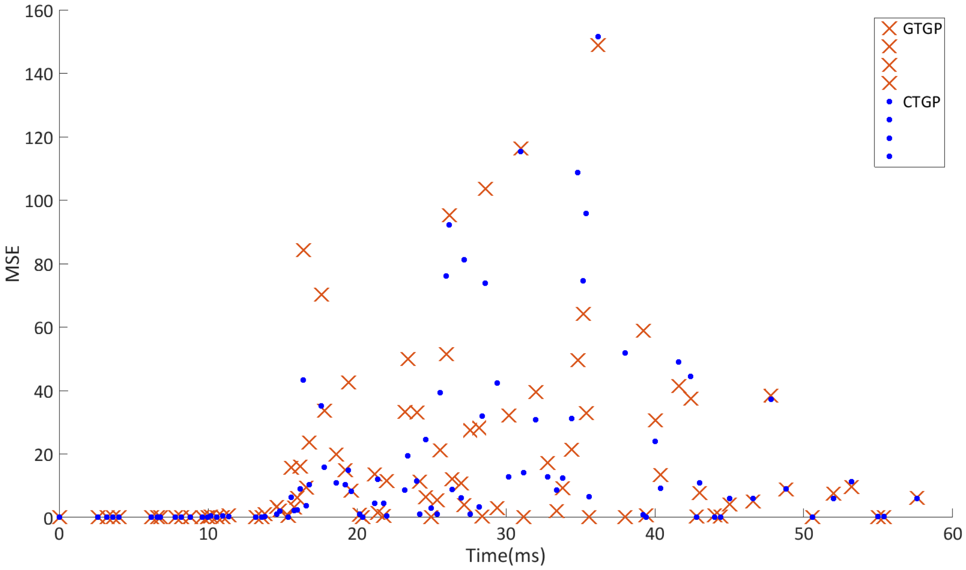


Figure 4.5: MSE comparison between the GTGP and CTGP.

Figure 4.6 shows the variation of the logarithmic posterior of the tree during the 50000 MCMC steps in the Markov space. The mechanism of the dual sampling scheme will unfold a fully randomised space for the tree as well as all its hyperparameters to explore. Therefore the variation of the logarithmic posterior bears a cluttered look with very limited predictability and traceability. However, the intertwined Gibbs samplings among the sampling participants will carefully guide such search towards the regions containing high MAP trees without violating the random behaviour too much. Hence it could be seen in the figure, approximately every 10000 steps the search will enter the high posterior region. In an MH-based MCMC sampling, it is not common to see the log posterior could drop for over 5 units at a logarithmic scale, which requires a very small acceptance ratio of less than 10^{-5}. In the figure, such a drop is pervasive, and even a plummet by over 50 units does occasionally take place. Such a non-ordinary behaviour is an ordinary outcome from the dual sampling scheme. In the dual sampling scheme, a pair of bridging Gibbs

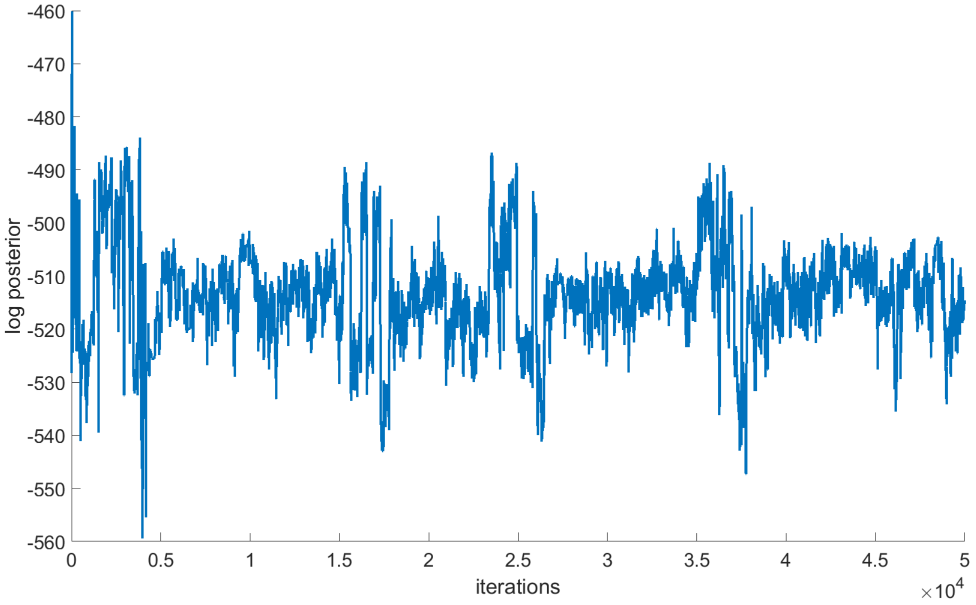


Figure 4.6: Variation of the log posterior during 50000 MCMC steps for the GTGP.

samplings is established between the tree structure and the hyperparameter space, namely $p(T|\theta)$ and $p(\theta|T)$. Every time when a new tree structure is accepted, random draws will be made across all leaves to update the hyperparameters. However good or bad these draws are, they will be unconditionally accepted without passing the check of the MH criterion like the tree structure did. Therefore, there could be a massive drop in the posterior. To unconditionally accept the draws of the hyperparameters is a contradictory matter with coexisting advantages and disadvantages. Its advantage is in the pertinence to the completeness of the sampling space. The unconditional acceptance allows the tree structure and the hyperparameters to form combinations with infinite possibilities, thus theoretically the search of the MAP tree through the GTGP should be more comprehensive and exhaustive. It also prevents those bad leaves from enhancing their hyperparameters too much, which could potentially trap the search process. The disadvantage is equally remarkable that those good splits will also suffer from its side effect.

Figure 4.6 shows the same plot for the CTGP optimised stochastically in a run for 300 MCMC steps. The figure shows clearly the convergence of the logarithmic posterior through the CTGP process. There is no major drop by over 5 units in the logarithmic scale. Such a well behaved posterior convergence, to some extent, manifests the CTGP's semi-stochastic nature. In the mono-sampling scheme of the

CTGP, the only sampling participant in the Markov space is the tree structure whose hyperparameter setting will be almost deterministically evaluated through either a stochastic or numerical optimisation. Since each tree is fully optimised, the difference between different trees in the posterior will be drastic. A good partition could gain an increase of over 10 units in the overall logarithmic posterior as shown in the figure (around the 100th iteration). Thus there is nearly no chance for a tree with bad partitions to outstrip a better tree during the MH check, unless the two trees strongly resemble each other. Hence the CTGP ensures a convergence in the posterior. However theoretically the strict convergence could potentially be a factor that negatively influences the search of the MAP; it resembles the general problem faced by using a greedy algorithm in regression [58]. Simply speaking, an assigned partition, if too informative, could mask other distinguishable characters in the behaviour of the data space. An example is shown in Figure 4.8, that if the initial partition was not put at point 28 like in Figure 4.4, the CTGP will eventually produce the partitions to separately model the lumped data at the bottom. It definitely suggests that the CTGP partitioning relies on the sequence of putting the partitions. Through a number of experiments, it has been found the locations around points 14 and 39 have the strongest attraction to partitions. In Figure 4.7, the first major escalation at the 7^{th} iteration and the second major escalation at the 88^{th} iteration correspond to these points respectively. From many experiments, it has been found that these two partitions are almost always the first two to be discovered by the CTGP, while in the GTGP the order might vary. Thus it could be said, although the tree is constructed from sampling in the CTGP, its final partition view is rather deterministic. While in the GTGP, the final partition view has more chance to vary, which could be argued as one essential advantage over the semi-stochastic CTGP.

4.3.3 MAD with the CTGP Optimised Numerically

The CTGP with the stochastic optimisation has the advantage of being more accurate in terms of inference. However, its high computational cost is limiting, although there are solutions to relatively allay the computational repetition to some extent. To substantially reduce the computational repetition for the CTGP, the stochastic optimisation may be replaced by the numerical optimisation for the hyperparameters. The basic theories about the three approaches used in the numerically-based optimisation have been introduced in the last chapter: the Newton's method (NM),

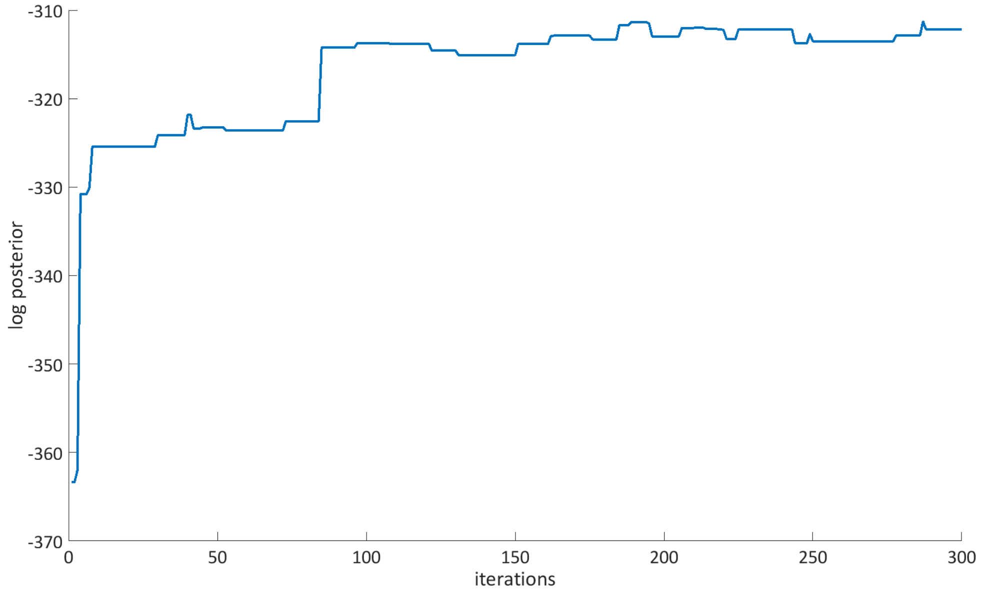


Figure 4.7: Variation of the log posterior during 300 MCMC steps for the CTGP.

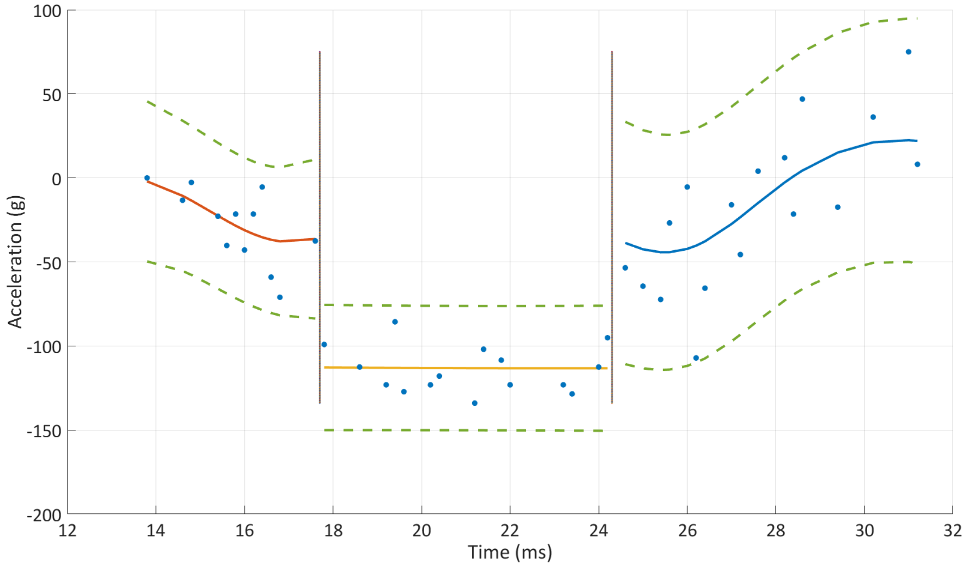


Figure 4.8: Partitioning the MAD in the interval [14,32] (95% CI).

the Quasi-Newton method (QNM) and the Nonlinear Conjugate Gradient method (NCGM). The three methods will in order be applied to the MAD. The predictive result from the NM is shown in Figure 4.9. The number of MCMC rounds for the CTGP with any numerically-based optimisation is the same as the CTGP with the stochastically-based optimisation. Since the numerically optimised CTGP runs much faster, 300 rounds are assigned for an enhanced convergence.

Newton's Method

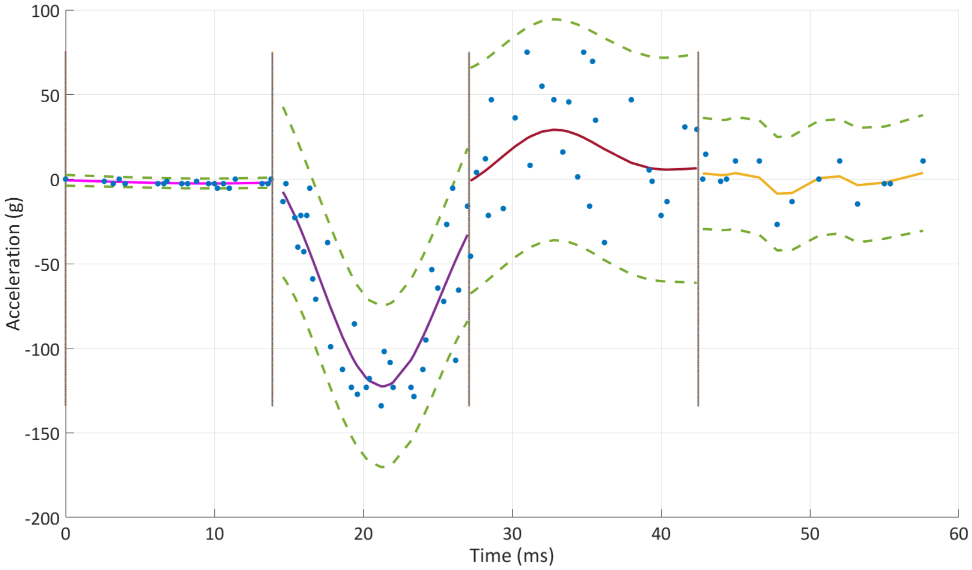


Figure 4.9: CTGP with the Newton's method on the MAD (95% CI).

The result from Figure 4.9 agrees with the prediction given by the stochastically optimised CTGP shown in Figure 4.4. The almost exact coincidence in the partitions is to some extent, beyond the expectation. The predictive result via a numerical optimisation process will chiefly suffer from two inherent problems: the multiple local maxima (or minima, depends on the target function) issue and the attack from saddle points. The prediction in the region IV has revealed that the issue with the local minima definitely has its role played during the optimisation process. The staggering predictive curve in region IV indicates a local maximum for the logarithmic GPML with a small length scale hyperparameter l^2. But as one can see from the result, despite the overstress on the l^2 providing the prediction in region IV, which still seems reasonable considering the lack of data points there, no other disastrous negative effect has shown its presence at least in this particular case study. In fact through repeatedly experimenting on the NM with the MAD, it has been found the treed model itself is a natural counterbalance to the negative effect from the attack of the saddle point. The multiple local maxima will be the only dominant force to influence the search of the global optimum. The isolated experiment on the dataset consisting of the first 25 data points in the MAD is used to clarify this intriguing phenomenon.

Figures 4.10 and 4.11 show the inference on the experimental data by the stochastic and NM optimisation respectively. Apparently the inference with the NM failed to capture the trend of the data. Rather than arriving at a local maximum, it is trapped by a saddle point (because at its optimum hyperparameter set, the Hessian matrix of its logarithmic GPML function has both positive and negative eigenvalues).

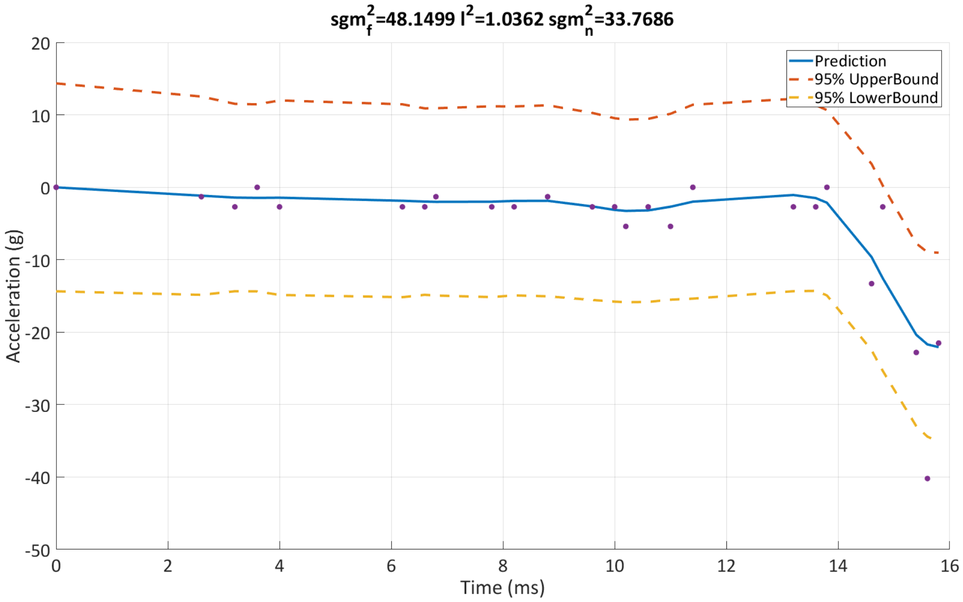


Figure 4.10: inference with the GP optimised stochastically.

It has been found that the searching process will be attracted by the saddle point for 40 times on average out of 300 MCMC steps; but there is good news. It has also been found that most of the attractions take place in ill-conditioned regions featuring an abrupt change of trend such as in Figure 4.10. Since such regions are less desirable to be accepted, a reduced GPML induced by the saddle point does unintentionally help the algorithm to reject the trees with those bad leaves.

Multiple local maxima for the logarithmic GPML function is another classic problem which can cause the NM method to generate inferior predictions within a leaf. But the inferior predictions within the leaf could be resolved by re-performing a stochastic optimisation to find the global optimum.

There is one additional issue worthy of one's attention apart from these two classical problems, which could also trouble the performance from the NM optimisation. It is the stopping criterion of the NM optimisation. To choose a good stopping criterion

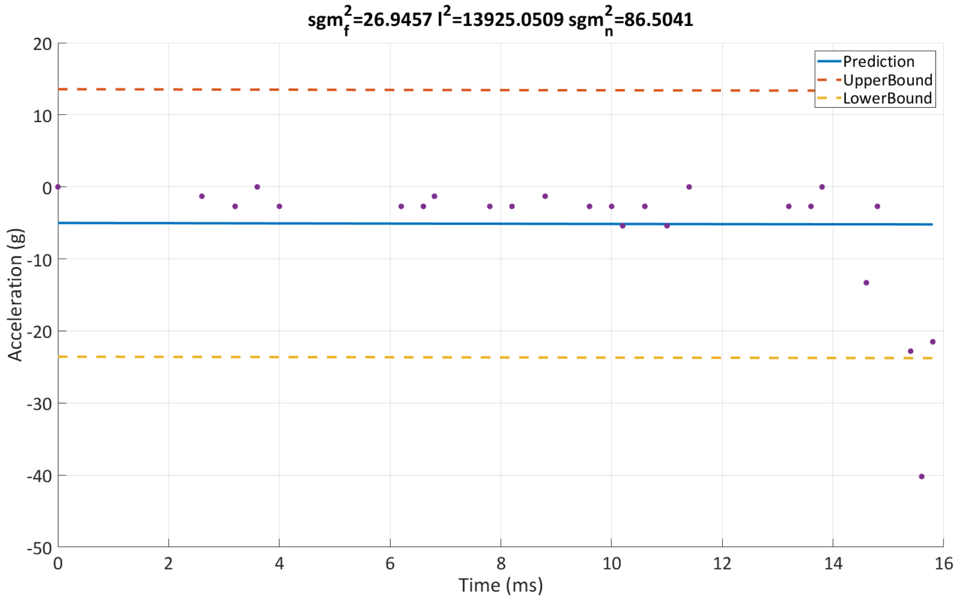


Figure 4.11: inference with the GP optimised by the NM at a saddle point with bad length scale l^2.

is not merely an issue concerned with the NM, but a general essential procedure in all the numerical optimisation methods. Hence the criterion used in the NM will have the chance to apply to the QNM and NLCGM as well. The choice pool for the stopping criteria is enormous, but to find one that fits well to the data space is non-trivial [59][60]. Because usually a criterion monitored by a static threshold that tries to capture the onset of the convergence fails to be effective in a complex data space, especially those high dimensional function spaces containing multiple local modes (maxima or minima). Since different local modes have different extremity values and curvatures, the convergence rate and the final convergence precision could be drastically different. In a treed model, things are even worse. The regions generated from the partitioning yield separate new GPML function spaces subject to the individual optimisation, which means a good threshold for region A might be an extremely bad one for region B. Thus the common choices for the stopping criteria shown below will all fail in the treed model.

$$
\begin{aligned}
|x_{n+1} - x_n| &< \epsilon \\
\frac{|x_{n+1} - x_n|}{x_n} &< \epsilon \\
|f(x_n)| &< \epsilon
\end{aligned}
\tag{4.1}
$$

where x_n is the n^{th} input vector during the optimisation search, and $f(x_n)$ is the n^{th} target function value and ϵ is the threshold.

In fact, the second criterion shown above does contain the concerns on the dynamic behaviour of the function characters across the regions, but it still fails at times. It is because a good ϵ of the second criterion for a large scale region still might be a bad one for being too greedy and demanding in another much smaller scale region, where the same ϵ will drive the optimisation search to a point evincing matrix singularity problems where the determinant of the characteristic matrix is exceedingly small (more details in the discussion part of Chapter 6). The Figure 4.12 shows an example of the failure of the optimisation in a region due to l^2 being too small. The determinant of the Hessian here dropped below 10^{-16}.

An extremely small hyperparameter can cause the convergence to collapse, as does an extremely large one. The presence of an extremely oversized hyperparameter set at a scale of 10^{20} could also happen during the search if the stopping criterion is not operated properly. An extremely large hyperparameter set encodes the belief of the algorithm towards the data space that all the data are interpreted as a form of noise. Through the second derivative test, it has been found that the large scale hyperparameter region in the GPML function space is a high dimensional plane, as the second derivative test presents a 3×3 (SE kernel) Hessian matrix whose entries are all extremely small value close to zero and all the partial gradients are close to zero as well. It is rather reasonable, because everything is interpreted as an extremely high level noise, the data pattern will have little effect on the GPML. The problem of being dragged into the large scale hyperparameter plane is caused by the fact that the convergence is not fast enough to reach below the threshold before it enters that plane. Once the search entered that plane, because both the gradient

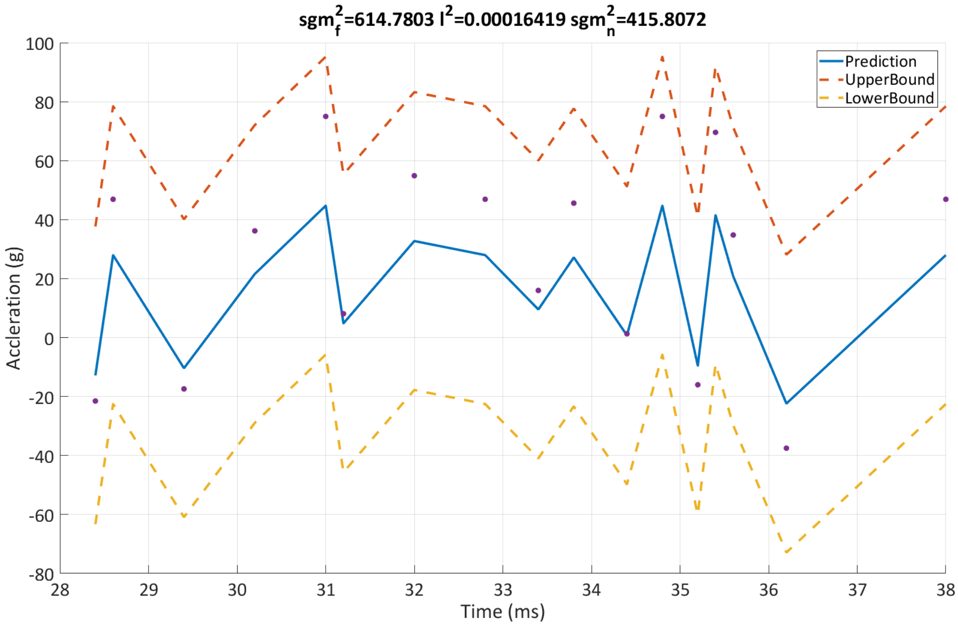


Figure 4.12: optimisation collapses when l^2 is too small.

and the slope of the gradient become flat, the actual step required to the next point will increase rather than decrease (for mathematical proof, check equation (3.48)). Hence it will diverge if criterion one or two is used. The third criterion could resist this problem to some level. Because the third criterion could use the first Jacobian vector (gradient information) as the standardised target function, and set its ϵ to a positive value close to zero to stop the divergence process. However, it has been found the third criterion suffers from the sporadic presence of an exceedingly small hyperparameter. Because the convergence rate for each parameter is not identical, sometimes it could be too demanding to repress one hyperparameter under a certain threshold, which could collaterally cause the other hyperparameters becoming too small.

Considering these two major causes that collapse the optimisation, one could sense that either a set of exceedingly small or large hyperparameters does not suffice a good prediction. Therefore, as long as the optimisation exits safely without collapse due to matrix singularity, the trees containing the bad predictions will eventually get rejected by the Metropolis Hasting algorithm. Thus here a hybrid stopping criterion is used,

$$\begin{aligned} max(abs(Jacobian)) &< 0.001 \\ abs(det(Hessian)) &< 10^{-16} \end{aligned} \tag{4.2}$$

Where $Jacobian$ is the first Jacobian vector of the logarithmic GPML function, and $det(Hessian)$ is the determinant of the Hessian matrix, abs means to take the absolute positive value.

Quasi-Newton Method

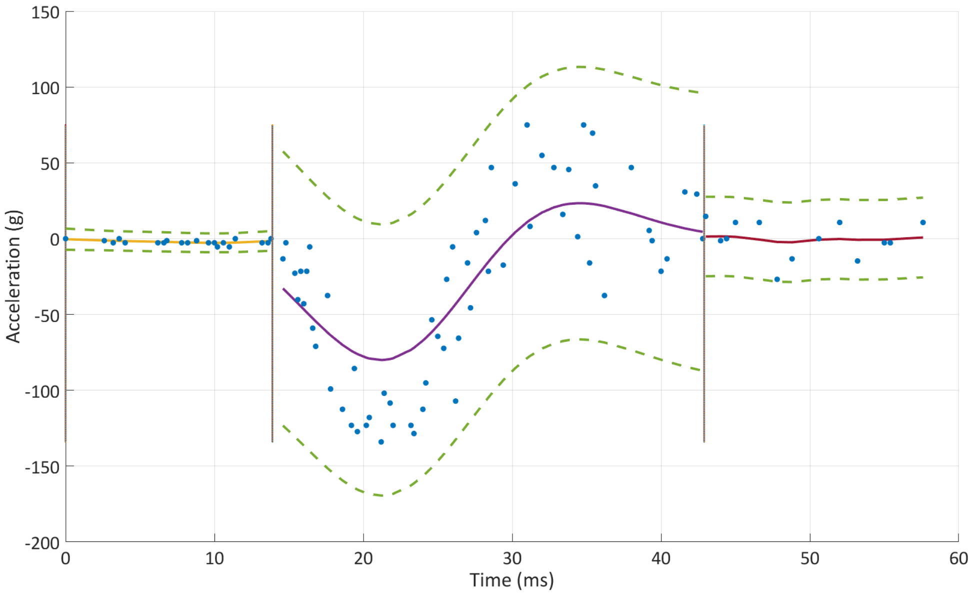


Figure 4.13: MAD with the QNM optimised CTGP (95% CI).

Figure (4.13) shows the typical performance from the QNM with Wolfe conditions incorporated (for reasons why Wolfe conditions must be applied here, see the last part of Chapter 3). One can observe from the figure that the algorithm captured the trend in the first and last region well, but the prediction in the middle region does not conform well with the trend of the data. It failed in modelling the sharp turning at the bottom of the middle region. The confidence interval at the beginning also seems a bit larger than those presented in the former figures. The general cause of such inaccurate modelling ascribes to the incorporation of Wolfe conditions. In a line search-based optimisation incorporated with Wolfe conditions, a suitable value for the additional variable t (the step scale between the two adjacent search points in the line search) in equation (3.57) must be assigned so that the search could approach the local minimum at an effective rate. However, to find an appropriate value for t is not a simple task, especially in a complex function space such as the negative logarithmic GPML which could become extremely intricate when the data space is in a disordered fashion. The current way of finding the t is to use the bisection method to gradually locate the area where a good t lies [61]. However it has been found that there are situations where the the bisection method can never locate a good t, or it could be said there exists such a local minimum whose proximity is highly non-quadratic. Through tracking the history file of the failed search of t, some weird behaviours have been spotted. The first weird behaviour is associated with the distance between the tentative search point θ^*_{n+1} ($*$ indicates a tentative search point) and the current n^{th} search point θ_n (equation (4.3)) shown in Figure 4.14.

$$\theta^*_{n+1} - \theta_n = t \times h \tag{4.3}$$

Where h is the search direction, which remains as a constant depending on the current θ, $t \times h$ is the tentative step size.

As shown in Figure 4.14, the Y axis describes the difference between the θ^*_{k+1} and the current θ_k (step size at k^{th} search point) in a logarithmic scale. The continuous linear drop of the step size has been found related with the consecutive failure in satisfying of the first Wolfe condition equation (3.57). For the Quasi-Newton method, the t has a natural starting value of 1 to initialise the search [62]; this is considered to be a sufficiently large value of t for the tentative search to pass the region which satisfies Wolfe conditions. Then every time the first Wolfe condition is not satisfied, the smaller value for t will be assigned (bisection shrinking) to gradually locate an adequate t. However in this figure, a drop of th to a level of 10^{-13} (t alone will drop to a level below 10^{-16}, this value varies with the initial choice of the approximate Hessian matrix) still failed to find an adequate t. At the step number 44 (the 44th loop in search of t), the sharp unnatural turning indicates a round-off issue, where the MATLAB cannot handle a smaller scale of the value but rounds it off to stay at a level of 10^{-13} (the relative machine precision in MATLAB is 2.2204×10^{-16}, remember the t alone will drop below the limit 10^{-16}, thus the step size $t \times h$ will stay at a level below the limit). Since the step size is so small and Wolfe's first condition still stay unsatisfied, there must be something special about the local geographic profile of the function in that area. It could be further proved from the weird behaviour in Figure 4.15, even without the round off issue, there is no chance for the search to match Wolfe 1st condition. Figure 4.15 shows Wolfe first condition residual ($\delta_{wolfe1} = f_{GPML}(\theta_{tentative}) - f_{GPML}(\theta_{current}) - c_1 th \nabla_{\theta_{current}}$) on a logarithmic scale. The first condition is satisfied by a negative value of the residual.

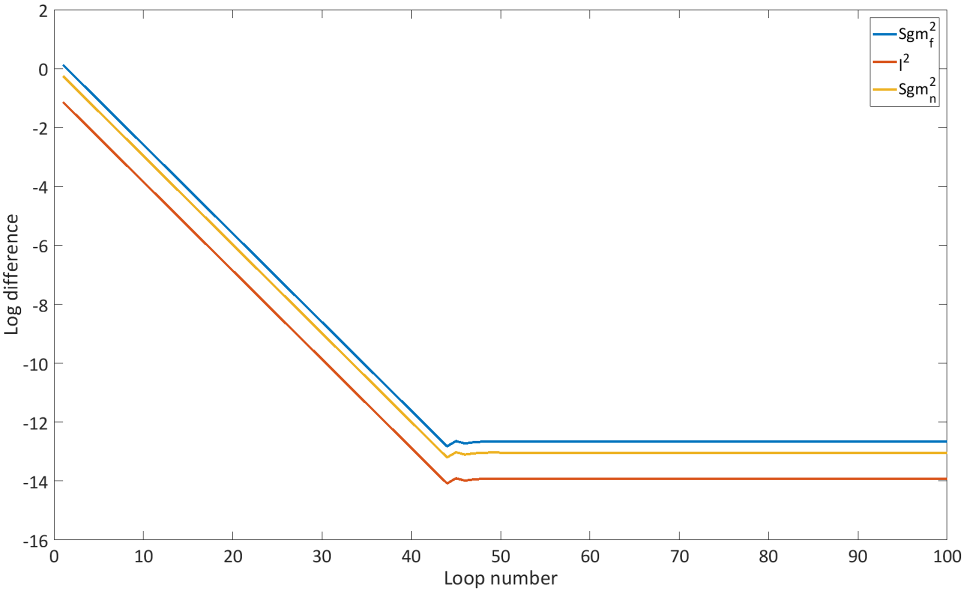


Figure 4.14: Tracking of θ in a failed Wolfe condition.

In the figure, apart from the aberration in the behaviour after point 40 due to a round off issue, the curve exhibits a linearly declining trend when the computation is properly operated. The residual in the normal scale stays above zero before point 40. Recall the general relation between the gradient of the $\log y$ and normal y with respect to x, $\frac{d\ln y}{dx} = \frac{d\ln y}{dy}\frac{dy}{dx} = \frac{1}{y}\frac{dy}{dx}$ (It does not matter to have ln in place of log, the residual will be absorbed into a constant, the linear behaviour will uphold). Then let y be the first Wolfe condition residual, and x be the loop number, it could be obtained that, before y reaches zero, the $\frac{dy}{dx} = y\frac{d\ln y}{dx}$ will be strictly positive, and at $y = 0$, the gradient will be zero. Hence the $y = 0$ is the asymptote for y, thus $y < 0$ is unattainable. Hence Wolfe conditions will fail completely to regulate the descending towards the minimum.

However Wolfe conditions have been rigorously proved to be effective; their failure in the GP is not an exposure of an inherent fallacy, but mainly a reveal of the

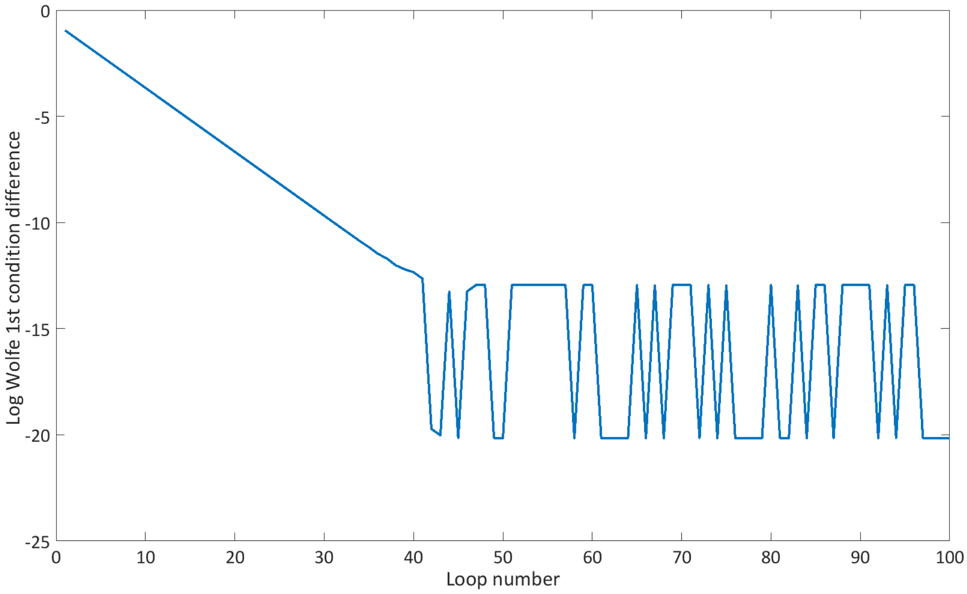


Figure 4.15: Tracking of Wolfe first condition difference.

complexity of the GPML function space. It has been found that Wolfe conditions fail at those hyperparameter positions in the GPML function space where the gradient and the approximate Hessian are both extremely large at a scale over 10^3. The extremely large slope with a bad guess of Hessian will give the Wolfe conditions a false signal that the actual minimum is distant from the current search point. Because the Wolfe conditions ensure a sufficient approach to the target, the satisfying threshold on the RHS of equation (3.57) will be dynamic with the change of t. Hence, what happened in a failed search of t is that, based on the gradient and the approximate Hessian, the Wolfe conditions falsely believe the actual minimum is always far beyond the tentative search point. This issue is illustrated in a simple 2D example in Figure 4.16, where any search on the curve is dynamically above Wolfe threshold. From the illustration, it could also be derived that the occurrence of a failed Wolfe condition is most likely to take place in the proximity of those badly behaved GPML minima featuring abrupt sharp changes which badly violates the quadratic property.

Currently no solution has been found to counter such inevitable failure of the Wolfe conditions. The endless loop in search of the t has to be terminated at some points when it shows clearly no chance of discovering that t could happen. This fact will cause the hybrid stopping criterion used in the NM to be rarely fulfilled in the QNM as well as the NLCGM. Thus expediently, the stopping criterion is monitored by

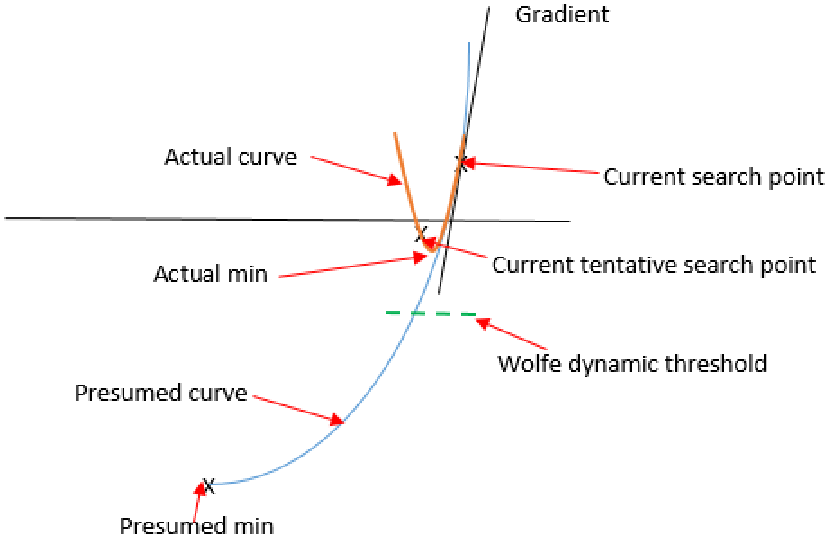


Figure 4.16: Illustration on the failure of Wolfe conditions.

the increasing rate of the logarithmic GPML, when the increasing rate dropped to a certain level, such as: $\log GPML_{n+1} - \log GPML_n < 0.1$, it will terminate. The side effect from this stopping criterion is its reduced accuracy for modelling the data. It is also why the algorithm failed to put a partition around point 30, as well as making an unreliable prediction in the middle region.

Nonlinear Conjugate Gradient Method

Finally the NLCGM presented an extremely bad modelling of the data in Figure 4.17. Except in the first region, the confidence intervals in other regions are so large that they exceeded the bound of the plot. It is caused by an extremely bad t value. For the NLCGM, the starting point of searching for t is arbitrary rather than a fixed 1 in the QNM. Thus the search for t is even more complex than in the QNM. Considering so many drawbacks in using Wolfe conditions, nothing really could be done to remedy the problem; because it is a strict inexact line search problem in the absence of the explicit Hessian matrix for the QNM and NLCGM. A direct approach for the NM method will not be attainable in either QNM or NLCGM. Apart from Wolfe conditions, one can still apply backtracking line search which is based on the Armijo-Goldstein condition [63]. However the Armijo-Goldstein condition is a special case of Wolfe conditions, it shall not work either. Hence the QNM and NLCGM must be abandoned, which makes the Newton method the only valid numerical optimisation in this thesis.

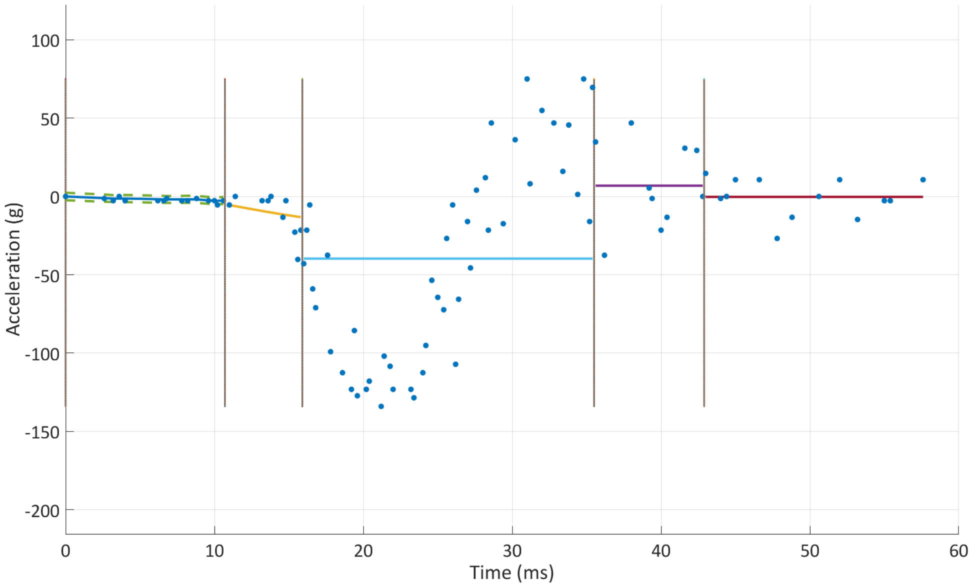


Figure 4.17: Failure of the NLCGM on the MAD.

4.4 Benchmarking of the computational cost

The computational expense is always among the chief concerns for any mathematical modelling subject to practical implementations. In the context of the TGP, the computational cost mainly consists of two parts: the sampling rate and the sampling size. In the TGP, the sampling rate describes how fast the algorithm produces a new state of the tree in the Markov space. The sampling size is the necessary number of samples required to find the MAP of the tree.

At the beginning of this chapter, it has been mentioned that the GTGP has extremely high sampling rate. During one second running on the MAD, the GTGP could evaluate approximately more than 100 candidate trees (depends on the complexity of the tree) and produce approximately 10 valid MCMC states of the tree.

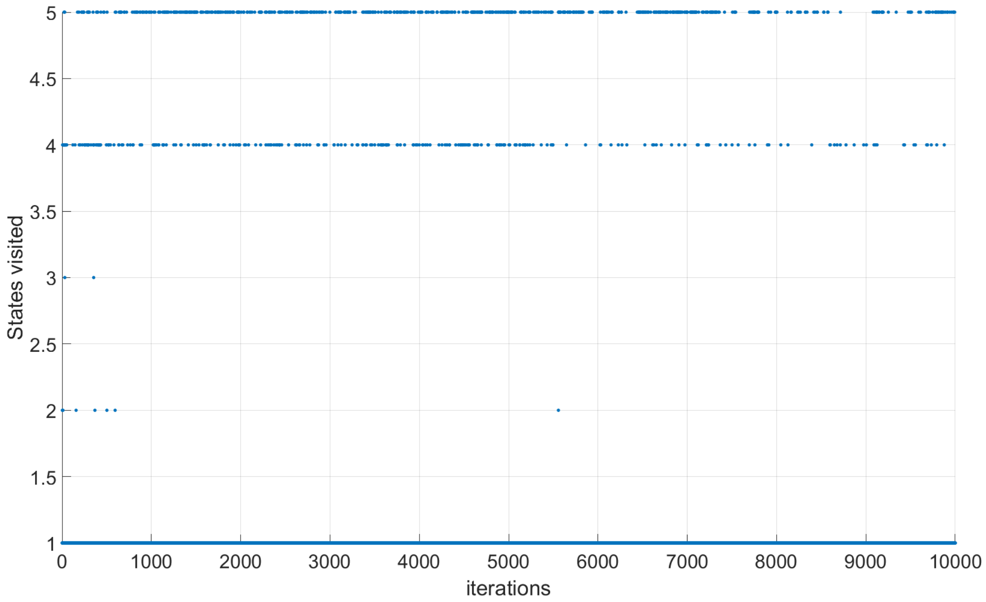


Figure 4.18: GTGP tree alteration history during 10000 MCMC steps.

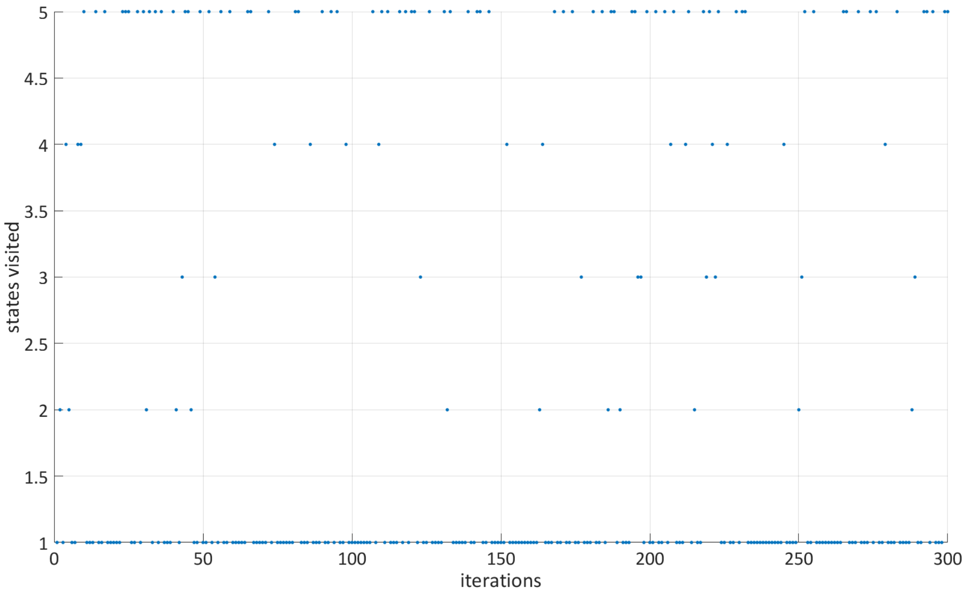


Figure 4.19: CTGP tree alteration history during 10000 MCMC steps.

Figure 4.18 shows the typical tree alteration history during 10000 MCMC steps. The numbers on the Y axis have no quantitative meaning, they represent the 4 tree alteration executions plus the rejection of the tree: 1=*Rejection*; 2=*Grow*; 3=*Prune*; 4=*Change*; 5=*Rotate*. In this generalisable particular figure, out of 10000 MCMC steps, 8889 times a proposed new tree state will be rejected, which gives a rejection rate about 89%. The Figure 4.19 shows the same plot with regard to the CTGP optimised stochastically in 300 MCMC steps. It is also generalisable to the CTGP optimised numerically by the NM, since the NM gives similar posterior as the stochastic process. It has been found the rejection rate for the CTGP is 66.8%. Provided that 300 MCMC steps take on average 400s for the stochastically optimised CTGP to run, thus for one second the stochastically optimised CTGP can produce 0.25 valid MCMC trees. For the NM optimised CTGP whose 300 MCMC steps approximately take 130s, in one second it could produce 0.8 valid MCMC state. Overall the GTGP has a sampling rate at least 12 times faster than the NM-optimised CTGP and 40 times faster than the stochastically-optimised CTGP. In fact, it is comparatively straightforward to improve the sampling rate for the CTGP, since the high computation cost is fully commanded by the evaluation of the GPML with the exhaustive optimisation involved. For the stochastically-optimised CTGP, the current default choice for the number of samples in the optimisation is set to 900 for all three hyperparameters. Figure 4.20 as a typical example, shows that posterior convergence occurs normally before 300 iterations, thus it is possible to theoretically save 2/3 time for the stochastically optimised CTGP. For the numerically-optimised CTGP, the stopping criterion could be slightly slackened to allow the optimisation search to stop at a higher gradient. However by doing so for either CTGP, the hidden risk of missing the convergence will rise. Other methods for optimising the computational cost could also be found under the title of efficiently handling matrix manipulations, such as using the Coppersmith algorithm for matrix inversion [64] etc. Such areas are rather broad and also require specialised skills in Mathematics.

For the sampling size, it has been reiterated for several times in the former content, that the GTGP requires a much larger sampling size to glean sufficient number of trees to find the MAP. It has been found that the required size is around 30000 MCMC steps for the MAD (30000 is a conservative number, normally 20000). It will take approximately 95s for the GTGP to run through. The sampling size for the CTGP is around 300 MCMC steps. Although being considerably smaller in the sampling size, the CTGP with the stochastic optimisation still requires over four times the running time to achieve a steadily converged state of the posterior (the GTGP does not converge, but 30000 steps is more than enough to hit that posterior). Superficially say, it is because the sampling rate for the CTGP is much lower than the GTGP. There is deeper insight than such from direct observations. In fact it is an inherent deficiency for the stochastically-optimised CTGP that its sampling efficiency is substantially lower than the GTGP if its mono-sampling scheme is con-

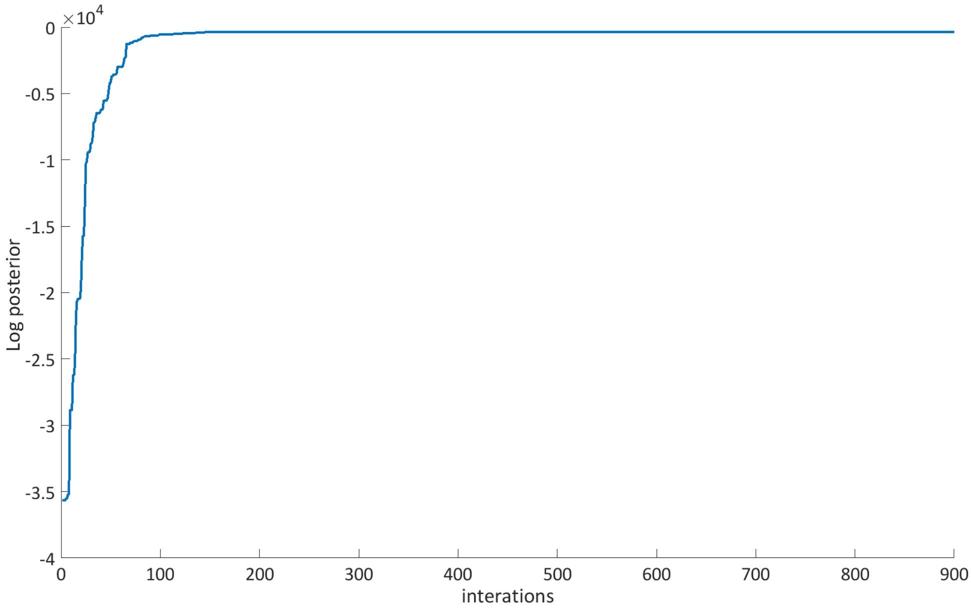


Figure 4.20: The posterior convergence in the optimisation of the hyperparameters.

sidered as a mono-sampling scheme with subordinate samplings. The optimisation process for hyperparameters is aimed at functioning as an exhaustive search. Thus at any tree in the Markov space, the subordinate samples of the hyperparameters is subject to peremptory enforcement. This leads to a problem that if the current tree is a rather bad one, what is the point of doing the same number of sampling for a tree that has no further contribution to the search for the MAP (bad trees will be rejected from the MH check, only good trees can remain their posteriors). Such a fact also means the 2/3 rejected trees, in terms of computational cost, are accompanied with great waste. While it could be said that the GTGP is parsimonious, where every bit does count in the sampling space (In the GTGP, drops in the posterior are a natural way to escape local minima). The CTGP with the NM optimisation also suffers from such unnecessary optimisations in unpromising trees. However the harm incurred is not as heavy as in its stochastic counterpart, where the stochastic optimisation is much more expensive.

4.5 Conclusion

The MAD, as expected, is made full use of in its statistical characters to successfully form a benchmarking ground that highlights the differences in the GTGP and the CTGP. The performance of the GTGP and the CTGP are both reasonable enough to demonstrate their capability in dealing with the data containing heteroscedasticity. Their nuance in the sensitivity to the variance and trend switching, signifies their characters well. Not a general bad or good, nor worse or better could form a solid and unanimous verdict for either model. The CTGP does hold some advantages like ensured convergence, stable final prediction etc. However neither of these advantages is absolute without also bringing some disadvantages for the model. Likewise the hugh advantage in the computational cost on the side of the GTGP is also a double-edged sword which debases its predictive reliability. It is frustrating to see the failures of the QNM and NLCGM optimisations with Wolfe conditions involved, although the cause of failure has been spotted. At last in summary, the following figures give a qualitative comparison and contrast between the GTGP and the CTGP in terms of various criteria (DOF stands for Degree Of Freedom).

Models	Accuracy	Stability	DOF
GTGP	High	High	High
CTGP (Stochastic)	Very High	High	Low
CTGP (Numerical)	High	Very High	Very low

Table 4.1: Comparison base on performance

Models	running time	sampling rate	sampling size
GTGP	Very fast	Very fast	Very large
CTGP (Stochastic)	Slow	Very Slow	Small
CTGP (Numerical)	Slow	Fast	Small

Table 4.2: Comparison based on computational cost

Chapter 5

Experimental measurement and data analysis on the Particle damping

5.1 Chapter Overview

This chapter is mainly about the configuration of the experiment and the discussions on the results. It starts from justifying the measurement objects, the loss factor. Then the experiment configurations and procedures will be explained followed by the analysis of the experiment results. The measurement theory and other issues concerned with data post-processing will also be included along the way.

Measuring the loss factor η is a common way to assess the damping performance of a vibrating system. For particle damping, the literature regarding such measurements is large. The traditional measurement for loss factor is to perform a standard FRF analysis on the target system, by equivalently treating it as a SDOF (Single Degree Of Freedom) spring-damper system. However, since the damping property for a particle damping system is not constant throughout varying the vibration conditions (e.g. change the excitation power or frequency), thus such an approach requires the measurement to be taken close to the resonant frequency so as to enhance its reliability. Poising against such a drawback from the classical damping test, Yang [65] proposed a power-based approach to measuring the damping, which is used in this project.

The damping property in a particle damping system has a rather wide range of influencing factors, which could generally be classified into internal and external factors. The internal factors refer to the class of factors remaining unaltered during the experiment, such as filling ratio, material friction and mass. The external factors are basically the factors subject to alteration during the experiment, such as vibration amplitude and frequency as a result of change of excitation power and frequency. For the current consideration of the experiment, the chief concern is focused on the variation of the damping at different levels of vibration frequency and amplitude (the amplitude of the acceleration is studied, thus the acceleration used in the following content refers to the amplitude of the acceleration).

5.2 Quantification of Damping

To measure the damping performance of a particle damping (PD) system, is the chief experimental focus here. The damping is a quantifiable property which could be generally reflected in the following three measurable quantities [66]:

- resonance amplitude reduction,
- incremental decay of free vibration,
- cyclical energy loss in forced vibration.

Thus a suitable quantification for the damping must show capability of fitting well with all the three quantities above. The two most prevailing quantifications for damping are the damping ratio (ζ) and the loss factor (η).

The damping ratio is a conventional description for the damping of a Single Degree of Freedom vibration system of the viscous type. It is taken as:

$$\zeta = \frac{c}{c_c} \tag{5.1}$$

where c is the damping coefficient or actual damping and $c_c = 2\sqrt{km}$ is the critical damping, k is the stiffness and m is the mass. To use such a quantification, the system should be approximately regarded as a SDOF vibration and a general viscosity also should be assumed. Therefore, using damping ratio as a damping measurement is rather restricted.

Using the loss factor as a general damping measurement gains a prevailing popularity tracing back decades [67]. The tenability of using the loss factor is premised by an a priori assumption of viscoelasticity. The loss factor was initially introduced as,

$$\eta = \frac{E''}{E'} = \tan(\phi) \tag{5.2}$$

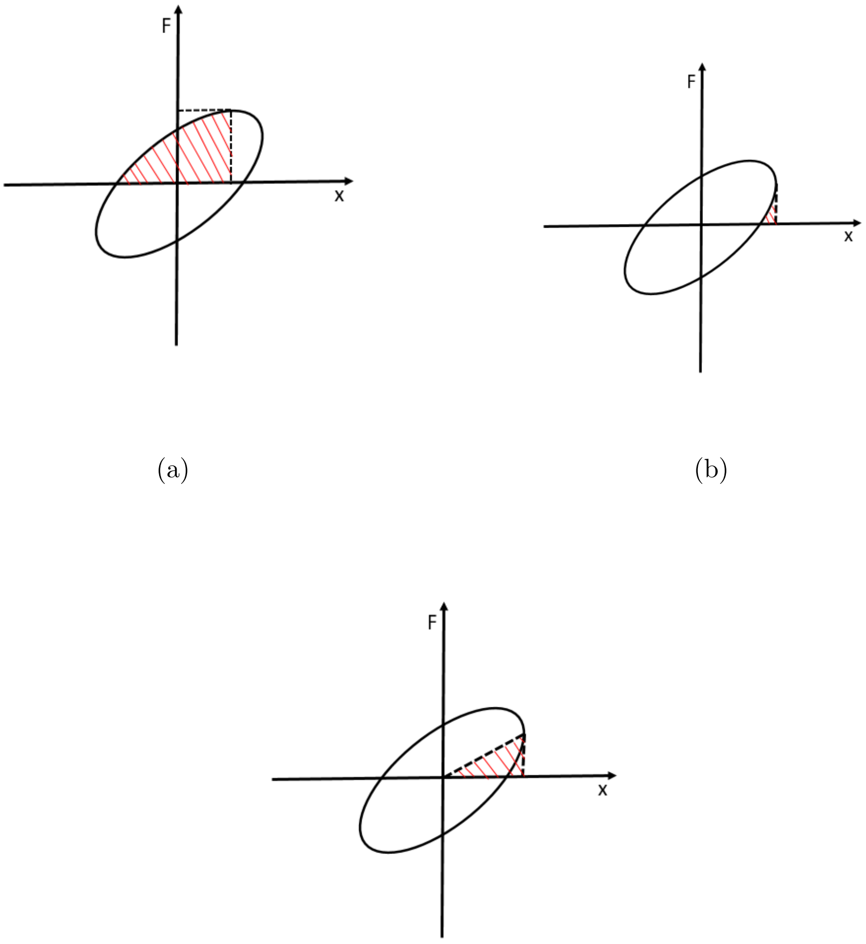
The equation above describes the loss factor as a ratio between two moduli E'' and E', which are both intrinsic properties held by the material. For more explanations of the symbols, one could refer to ISO standard 6721 [68]. For more intuitive understanding of the loss factor, it is an alternative to associate loss factor with the energy theory,

$$\eta = \frac{D}{2\pi U} \tag{5.3}$$

where D is the energy dissipation per oscillating cycle and U is the stored vibration energy per oscillating cycle in the system. In fact the term *vibration energy* is too general to describe the energy of vibration, and thus provides little insight into why the loss factor is a suitable general measure for damping.

To intuitively explain the energy composition for vibration energy U, it is worthwhile to introduce the idea of a hysteresis loop. The hysteresis loop itself is not exclusive to the field of dynamics, it is more widely recognised and used in the theory related to magnetic engineering. But the usage in each case has a common conceptual ground. Hysteresis means delay and retention, thus such loop is used to describe a physical behaviour with a dependence on the history of its states. In dynamics, since universal materials are in fact an assembly from micro-structure of atomic alignments, it is inevitable to concede energy loss during deformation as a result of friction in micro-structural displacement. Because deformation absorbs energy, the strain of the material will be delayed to reach the value corresponding to the stress in action, this phenomenon is called hysteresis. It is well known for a spring-damper system; the hysteresis loop can be approximately treated as an ellipse below on the force-displacement Figure 5.1,

According to Lazan's pioneering work [67], he suggests to define the U as either of



(a)

(b)

(c)

Figure 5.1: Three types of definitions for the loss factor (for text description, see below).

the following three definitions, each of which corresponds to the shaded area in the hysteresis loop in Figure 5.1:

(a) energy stored during loading (stress from zero to maximum);

(b) energy released during unloading (maximum strain to zero stress);

(c) maximum deformation energy stored (elastic energy).

For lightly damped systems ($\eta < 0.1$), the three quantities hold well in balance, and are close in value. For less lightly damped systems, typically with the involvement of elastomers, drastic differences in value could invalidate the alternatives in defining U from the three, and one must be selected. Over decades of investigations, the third choice above gained a prevalence over the other two. The definition (c) is a static energetic quantity which solely depends on the elastic strain energy stored during the loading. Therefore, under such a definition, the U could also be effectively considered as the total energy,

$$U = \frac{1}{2}k'X_0^2 \tag{5.4}$$

Lazan [67] also shows that, if the governing behaviour is linear, the stiffness could be defined as a complex term to describe the viscoelastic damping system equivalently as a classic simple spring-damper system. Henceforth, the following derivation could be established,

$$\eta = \frac{D}{2\pi U} = \frac{\pi k'' X_0^2}{2\pi \frac{1}{2} X_0^2 k'} = \frac{E_{dissipated}}{E_{maximum}} \tag{5.5}$$

The above derivation successfully shows a union of equations (5.2) and (5.3) through the scope of energy dissipation, though they are grounded in two separate initial definitions. such intersection of definitions reflects the generality of the loss factor η as a common measure for the damping.

5.3 The Damper's Configuration

The last section fortified the choice of measuring the loss factor as the experimental objective. The next stage is to establish the concrete experiment plan. The paramount issue here is not a selection of the specific material, nor a design of the damper's shape and etc. In fact recalling Saluena [5], the classic particle damping is characterised by three phases: solid, liquid and gas, which itself is an intrinsic property of particle damper. It is natural to expand generality of the experiment by covering all the three phases (Figure 5.2). But one crucial point worthy of attention is that, Saluena's paper has generalised the three phase characteristics of the particle damping, but not Figure 5.2. Figure 5.2 only applies to Saluena's experiment where the PD was tested horizontally.

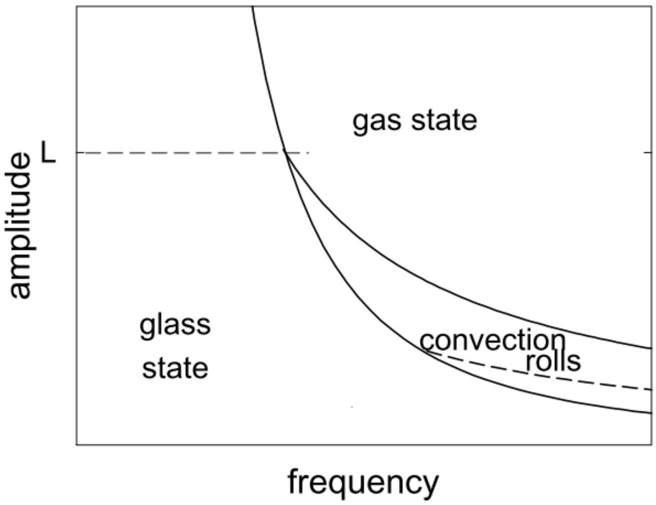


Figure 5.2: Phase map for particle damping re-plotted for fast referencing.

To facilitate the fulfilment of such a purpose, a simple dome shaped damper with a spherical cavity made from polycarbonate (Density: $1200kg/m^3$) is used. Because polycarbonate is transparent, such a choice of material is generally for the purpose of observing the behaviour of the particles during the test.

To cover the three phases, the test will be conducted in a frequency-guided fashion. Measurements will be taken from 50 to 500Hz incremented by 25Hz. At each frequency, the acceleration amplitude will be increased uniformly to cover all three phases which are discerned from visual observation on the particle bed.

Although damping could be quantified by the loss factor, the loss factor is not a physical property subject to direct measurement like speed, temperature etc. It is more abstract, since it is an embodiment and also a related calculation of power dissipation which associates with force and velocity. Therefore, it is natural to choose a force transducer and laser-based velocity measuring sensor to take measurements of the aforementioned two quantities. In vibration tests, since the motions are commonly sinusoidally or co-sinusoidally characterised along with the time, there are extremely simple mutual conversions among the displacement, velocity and acceleration through differentiation and integration. It follows that either a measurement of displacement or a measurement of acceleration could be used in lieu of the velocity measurement. Circumstantially, the accelerometer gains a preference over the other two. Because the measurements of displacement and velocity both use laser equipments; however, the vibration amplitude shrinks drastically with the increase of the frequency for a given fixed power supply, which means more resolution in the laser equipment is required to capture the details of the motion. Frankly, such high resolution laser meter is available on the market, but that cost massive amount of money. For parsimonious reasons, the accelerometer is chosen as the chief motion capture sensor. The laser is still mounted to measure the velocity under 100Hz, which serves as a supplementary measurement to validate the acceleration measured. Figure 5.3 shows a sketch for the damper's configuration along with its measuring units attached. Figure 5.4 shows the specific dimensions of the cylindrical and domed parts of the damper. The total weight of the damper filled with particles (The particle material is discussed in the next paragraph) is 167g.

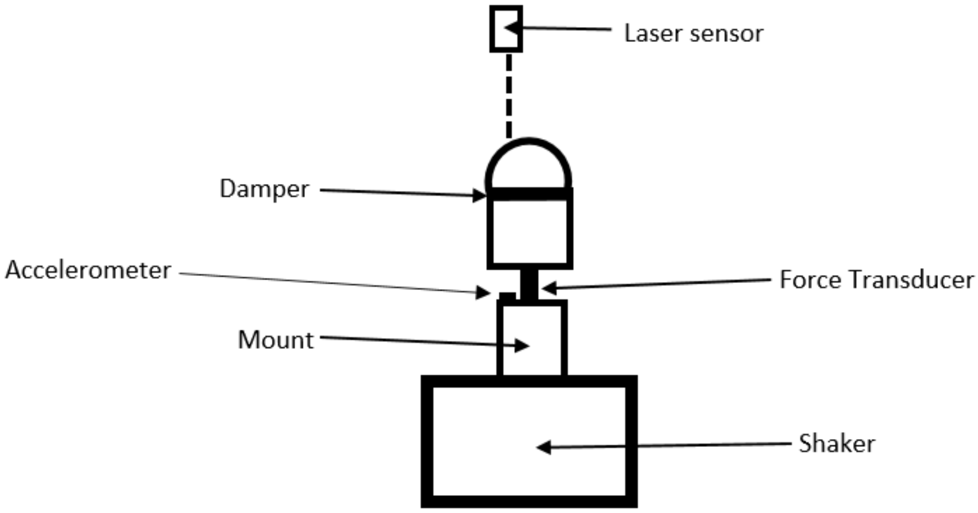


Figure 5.3: Configuration of the damper's measurement system.

The choice of particle material will be the simple spherical steel particles (Density:$8050kg/m^3$, diameter: $2mm$). The spherical steel particles entail a damping system where the energy will be predominantly dissipated through the friction between particles. It is a conservative choice without adding too much complexity into the model such as large deformation energy absorption from rubber particles, or high uncertainty

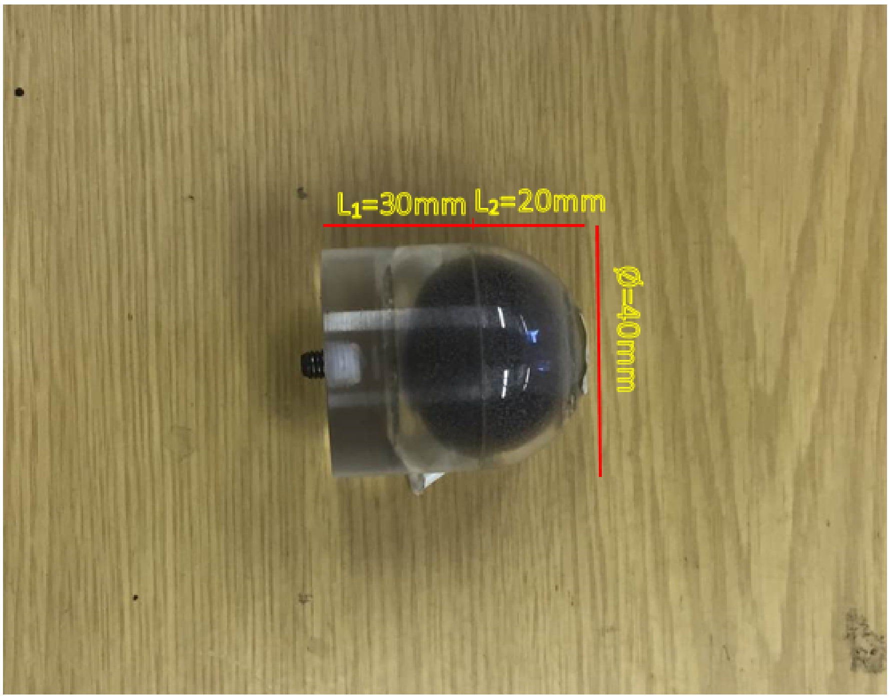


Figure 5.4: testing damper with steel particles.

from irregular particle shapes, etc.

The filling ratio will be around 90%, which intends to allay the complexity potentially introduced by chaotic collisions in the gas state of the particle bed.

5.4 Experiment Planning and Components

In some vibration tests, to maintain the sinusoidal form of the motion is essential, because the subsequent processing of the data, such as computing the power dissipation, is built on the assumption of ideal trigonometrical behaviour of the motion. To achieve the enforcement of sinusoidal behaviour, a control loop could be designed to monitor the measurement. Because a control loop is a global design for the measurement systems, it also partially determines the composition and choices of test components. The design of control systems is rather multifarious. The simplest way, and also the most cost efficient, is to use the Proportional-Integral-Derivative (PID) controller. A PID control loop, like other control loops, is an input-output monitoring process where the output signal is subject to adjustment, such as truncation and reparation, based on the output feedback to meet the pre-set level.

The PID controller requires a manual tuning of its three parameters (P,I,D) to achieve a decent performance. Such a procedure is interfaced by Labview^{TM} in the PC. The detailed tuning rules are included in the later sections.

Taking a global look at the effect of incorporating the PID control system into the test on particle damping, it consequently results in a measurement system which can be depicted in the block diagram in Figure 5.5.

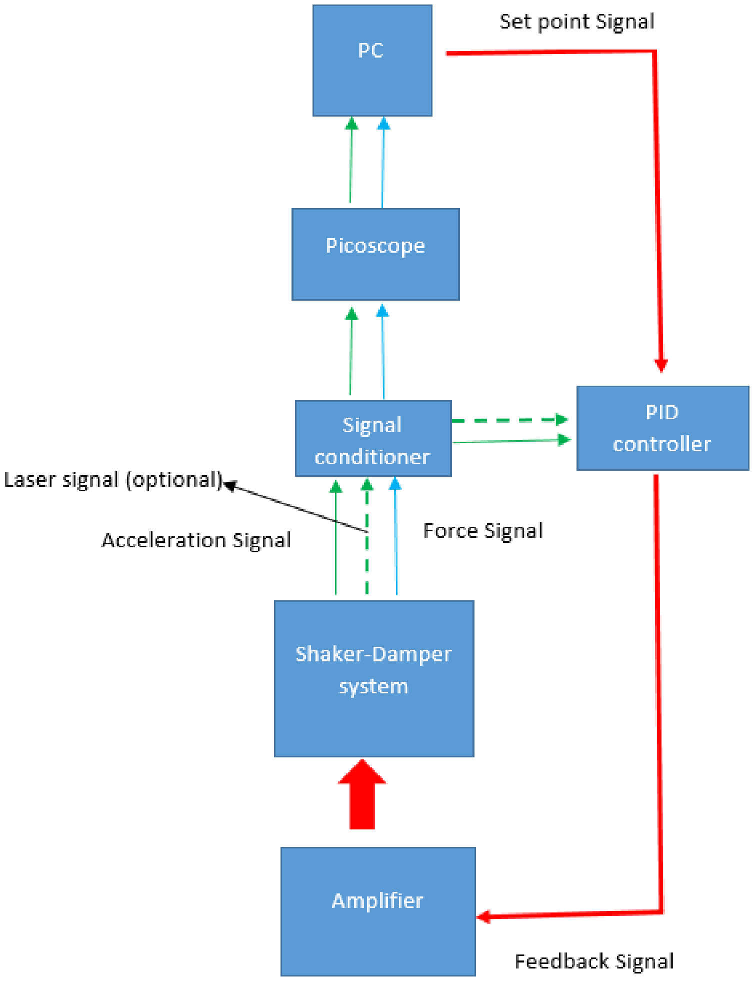


Figure 5.5: Test block diagram.

Most of the equipment involved in this experiment is typical in vibration-based tests:

The shaker (Figure 5.6 [a]) is used for imposing direction excitation to the damper.

It is the direct source of vibration.

The power amplifier (Figure 5.6 [b]) is directly connected to the shaker to magnify the level of vibration. It supplies the high-current signal to drive the shaker.

The signal conditioner (Figure 5.6 [c]) for an accelerometer (sensitivity: 9.5mV/g) or force transducer (sensitivity: 21.67mV/kN) changes the electrical displacement, generated when the piezoelectric plates in the sensor are deformed under vibration, into a voltage signal that is measurable on a scope. The laser signal conditioner is a very different device. It depends on the type of laser you are using. If it is a velocity laser, the changes the doppler signal read by the light sensors into a voltage that is proportional to velocity of the measured object.

The Picoscope^{TM} (Figure 5.6 [d]) is a data acquisition device used in pair with its companion software on the PC. In this way, the PC works as an interface to supervise the real time data acquisition, and meanwhile send set point data to the controller.

5.5 Measurement Procedures

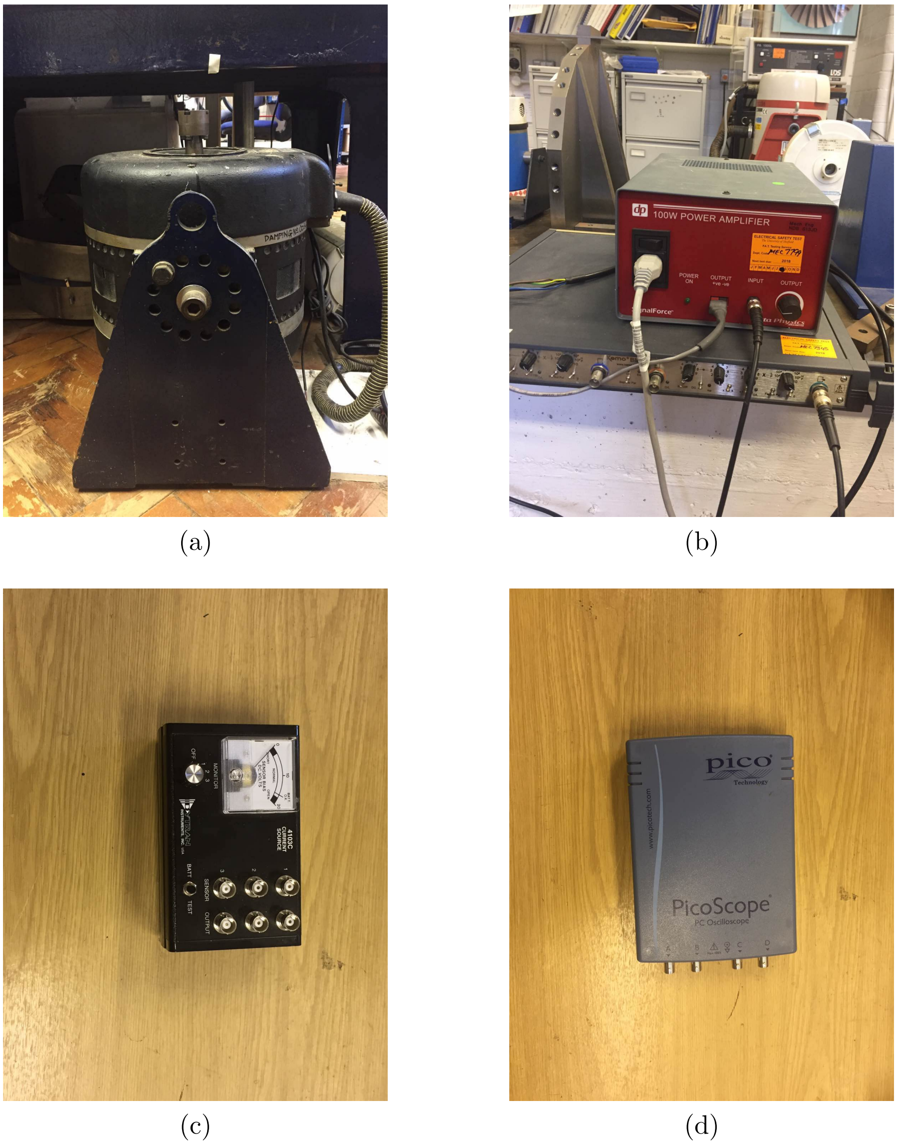
Following the test objective, which is to measure the damping at different levels of frequency and acceleration, the procedure could be macroscopically encapsulated as:

1. Choose a frequency for testing
2. Take measurements of the force and acceleration (velocity measurement from the laser is optional) signals at different levels of acceleration amplitude based on controlled acceleration under that frequency

2.1. At the beginning of each measurement, set a set point for the acceleration amplitude to be measured.

2.2. After the set point is defined, tune the PID parameters in Labview to ensure a reaching of the set point level and the maintaining of a sinusoidal form for the acceleration measured.

2.3. During each measurement, allow the Picoscope to record data for 5s after the signal is stabilised from the effect of tuning and settling.



(a)

(b)

(c)

(d)

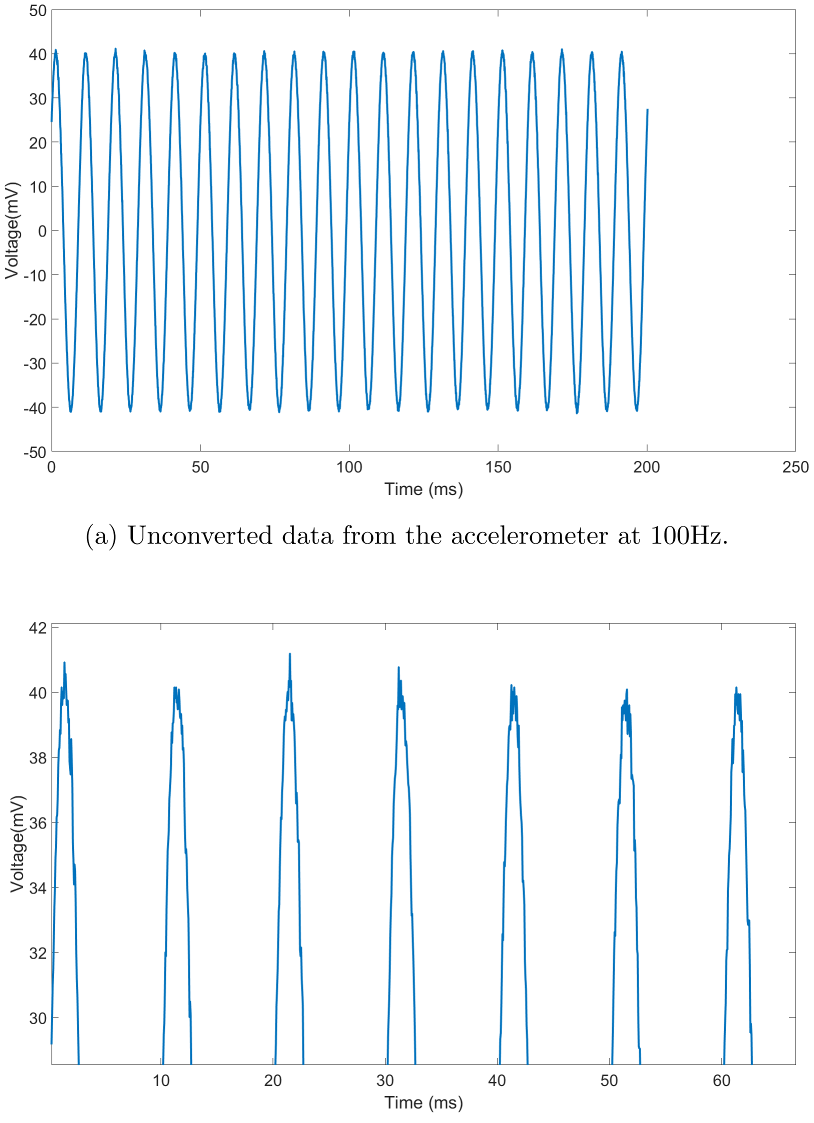
Figure 5.6: Test components:(a) Shaker (b) Amplifier (c) Signal conditioner (d) Picoscope.

3. After the set of tests done for a given frequency, increase the frequency by an increment of 25Hz and repeat from step 1.

5.5.1 The Use of the Butterworth Filter

It is rather a pity that the current PID control system failed its fulfilment to maintain a sinusoidal acceleration with a good quality, thus it eventually had to be abandoned. It is not a strict failure due to irremediable inherent defects of the controller itself. In fact, it is more of failure due to impracticability. The basic PID control requires a manual tuning of the parameters. The general procedure for tuning the PID is simple. The main operation to take is to increase P slowly to approach the set point, if any overshoot takes place, increase the I, and if the convergence collapses as out of control, stop the process and increase D, then increase P from 0. Although it is theoretically viable to achieve the correct performance through such a tuning process, the problem ingrained here is that such a process is highly uncertain and inefficient in the particle damping test. There are a number of issues which can be addressed as causes. Primarily the tuning process is highly sensitive to the change of P value. This yields a frequent loss of control of the acceleration level which can be manifested by a divergence from the set point. If the D is increased to counteract the negative effect of divergence from a large P, the convergence process will be substantially increased. Another potential problem is from the nonlinearity of the system. The control of the particle damping system is in general not a simple linear control, because the controlled quantity (e.g. acceleration) is also an influencing factor to the damping which reciprocally affects the acceleration itself. Such fact will introduce complexity in the tuning process and also reduce the effectiveness of changing the PID. Through many sets of tuning, it turns out that it could often take more than 1 hour to tune the PID for one measurement at a given amplitude and frequency. There are also cases where it does not seem to converge for any set of PID values. In this sense, the original plan of PID control has to be abandoned; but the experimental design of the measurement in Figure 5.5 does not need to alter, because even though the PID no longer functions, the loop still could be treated as an open loop system where the amplifier directly magnifies the uncontrolled signal from the PC and sends it to the shaker for excitation.

Without an effective control, the measured acceleration will present a distorted sinusoidal profile which approximates a sinusoid but resembles rather badly in the vicinities of the peak. Figure 5.7 illustrates the distortion in the sinusoidal signal. It can be observed from the figure, that in the vicinity of the peaks, the desired sinusoid is distorted by the irregular presence of vertical spikes. This issue is mainly caused by the impact between the particles and the damper's inner wall as a result of inertia de-synchronisation.

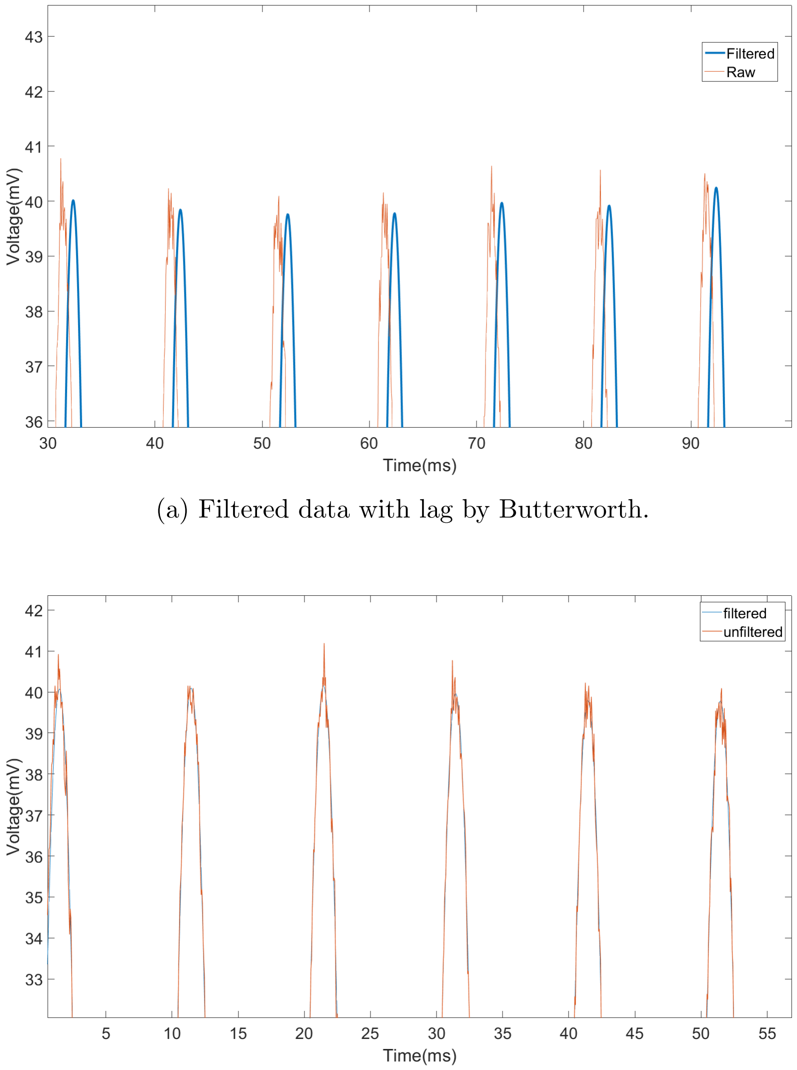


(a) Unconverted data from the accelerometer at 100Hz.

(b) same plot zoomed at the peak.

Figure 5.7: Unconverted data from the accelerometer at 100Hz.

Even though direct PID control does not function satisfactorily, it is still possible to apply post-processing on the measured data to expediently get a clean measurement. Here a Butterworth filter is designed to eliminate the high frequency response from the data. Because the spikes are rather dense compared to the general sinusoidal trend, an appropriate low-pass filter should be able to obtain a clean sinusoid at the main oscillating frequency without suffering too much from the residual effect from the frequency cut-off process. The residual effect refers to the inevitable loss over the entire frequency span due to the cut off. After applying the Butterworth filter, Figure 5.8 could be plotted.



(a) Filtered data with lag by Butterworth.

(b) Zero lag filtered data from modified Butterworth.

Figure 5.8: Eliminating the phase lag induced by the Butterworth filtering.

Figure 5.8(a) shows the filtered data against the raw data through direct application of the Butterworth filter. However, it could be easily captured that after processing the data through the filter, though the redundant spikes have been culled, there has been a development of a phase lag relative to the raw data. Phase lag is a typical phenomenon not exclusive to Butterworth filter, but ubiquitously dwells in general filters, amplitudes and etc. To eliminate the phase delay, one could pass the signal twice both forward and backward through the filter to correct the phase delay. The Figure 5.8(b) shows the outcome from a zero-lag Butterworth filter.

5.6 Yang's Approach

In the chapter overview, it has been briefly addressed that Yang's power-based approach has been used in calculating the loss factor. In Yang's theory, the average power flow is estimated as:

$$P = \frac{1}{2} F_{pk} V_{pk}^{*} \tag{5.6}$$

Where F and V denote the complex force and velocity respectively, the index *pk* stands for peak value, V^* is the complex conjugate of V.

In harmonic motion, the term root-mean-square is commonly used, which corresponds to $1/\sqrt{2}$ of the peak value of the harmonic signal. Therefore the equation above could also be written as:

$$P = F_{rms} V_{rms}^{*} \tag{5.7}$$

Recalling that the loss factor could be defined as a ratio between two complex energy terms (5.5). Both the dissipated energy and the maximum energy stored in the system on a cyclical basis are the real part and the imaginary part of the complex power respectively. In this sense, the loss factor eventually could be expressed in terms of all the measured quantities.

$$\eta = \frac{Real\{P\}}{Imag\{P\}} \tag{5.8}$$

5.7 General Data Analysis

By applying Yang's approach to the acquired data, the following 2D contour plot for the loss factor η against both acceleration and frequency could be obtained, as shown in Figure 5.9.

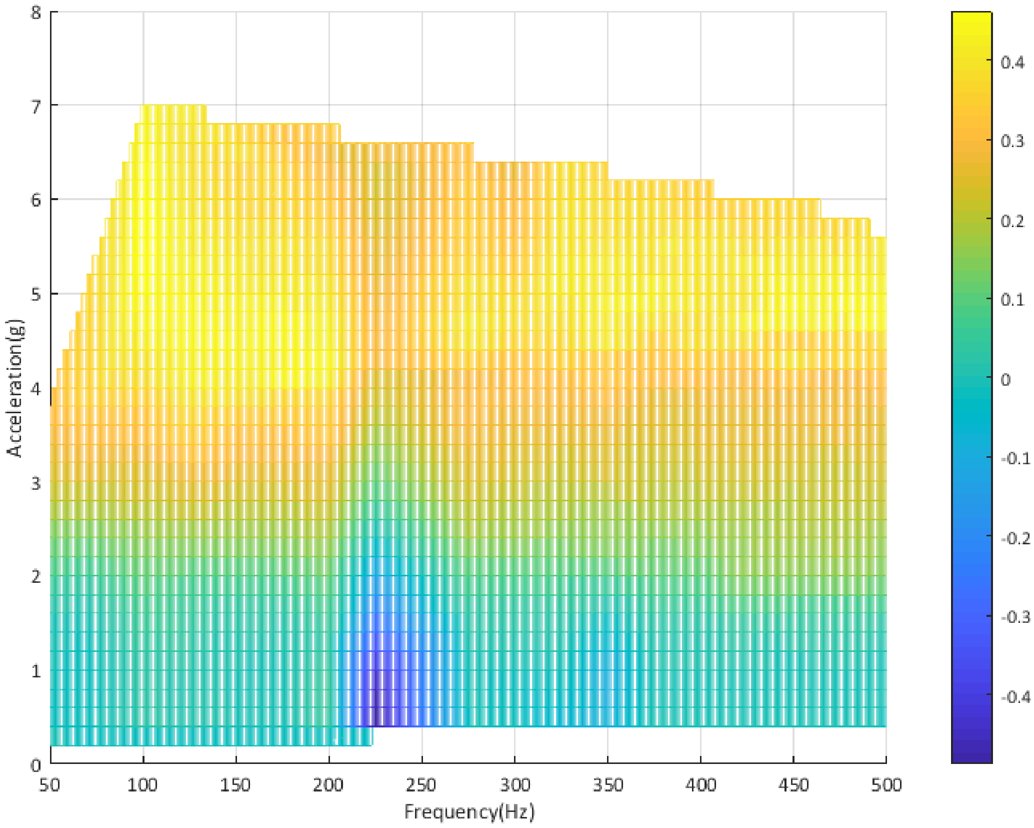


Figure 5.9: contour plot for the loss factor.

First, it is necessary to explain the shape of the plot. It appears that the measurements taken at each individual frequency have been bounded in different ranges of the acceleration amplitudes. For the two tests at 50Hz and 75Hz, the maximum level of the acceleration amplitude has been limited to under 5g which forms an evident contrast against the rest of the tests at higher frequencies. The reason for this is related to the physical insight of the experiment. At lower frequencies, to generate the same amount of shaking level as represented by the vibration displacement amplitude, less power is required when compared to those tests at higher frequencies. If the excitation acceleration is higher than 5g, the actual vibration displacement amplitude will be so strong as being visually observed. This could on the one hand be a potential threat to damage the damping component and on the other hand introduce more interfering horizontal vibration into the system.

When it enters the frequency range over 100Hz, there is a steady descending of the acceleration amplitude with the increase of frequency level. Such an issue is a consequence of the limitation in the power supply. Under the current combination of shaker and amplifier, the power can only drive the damping system to reach a level of acceleration amplitude as shown in the graph at each frequency. Since more power is required to reach the same level of excitation with the increase of frequency, the end point in the acceleration amplitude at each frequency will decrease gradually as the frequency increases if the power cannot be increased.

Another rather interesting observation from the plot is the heavily negative loss factor concentrated at the bottom around 225Hz (as represented by a blue region. from the colour bar); the concentrated blue at the regional centre corresponds to a value as low as -0.4, which is definitely an uncommon value for loss factor and in a sense could be an indication of erroneous measurement. Several sets of repetition have been done for this particular frequency range, they all show agreement to such a strong negative value. Because a negative loss factor indicates an introduction of energy into the system, the real part of the complex power, which represents the energy dissipation, must be negative. The composition of the real part of the complex power by definition could be expressed as,

$$Real(P) = Real(F_{rms}) \times Real(V_{rms}^{*}) + Imag(F_{rms}) \times Imag(V_{rms}^{*}) \tag{5.9}$$

On the right hand side of the equation, the first term in the expression of a real part multiplication, represents the actual power flowing into the damping system. The second term for the imaginary part multiplication is somewhat abstract at its physical meaning. It could be effectively considered to be the mixture of induced strain energy and kinetic energy as a result of the application of external force. To get a better insight into the relation, it is possible to make a better intuition by looking at the Frequency Response Function:

$$\frac{V_{rms}}{F_{rms}} = \frac{i\omega}{k - \omega^2 m + i\omega c} \tag{5.10}$$

Such a relation gives the opportunity to represent F_{rms} or V_{rms} with respect to one or the other. Thus,

$$F_{rms} = V_{rms}(c + \frac{(\omega^2 m - k)}{\omega} i) \tag{5.11}$$

Hence the energy dissipation term 5.9, after some mathematical manipulations, becomes extremely simple (detailed derivation, see the appendix):

$$Real(P) = c(Real(V_{rms})^2 + Imag(V_{rms})^2) = c|V_{rms}|^2 \tag{5.12}$$

The expression above implies that the energy dissipated out of the system is linearly related to the magnitude of the velocity, if all conditions and assumptions hold well. Here the requisite premise is that the equation (5.10) must hold. The FRF relation itself is a linear relation specifying that the ratio between the magnitude of force F and velocity V must be a constant under the circumstance of fixed k (effective Hooke's constant), c (effective damping coefficient), m (effective mass) and ω. As in particle damping, different vibration levels in terms of acceleration amplitude and frequency yields different k, c, and m, the ratio between force and velocity will not be a constant. At low amplitude excitation, the system can be effectively treated as a linear spring-damper system, where the linearity holds between F and V if the frequency is fixed. The expression is as follows,

$$F/V = \sqrt{c^2 + \frac{(\omega^2 m - k)^2}{\omega^2}} \tag{5.13}$$

Because in most of the particle damping, c and k, especially at low excitation level, will be substantially lower than $\omega^2 m$, the relation above could be reduced to $F/V = \omega m$. Therefore, if the excitation level is low, the ratio F/V will solely depend on the frequency (effective mass m can be approximately treated as constant). This gives theoretically, the frequency vs F/V plot should give a straight line at a given fixed amplitude. At an excitation level of $0.6g$, from 100Hz to 300Hz, the plot of frequency vs F/V gives Figure 5.10.

By observing Figure 5.10, it is easy to detect an underestimation deviated from the linear behaviour at 225Hz, which indicates that at 225Hz the same force can induce more response in terms of velocity than expected. In addition, its neighbouring frequencies of 200Hz and 250Hz are both only minimally influenced. Thinking an amplifying effect concentrated around a certain frequency is very commonly observ-

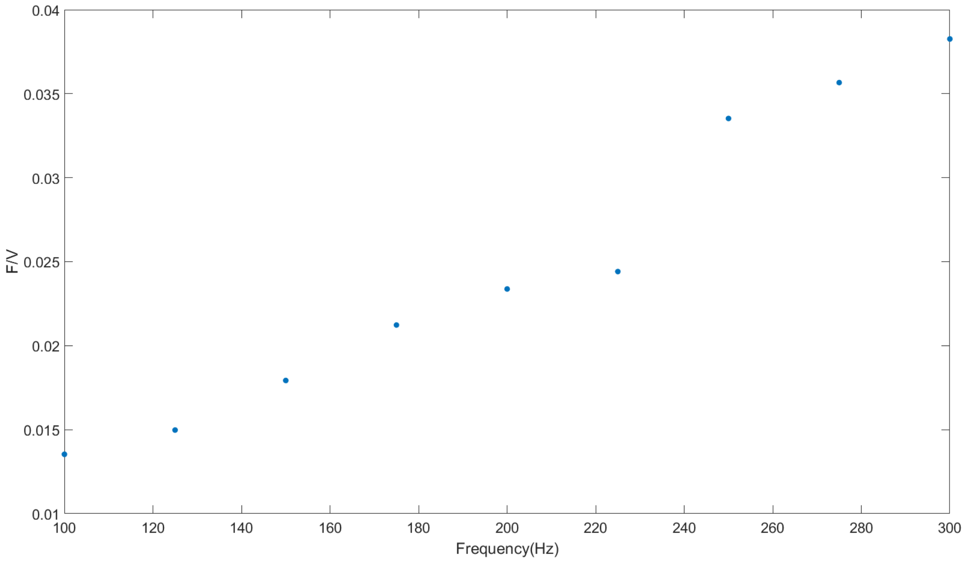


Figure 5.10: Excitation frequency vs F/V at 0.6g.

able in the event of resonance, therefore, the probable cause could be very likely linked to a resonance in the system such as a resonance in the amplifier caused by the phase lag in the sensor. However, to locate the exact cause requires a separate dedicated experiment which can be included in the future work of this project. Despite the lack of the knowledge of the exact cause, since negative loss factor is undoubtedly an unacceptable result, the entire test at 225Hz will be completely nullified.

After culling the nullified data, as the colour bar indicates, the highest loss factor as induced by the vibration is in general between 4g and 5g. Figure 5.11 shows the 2D plot for the loss factor directly against the acceleration amplitude with selected frequencies stacked on the plot. The frequencies lower than 100Hz are removed for lacking of enough data points. Above 100Hz, frequencies in every 50Hz are plotted for the purpose of clarity. It is much easier to perceive a picture of the general trend of the curve. The figure shows that at low acceleration (0-1.5g), the loss factor is in general mildly at a constant level at each set of frequencies. Such fact evinces at the low excitation level, everything in the system could be approximated properly as a simple linear spring-damper system. According to the visual observation, all the tests across the entire frequency span did remain in the solid state within excitation level (0-1.5g), where most of the particles remained stationary and only very few in slight shaking (at top) or rolling (below top) can be sporadically observed.

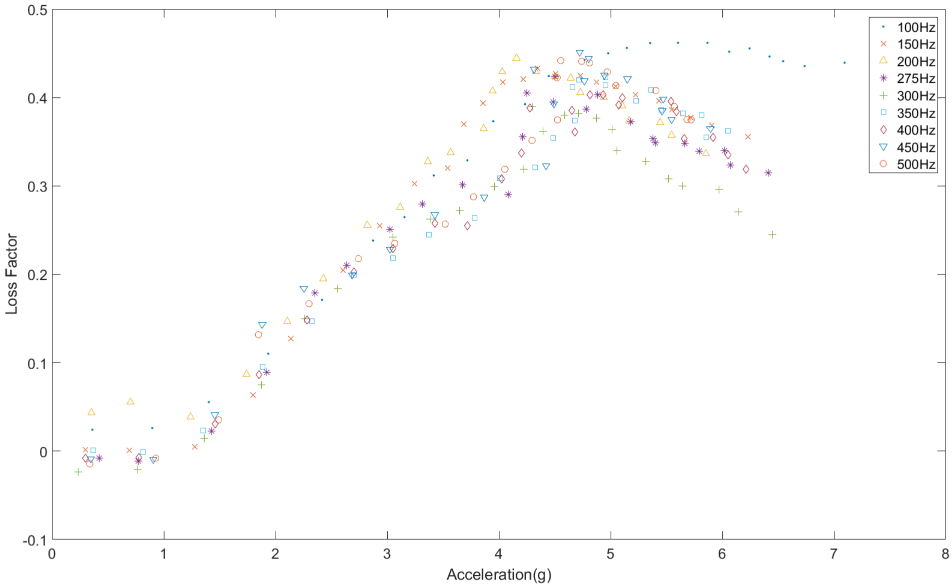


Figure 5.11: loss factor against acceleration amplitudes at different frequencies.

Slightly negative loss factors are common at this stage due to measurement errors and post processing with filters. Alternatively, they can be treated as a mark of a negligible damping level. There are exceptions at 100Hz and 200Hz, but they are still too small to signify distinctions.

In raising the excitation level, there is a clear increasing trend for the loss factor till it culminates within the range of 4g to 5g. Through carefully tracking of the trend, the acceleration at 4g could be characterised as a turning point, before which the increasing trend is steady for all frequencies, and after which there is an evident change in the slope. From the visual observation, it has been found before 4g, a quite noticeable portion of the particles are stimulated. The stimulated particles mostly are shaking or rolling at their original positions, and some particles move slowly within a small range (4g is not the exact number applicable to all frequencies, it could be smaller or large for tests at different frequencies. 4g is an approximate one, all tests exit this stage before 4.5g). Between 4g and 5.5g, large convectional motion of the particles can be observed, where the particles at the edge exhibit conspicuous fluid-like motion.

Within the peaking window, Figure 5.11 shows that almost all the curves peak coincidentally within the interval of 4.5 to 5g regardless of the frequency. Therefore, the occurrence of the maximum loss factor in terms of the excitation level only associates loosely with the frequency level.

In the high excitation region space (5g to 7.5g), all the frequencies behave congruously with an exception at 100Hz which sets a clear distinction from the rest. This phenomenon does agree with Saluena's phase diagram for their PD test (Figure 5.2), which shows at lower frequencies, higher acceleration are required to enter the gas state. According to the visual observation, at 100 and 125Hz, the PD enters the gas state around 5.5g (it is very difficult to distinguish the exact point of entering the gas state), which is a bit later than the tests at higher frequencies (at higher frequencies, they enter before 5g approximately). Thus the prolonged high damping level might be a result of the lack of records in higher acceleration (greater than 7g), where the same decreasing trend might happen. The gas state has an attenuating effect on the level of damping which has been agreed by the tests at other frequencies. In the gas state, there is a convex-like dependency for the loss factor on the frequency, which shows the loss factor at both ends (100Hz and 500Hz) is higher than in the middle (300Hz).

The convex-like dependence of the loss factor on the frequency at its physical interpretation is theoretically reasonable to be explained in terms of the means of dissipating the energy. It is well known and easy to perceive that the damping induced by the particle bed in vibration has two dominant sources. The first is the friction between particles, whose dominance signifies at lower excitation levels in terms of the combination of amplitude and frequency. The second source is the inter-collision between particles as well as the impact between the particle bed and the damper's interior wall. The energy discharge through impact could be largely classified into two categories. The first category has an almost exclusive association with the high amplitude vibration at low frequency, where only the solid state of the particle bed exists. In such situations, a high excitation level will cause the entire particle bed moving as a lumped integrity which will impact with the interior wall of the damper with a certain frequency. Apparently, such a type of impact possesses macroscopic sense for the entire particle bed; but such a scenario is not part of the current experiment. The second category is the microscopic particle-to-particle impact, which is more attached to the spirit of particle damping. At those more common stages in particle damping, the energised particle bed is subject to chaotic motions. Especially in the gas state, the particles fly around in the damper and randomly make impacts with each other.

The highest damping is supposed to be achieved at certain ratio between the energy dissipation through the friction and impact. Therefore, at a fixed frequency, the gradual increase of the excitation power, which naturally increases the acceleration, will at first increase the level of damping by introducing the means of energy dissipation into the system in both ways of friction and impact. The damping as represented by the loss factor peaks when the combining ratio between the friction based and impact based energy dissipation is at its optimum. The further increase of excitation level will vaporise the particle bed, and the system enters the gas state. At a fixed frequency, in the gas state, the increase of vibration amplitude is tantamount to increasing the distance for particles to move in the same direction. It means a reduction in the amount of impacts between particles, thus there is an obvious decrease in the level of damping in the high excitation power level. However, if the acceleration amplitude is held fixed, with the increasing of frequency in the gas state, the loss factor will decrease first and then rise again (convex dependency as mentioned before). It is because at lower frequencies, there is still quite a considerable amount of energy dissipated from friction between particles. But the further increase of frequency will set a more chaotic environment inside the damper, where friction based dissipation declines, and the impacts in the system is not yet drastic enough to uphold the damping level, thus there is a reduction in the loss factor. If the frequency still keeps increasing, the chance of impacts between particles as well as between particle and the interior wall will increase, since higher frequency vibration means more chances for two random particles to move in opposite directions. Therefore, a resurgence of the loss factor will eventually unfold through keeping increasing the frequency.

5.8 Conclusion

Overall from the 2D plot for the loss factor, it is by far reasonable, and strongly putative, to arrive at a conclusion that the loss factor is primarily influenced by the acceleration, compared with which the influence of frequency is duly subordinate. This biased dependence of the loss factor on the acceleration gives the chance to simulate the data all at once using any of the algorithms in Chapter 3.

On the implementation of the experiment, there are a number of points worthy of emphasis. Overall, the current set-up of experiment has generated data with decent quality with the adding of a digital filter in the post-processing step. However, the digital filter is after all a makeshift solution to the absence of good control over the measurement process. Besides, through the filter, even the primary signal which is set to maintain will still suffer from the excessive cut-off from the filter to some degree, which suggests a loss of information. Therefore, it is strongly recommended, in the future, to perform a refined version of the current experiment with a reliable and more sophisticated controlling system.

Despite some drawbacks in the experiment, the data acquired does exhibit piecewise behaviours in terms of the data variance and trend. Therefore, to use such a data for demonstrating the efficacy of the TGP model is eligible.

Chapter 6

Case study on the particle damping data

6.1 Chapter Overview

In the last chapter, via Yang's method in conjunction with some proper post-processing steps such as digital filtering, the 3D and 2D data of the loss factor against the acceleration amplitude and frequency could be obtained. To analyse such data composition, it is optional to perform the regression algorithm in either a 3D or 2D space. To perform a 3D analysis provides the advantages of explicitly establishing the relations between the output axis with all the input axes all from one single run of the algorithm, while to achieve the same amount of details in performing a 2D analysis, one should run it multiple times with respect to each axis. However, it is a treed model to be implemented, which means that the data space is subject to partitions parallel to the axis. Such a character of partitioning the data space parallel to the axis will cause trouble in performing simple high dimensional regression. As shown in Saluena's phase plot (Figure 2.1), the actual boundaries separating regions are not by any means parallel to either axis. Therefore, the ideal performance from a 3D regression is not possible to realise through any of the algorithms discussed in this thesis. As shown in the 2D plot in Figure 5.11, the variation of the loss factor with respect to the frequency is comparatively trivial in contrast to that with respect to the acceleration. Such a fact entitles a simplified analysis to be enforced on the data, where one single application of the TGP model on the whole

2D space could be performed in lieu of the separate applications at each frequency.

In this chapter, Figure 6.1 will be studied through performing both the traditional Gaussian Process model as well as the CTGP model on the 2D dataset of the loss factor vs vibration acceleration amplitudes regardless of the frequencies.

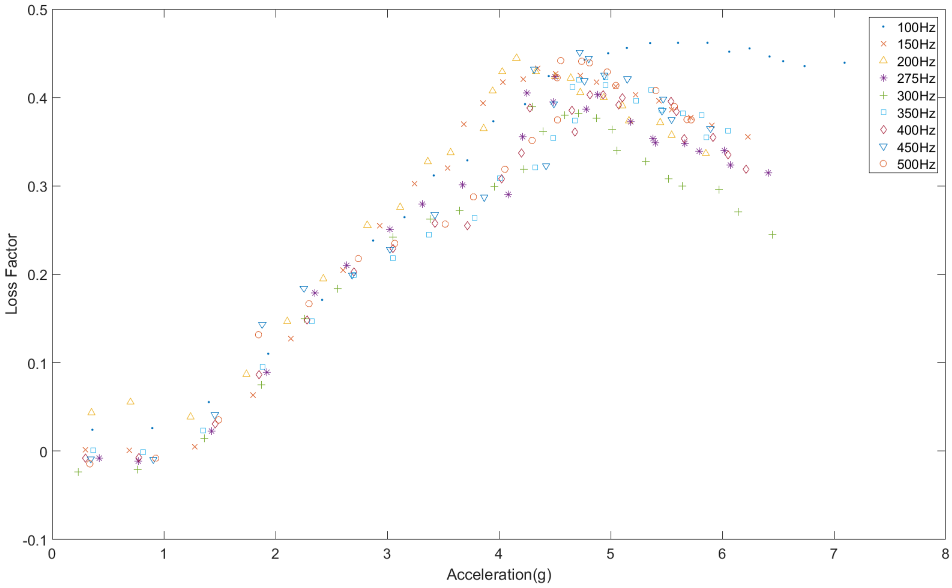


Figure 6.1: loss factor against acceleration amplitudes at different frequencies.

Because data from all the frequencies now form one single set, the partitions and regressions on such set in terms of physical interpretations intend to establish an averaged predictive relation between the loss factor and the acceleration if the excitation frequency is not given.

6.2 PD Data with the Traditional GP

Before applying the traditional GP model on the PD data, it should be reiterated that the GP model is a method that makes a mathematical statement in the form of predictive curves on the data space at a given set of hyper-parameters. It means that the GP model itself is not responsible for the inference of the optimal values for the hyper-parameters, thus the GP model for practical applications is always incorporated with an optimisation method for the choice of the hyper-parameters. This optimisation method could either be stochastically based or numerically based. In this section, both the two optimisation methods will be included with benchmarking between each other.

The first application of the GP model will not be directly performed on the entire dataset of the 2D loss factor plot. Instead, a preliminary application in a tentative sense will be performed on several sets of loss factor data measured at individual frequencies. By doing so, one could gain a refined and more accurate interpretation of the data space, moreover, to study the behaviour of the damping in detail with more intuitive understandings. Figure 5.11 has shown that the test at 100Hz is an exception to the general trend compared to the rest of the dataset. From the figure, the test at 275Hz also possesses a certain distinctiveness in the curve profile where it features an abrupt change of behaviour at 4g. In fact, all the other curves also features a switching of behaviour more or less at the similar acceleration; however the test at 275Hz stands out for behaving more abruptly at that turning point. In terms of the application of the GP model, such abrupt turning offers a technically rich ground to study how well the algorithm deals with discontinuity in the behaviour of the curve. Therefore, tests at 100Hz and 275Hz will be selected to undergo the application from the GP model.

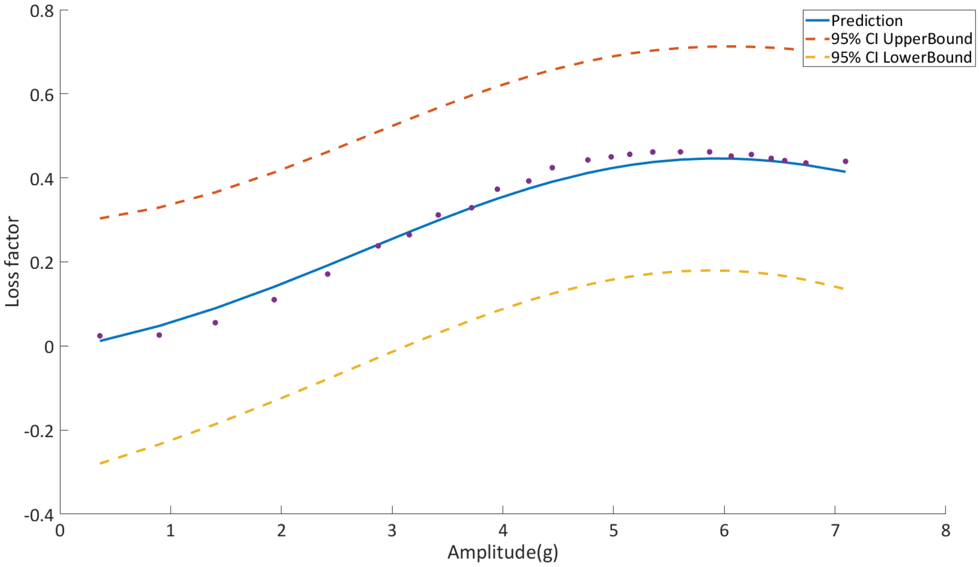


Figure 6.2: GP regression for loss factor at 100Hz.

The first application of the GP model on the data measured at 100Hz in Figure 6.2 shows the GP model has a disposition to interpret the data space in an approximate fashion of linearity through setting the predictive curve traverse all the data points from the middle. Although the general trend has been successfully captured by the GP model, the data less than 3.7g are overestimated, and the data over 3.7g are underestimated.

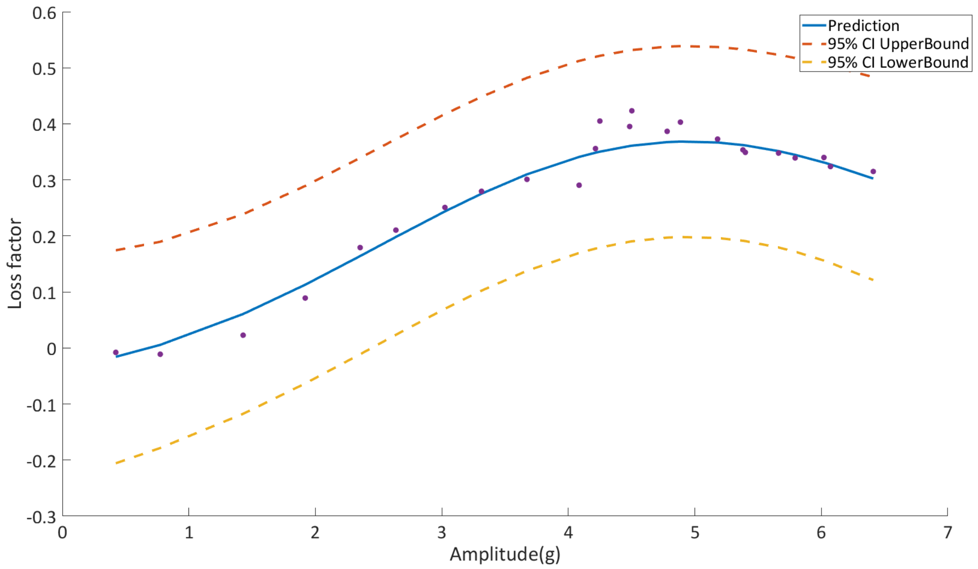


Figure 6.3: GP regression for loss factor at 275Hz.

For the data measured at 275Hz, again, the GP model presented a prediction in an averaged way by overlooking the details of the data. It is hard to say the algorithm succeeded in grasping the general trend, since the characterising feature of the data, the turning at 4g, has not been embodied in the predictive curve. The existence of the abrupt turning also exasperates the underestimation issues above 4g as a result of the sudden rise of the data after the turning.

Both figures above feature a comparatively wide confidence interval, which implies a diffidence in estimating the potential variability or uncertainty of the data. It indirectly suggests a lack of sufficient number of data points as evidence to enhance the credibility of the prediction. Such a fact also forms the motive for performing the algorithms on the full dataset as shown in Figure 5.11. Figure 6.4 shows the prediction from the GP regression on the complete PD data space.

As one can observe from the graph, with the increase in the number of data points, the space between the upper and lower bound are more densely occupied by the data bed when compared to the previous two figures. Such densely packed confidence interval states a much higher belief that the 95% percent of the chance for the predictive loss factor will fall within the interval.

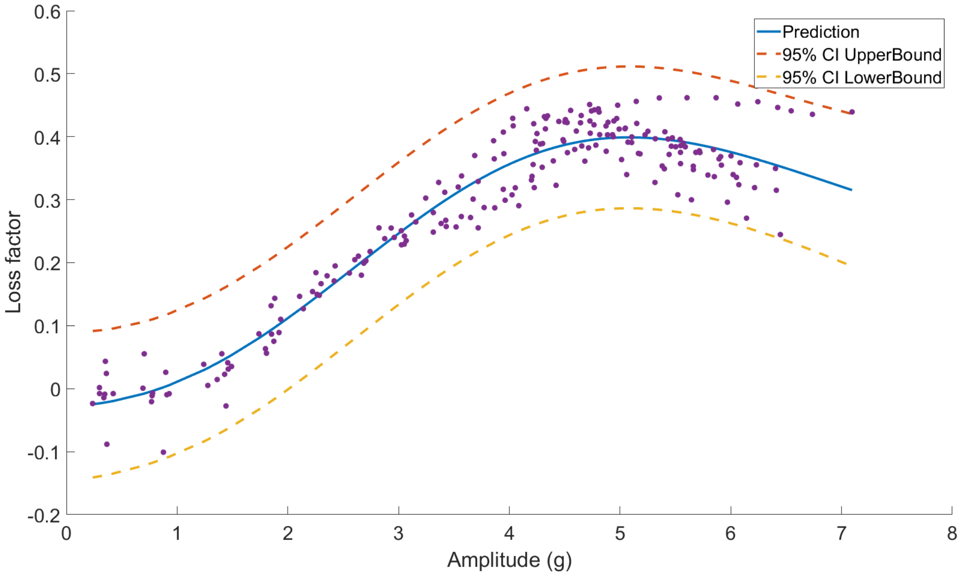


Figure 6.4: GP regression results for the complete PD data space.

The profile of the predictive curve does to some degree still maintain a resemblance to the curves in the previous two figures. The curve provided by the GP model for the data space is partially informative on the general trend of the data. It successfully located where the peak will take place (5g). The linear increasing period from 1.5g to 4g has also been modelled well, as well as the approximate linear decreasing stage after 5g. However, it is a crude modelling by treating the flat stage at the beginning as the inception stage for the linear increasing period. There also dwells a notable deficiency in such a way of predicting the data; the issue is with heteroscedasticity. Although the confidence interval has bounded tightly to invest high belief in the uncertainty in the data, such a consistent way of assigning the confidence interval uniformly along the axis is an inappropriate treatment of the data,0 where the change of variability of the data in terms of the location cannot be encapsulated in such models.

It is crucial for the algorithm to be able to model against heteroscedasticity in the particle damping data, because the heteroscedasticity among the data does encode important physical insight into the particle damping. The flat stage at the beginning is associated with a strict solid state of the particle bed where only a modicum of particles are excited to move. The narrow concentration of data around 4.6g does associate tightly with the achievement of maximum damping, and in terms of the physical observations, there is a constant behaviour of combined rolling and bouncing motion in the particle bed at this acceleration level across all frequencies. The declining stage at the end, which features a rather wide spreading, represents the drop of damping in the gas state. If all these differences in the variance could be captured by the algorithm, it is definitely conducive to establishing a better understanding of the physics in the particle damping by studying purely from the perspective of mathematics. Another advantage benefited from modelling against the heteroscedasticity is to selectively predict at locations of low variance.

6.3 PD Data with the CTGP Optimised Stochastically

Figure 6.5 shows the partitions of the data space as well as the predictions through 500 iterations of the CTGP,

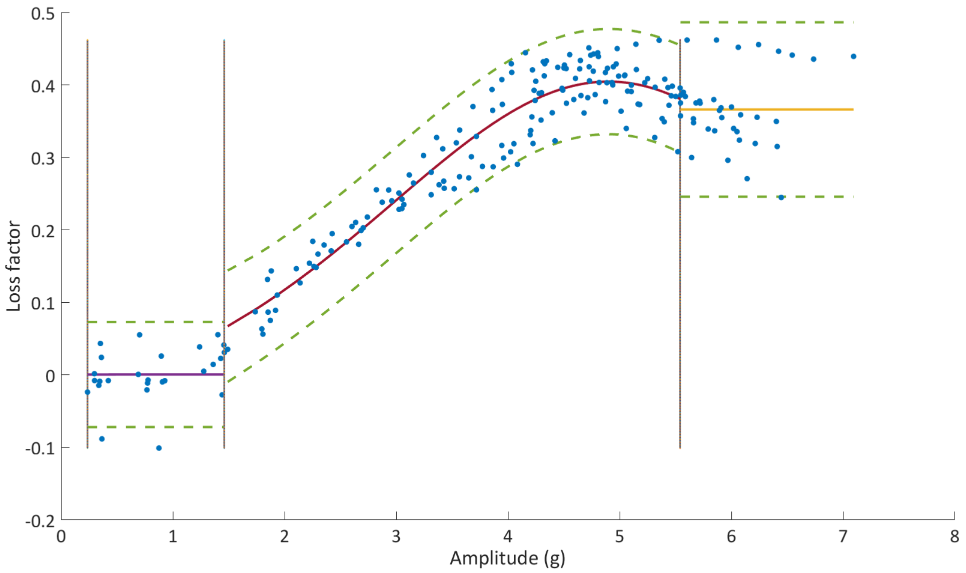


Figure 6.5: CTGP regression for loss factor (95% CI).

The algorithm has been run for multiple times, the partition scheme as shown in the figure is the most common result from the CTGP. There are rare occasions when another split will be put around 4g to more precisely separate out the nearly pure linear trend between 1.5g and 4g. In general, the CTGP did well in identifying the solid, liquid and gas states of the particle damping. In each of the regions, the exact predictive performance differs a lot.

In the first region, the CTGP does give credit to modelling the data as a flat line at a value of zero for the loss factor. Considering the physical insight of the particle damping, such a way of interpreting of the data at low acceleration is appropriate and reasonable. Because the assembly of polycarbonate container with the steel particles filled inside, maintains as a rigid body with all particles being still relative to the damper, there will be extremely low damping. From the confidence interval, one could also tell that the two points at the bottom, as deflected from the main stream of the data cluster, have been excluded as outliers from the first region. It is one essential advantage of using a TGP model. Because the outlier behaviour is relatively much more evident in the local region compared to the entire dataset, it allows the local variance to reveal its existence.

The second region features a curve resembling that which appears in the GP prediction. As mentioned before, the algorithm does occasionally put a partition around 4g to separate out the linear behaviour at lower accelerations. It also can be found that the variance between 1.5g and 4g is significantly lower than the variance after 4g, which putatively suggests such an additional partition. However, through examining the likelihood before and after the placement of that partition, the partitioned state only gains a slight margin over the un-partitioned state. Therefore, either to remain a complete region II or bipartite region II is not heavily one sided.

The last region is also modelled as a flat predictive line. This a somewhat problematic, because clearly as judging from the trend, at least it should be a declining straight line. However it is still reasonable to interpret in this way. Because the behaviour at 100Hz does not conform to a general trend exhibited at other frequencies. It is more linear and flatter, thus counterbalancing the declining trend at higher frequencies. Because of such a counterbalancing effect, the confidence interval in the last region is significant wider than in the previous two regions. Overall, such prediction is technically reasonable but practically inappropriate.

There are expected discontinuities between regions as well. Between region II and region III, the discontinuity does not cause massive trouble. But between region I and region II, the discontinuity is drastic with a severely disrupted appearance. It is troublesome, because if one wishes to predict in the vicinity of the split, the prediction will be embroiled in issues with high uncertainty.

Despite some drawbacks of using the CTGP, overall its performance is superior to the GP model in terms of prediction reliability, heteroscedasticity countering and linkage with physical insights. Among these three advantages, the heterosedasticity countering dominates. Because in the modelling of sparse data, high variances are ubiquitous, and it does inflict pains through introducing heteroscedasticity. That means the actual prediction at a given location is not so important since the potential variability for that prediction is high. For example, at the region I in Figure 6.5, although the prediction given at 1g is almost identical to the prediction at the same place in Figure 6.4, the tightened confidence interval does increase the reliability of such predictions. But there is a great advantage of using the GP model —the computational cost. The TGP is built on a series of GP applications on the data, it is definitely much more expensive; the expense is proportional to the number of iterations used. For this particular case study of particle damping, the TGP shall take time around 100 times more than the simple GP model to arrive at such a partition of the data space. But considering it purveys a linkage to physical insight of the particle damping from pure mathematical treatment of the data, such cost is worth a higher consideration. Obviously there are ways to alleviate the computational burdens, just like the replacement of stochastically based optimisation with a numerically based one.

6.4 PD Data with CTGP Optimised Numerically

The running results from the CTGP optimised numerically, exhibit uncertainty in terms of the predictive curve and partitions. Because a numerically-based optimisation is a deterministic approach to optimise a given function, its results will be affected by the initial condition of the optimisation search (eg. the starting location for the hyperparameters). Such dependence on the initial condition is, in fact, ironically an oxymoronic benefit, it is both inevitable and indispensable. This is because the reasoning of the numerical search for the optimum relies on the profile of the function, and it is the profile of the function that inflicts such dependency. The influence from the initial conditions will be exemplified later.

If say, the initial condition encumbers the searching process for the optimum, there is one issue that could potentially totally break down the optimisation. It is more of an elementary issue ingrained in the common ground of any numerical computations —the resolution issue. Unlike in the analytical approach, any computation conducted in a numerical environment is always subject to round-off and bounded limits for its size, like numbers could not be too big or contain too many decimal digits. All these issues do contribute to a notorious problem in matrix operation, which is known as *matrix singularity.* A singular matrix does not have an inverse if its determinant is zero. However, even sometimes a matrix with a non-zero determinant still cannot be inverted in the computational environment. The inverse is theoretically obtainable, but practically non-approachable, because the determinant is too small for the computer to handle without exceeding its limit in the storage or without losing a great deal of accuracy. In most cases the problem can be prevented by adding a random nugget term in the diagonal entries of a matrix. For the GP, if the SE kernel is selected, the third additional hyperparameter σ_n^2 not only describes the noise level of the data, but prevents the matrix from ill-conditioned for the matrix inverse. If the value of σ_n^2 is too small, the singularity issue will emerge. In the particle damping data, as seen in Figure 6.4, the data is strictly bounded between [-0.1,0.5]. In this sense, the variance, for example at the beginning of the data, will be rather small, thus during the numerical search, the algorithm will try to fit a small σ_n^2 to model the low variance, which collaterally generates the singularity problem. The only way to counter the problem is by magnifying the data space. The traditional way of data space resizing in the statistical community is the z-score normalisation [69].

$$z = \frac{y - \bar{y}}{\sigma} \tag{6.1}$$

where y is the output dataset, $\bar{y}$ is its mean and σ is its standard deviation.

Such a way of rescaling the data space will bound the data in [-1,1]. However, its new size is still not big enough to obliterate the matrix singularity problem. In the principle of magnification, one could naturally come up with the idea of multiplying the whole dataset with a number λ. But one should bear in mind that the choice of the λ will influence the final prediction. Therefore, the λ becomes a factor or even a hyperparameter involved in the simulation. However, in such a way, if the λ is modelled in the simulation, the model will be too complex. The current thought on the choice of λ will not be too harsh, that any λ that removes the matrix singularity is equally a good choice. Although such a treatment on the λ is not rigorous to the spirit of science, its applicability is reasonable. As mentioned before, the numerical optimisation depends on the initial conditions. Since the profile of the function is retained through magnification, ideally the initial condition with the rescaling forms a bound subject to relativity: It is equivalent to say to change the initial condition while keeping the scale is same to rescale while keeping the initial condition fixed. Technically, from the perspective of rigorous mathematical proof, such an equivalence is not tenable, because the gradient will change due to rescaling only for the output. But it can hold its ground in considering in terms of the searching destination. Theoretically, a certain initial condition will guide the search to a certain local optimum based on the function profile, if the function profile is retained in rescaling, that means in a certain scale any convergence from an initial condition to a local destination could find its resembling search in other scales. Hence, the λ rescaling is adopted. For the particle damping data, $\lambda = 100$ is found to be a good choice.

To see the influence of the initial conditions on the final prediction for a single GP, the 2D loss factor plot is used as an experimental subject.

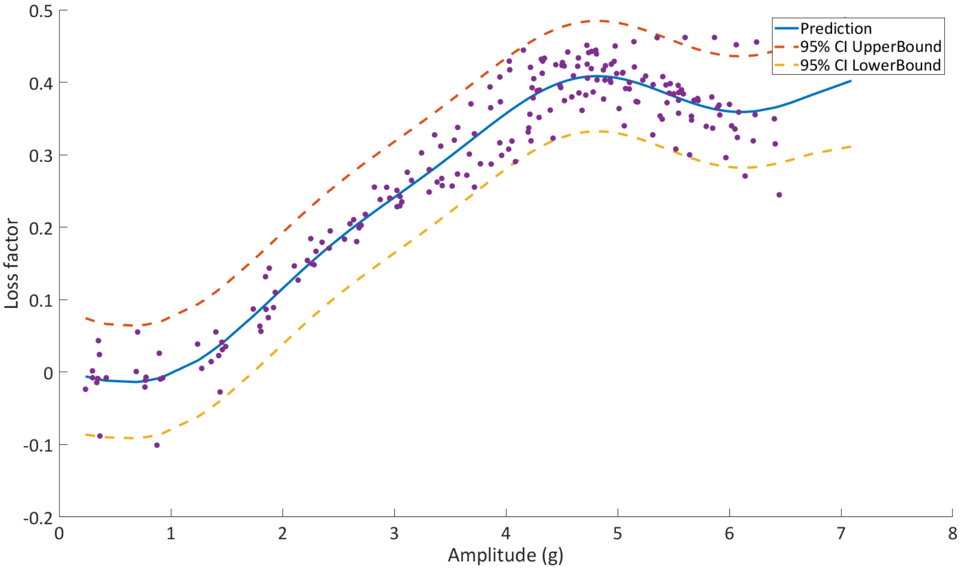


Figure 6.6: GP prediction at initial hyp setting [1,1,1].

Before analysing the performance, from the graph, one thing needs to be addressed first. The rescaling process is built to the algorithm, before plotting the graph, the prediction will be resized back to fit its original size. Therefore, no change will be made to the output size in the final plot.

Through comparing Figure 6.6 with 6.7, one could see at a small specification of the hyperparameter set, the search primarily moves in the dimension of σ_f and σ_n (check the title of the figure, which shows the final values for the hyperparameters).

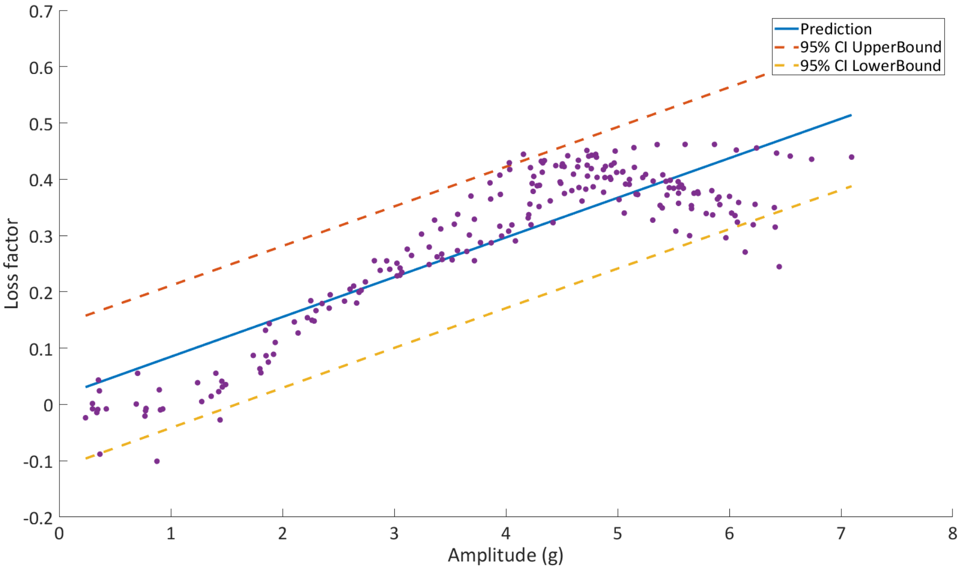


Figure 6.7: GP prediction at initial hyp setting[5,10,1].

However, in the case of much larger initial hyperparameter values, the search arrives at a place far from its starting location in every dimension. The small initial condition explains the entire dataset in a microscopic fashion by capturing its details in the function profile with the upper and lower bounds. The large initial condition explains the data in a macroscopic fashion by treating the dataset as a straight line as a result of a large scale length l^2. The dependence on the initial conditions for the search is rather obvious here.

As the purpose of applying a TGP model on the data is purely to differentiate region from region based on the details of the regional behaviour (e.g.variance), it will be reasonable to specify a small initial condition for the TGP model. If the same initial condition [1,1,1] is chosen for the hyperparameters σ_f, l, and σ_n, the following splitting space arrives in Figure 6.8.

Figure 6.8 shows an excellent agreement with the one derived from the stochastically-optimised CTGP. The running time for the numerically-optimised CTGP is on average around 500s for 200MCMC rounds, while for the stochastically optimised CTGP, it will take 1200s on average. The computational time also depends on the change of initial conditions. If large initial conditions [5,10,1] are selected, it takes less time, which averages around 340s. Its predictive space is given below in Figure 6.9.

The predictions and partitions given in Figure 6.9 are not beyond expectation, be-

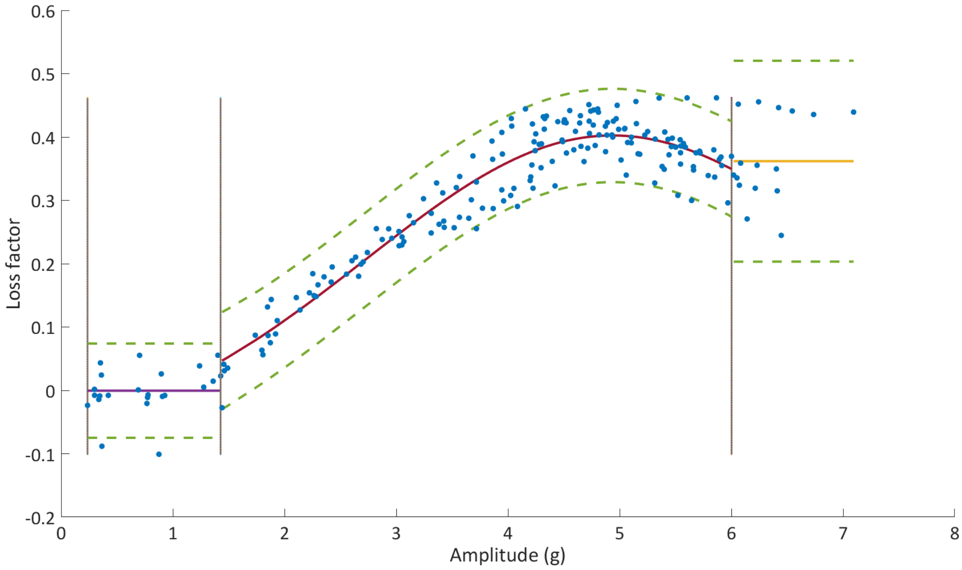


Figure 6.8: CTGP prediction at initial hyp setting[1,1,1].

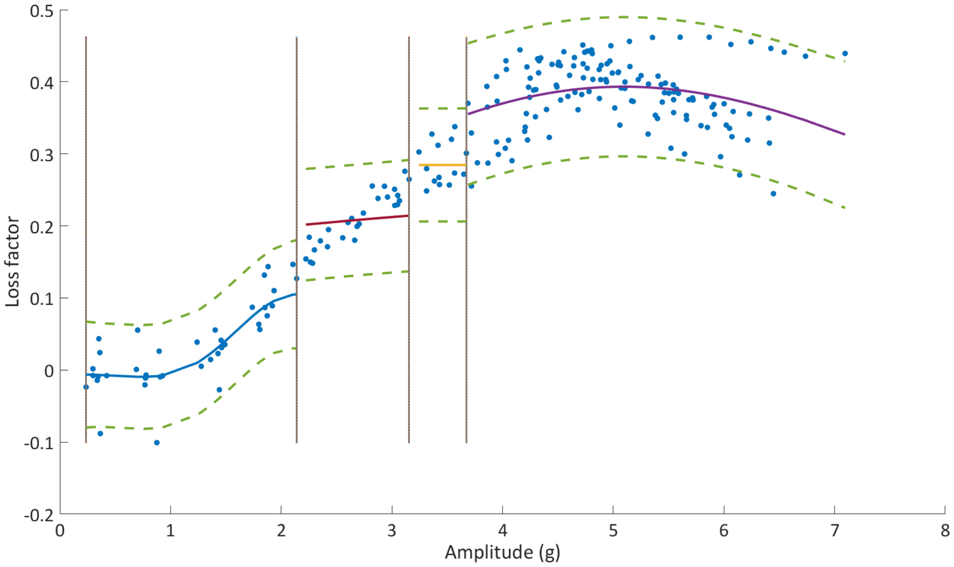


Figure 6.9: CTGP prediction at initial hyp setting[5,10,1].

cause it has been said before that the large initial condition will try to explain the input space as noise perturbing a linear trend; therefore, there are partitions at 3.2g and 3.7g. Obviously such a type of interpretation does not meet the desired expectation derived from visual observation of the data. Therefore, extra cautions must be paid in the selection of the initial conditions. Fortunately it has been found that small differences (e.g. change [1,1,1] to [1.5,1.5,1.5]) in the initial conditions do not cause a drastic difference in the final prediction, which means there is a healthy tolerance in guessing a good initial condition. To guess a good start point at a given rescaled data, one could use a stochastic approach to optimise the dataset first, and use its outcome as guidance for the guessing.

6.5 Conclusion

The behaviour of the loss factor in the particle damping is almost one-sidedly dominated by the acceleration. Such a phenomenon entitles a direct application of the GP and TGP models on the 2D data space of the loss factor vs acceleration. The direct application of the GP model on such a data space has shown the typical GP style of interpretation, where the predictive curve exhibits great smoothness and continuity. However, just as expected, the intrinsic problem with the heteroscedastity does set back the reliability and practicality of the GP model, for the variance of the data does hold important physical meanings. The application of the CTGP model has shown enough evidences in dealing with heteroscedasticity of the data, it has successfully identified different regions with different characters in terms of variance and trend. On the grounds of performance, one could firmly say the CTGP is indubitably superior to the GP model in modelling the particle damping; however, the inevitable significant rise of the computational cost must be endured in order to achieve such performance. Since the PD data and the MAD are two completely different case studies, it is highly expectable to see the CTGP capable of dealing a wider range of problems The next chapter will show the power of the CTGP in dealing with another completely different type of problem. In the end, the true practical significance of applying the CTGP model on the PD data needs to be clarified. As the current application is performed on single input and single output space which is easily visually observable, the more important benefits from applying the CTGP is that, the CTGP can provide more accurate confidence intervals than the traditional GP model. Therefore, such confidence intervals can help the user to identify measurement outliers which is potentially subject to high measurement error, or even is a signal of systematic faults.

Chapter 7

Z24 BRIDGE DATA ANALYSIS

7.1 Chapter Overview

The Z24 data case study offers the opportunity for the CTGP to show its capability in dealing with Structure Health Monitoring (SHM) problems. The SHM data measured in the Z24 study presents a great example of how a switch of physical property in the system influences the behaviour of the data, and how such physical switches could be identified through pure statistical modelling as offered by the CTGP. In the last two chapters regarding the particle damping, the CTGP is given a task to identify the different stages of the particle damping in terms of the behaviour of the particles. It could be said the case study on the particle damping placed the CTGP in a somewhat duplicated role as the Z24 data case study does, where both case studies are dealing with physical switches. However, in this study of the Z24 data, the data space will be more complicated in terms of the pervading high dispersion level across the entire data space. It challenges the CTGP on its capability of dealing with a scenario involving a high global variance. There are other interesting features dwelling in the Z24 data, which will be revealed later. By any means, it is a new field to test the performance of the CTGP in terms of the practical application, robustness etc.

7.2 SHM and Z24 Data

7.2.1 General SHM

Structure Health Monitoring (SHM) is an engineering discipline for estimating and supervising the state of health of a system (mainly structures) from data acquired from sensors permanently mounted on the system [70]. The health of structures, such as bridges, buildings etc, often could be evaluated from its dynamical behaviour in a form such as the acceleration response. The acquired data do contain the information indicating the potential risk of structural failure, because issues such as crack propagation and incipient damage can affect the dynamical behaviour of the structure. But to detect those subtle indicators of the damage in the vast data space is not a simple task, especially when those indicators are always concealed in the time domain of the measurement. To counter such a predicament, it is common in SHM to construct something termed as 'features', which are a series of low-dimensional vectors sensitively influenced by the damage. The common-vibration based SHM often uses the subject structure's natural frequencies or resonance frequencies as the features, where the damages caused by the unexposed cracks could be revealed through being modelled into the frequency, because the presence of crack induces a reduction in the local stiffness. The processing of the raw time-domain data to gain the damage associated frequency domain information is a typical way of *feature extraction* which is often undertaken in pattern recognition or machine learning [70]. The identification of the damage sensitive features allows the SHM inference engine to perform subsequent data analysis comprehensively in ways of both diagnosing and prognosing the health of the structure.

The straightforward schematic frame, when it comes to implementation, is not as straightforward as it seems. There are various impediments against the process of the feature extraction. The problem of *confounding influences* is the most common and notorious among those. As a systematic quantity, most of the features rarely associate exclusively with the damage. They could also show abundant responses to other benign changes in the system such as switch of mechanism, temperature change, wind speed etc. Influences attributing to these environmental or mechanical factors tend to adulterate the response of the feature, leading to confounding or concealment of the true damage-related change in the feature space. The natural frequency, as a common feature whose commonness as not only showing general sensitivity to the damage, but also shows a general dependence on the ambient temperature [71]. Against the problem of confounding influences, it is an alternative to either extract out features that bounds tightly and one-sidedly with the damage, or project out the confounding influence from the feature. Such a process is termed as *data normalisation.*

There are a large population of techniques currently feasible to conduct such data normalisation [72]. The criterion of choosing from these techniques is largely decided by the characteristic relation held by the feature with the different confounding influences. In vibration-based SHM, the measured time domain quantity (e.g. acceleration) does always exhibit a changing manner in a short time scale which associates its frequency with tens of Hertz. While, those confounding influences, such as ambient temperature and traffic loading, present their change in a much larger time scale, which associates with hours or days. Such a distinctive behaviour, when interpreted in the plot of the response surface for the time domain quantity, could be observed as a global-to-local contrast, where the confounding influences conduct the global cyclical behaviour of the response surface and the damage commands the local small variation of a noise type. If a conventional type of regression model is performed on such type of time domain data, it is almost certain that the model will overlook the local small variation and devote its inference in dedication to the general trend shaped by the confounding influences. Then this inference can be subtracted from the subsequent data to arrive at a new set of data that only depends on the damage.

The performance of the subtraction technique relies strongly on the reliability of the inference. Therefore, what the conventional regression techniques suffer from will become the primary concerns among which the problem of heteroscedasticity again will prosper, and it is where the TGP will show its power.

The presence of the heterocedasticity in such a context is mostly caused by the switching of operating mechanism or certain physical properties. Thus it suggests a piecewise heteroscedasticity which will fit perfectly into the bracket of the TGP. But one thing worth the emphasis is that, when compared with giving predictions to the data space, with no particular reason, the modelling of the regional piecewise variance is always considered as secondarily important unless the prediction is heavily disturbed by the existence of the heteroscedasticity. As has been stated in the chapter overview, the heteroscedasticity is modelled through identifying the location where the switching of data behaviour takes place. Such a location can always be paired with an important switching of physical property in the structure system. Again, it has physical significance just as in the modelling of the particle damping. But here, in the scope of the data normalisation in SHM, the identification of the switch brings another important advantage: It opens the chance of applying simplified fitting submodels (e.g. linear) within the individual region to generate predictions. For example, a selection of simple linear regression models could be applied to model the partitioned data space in stead of making a complicated global polynomial fitting which has a disposition to overfitting. Such simplified models are called parsimonious models. Parsimonious models serve the purpose of industrial simplification for pragmatic application; they trade off the prediction accuracy at a tolerable extent, but gain considerable improvements in flexibility, robustness and cost saving. The parsimonious model could either be achieved through the TGP or through a more generic idea, the Classification and Regression Tree (CART) equipped with linear regression. Worden and Cross have done some researches on the exact same case study with the original GTGP and the BCART models [44]. In the following sections, the current CTGP model will combat against both on the grounds of performance.

7.2.2 Z24 Case Study

The Z24 bridge is of significant importance in the context of SHM analysis. Prior to its retirement in the late 1990s, it had undertook a SHM campaign as a part of the 'SIMCES project' to explore the influence on the bridge from multiple environmental factors [73]. Through experimenting on the Z24, various influences on the structural health; such as wind speed, environmental vibration, temperature etc, could be set under the scope for investigation [74]. Of the current demonstrative case study, the feature of interest will be the natural frequency which is subject to confounding influences from the ambient temperature. The real physical scenario reveals a qualitative change in the asphalt as a source of confounding influence due to the stiffening of asphalt as a result of temperature variation. This change, although features a gradual transient process, could be interpreted as a switching of general curve behaviour in the input data space.

Figure 7.1 shows the four natural frequencies measured at 5652 samples. These samples are basically measured by the order of time spanning nearly a year from winter to late autumn. The dashed line at the point 4918 indicates the initiation of the damage. It has been found the second natural frequency (GREEN) is both sensitive to the damage as well as the temperature, and it is selected to study the performance of the CTGP in detecting the damage.

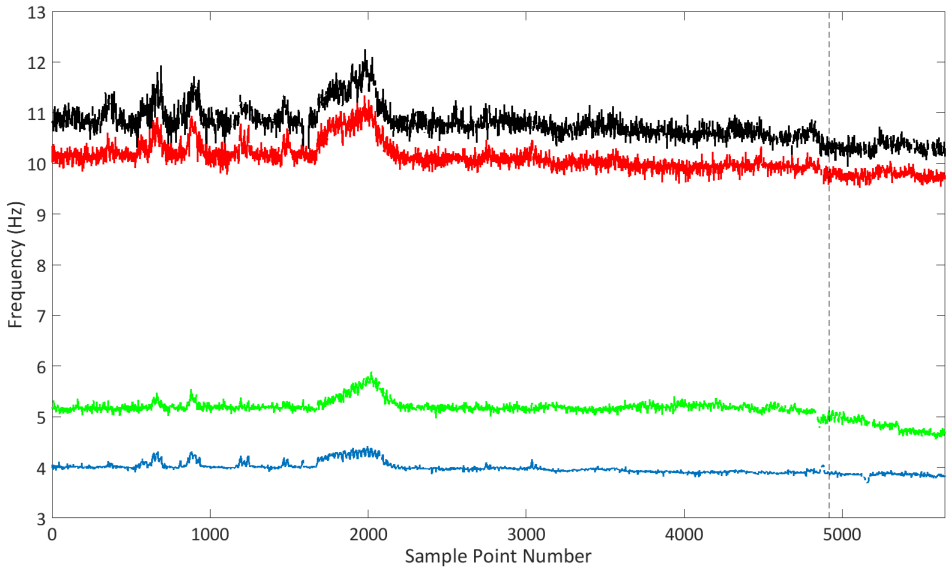


Figure 7.1: Z24 4 Natural frequencies.

The original data is comprised of 5652 samples, but some of them are void due to measurement failure. After culling out the failed measurements, 5277 samples are valid to use. The last 2000 samples, which contains the damage, will be used as the test data, and the starting 3277 samples will form the training data space.

The figure 7.2 shows the 2nd natural frequency as the feature against the confounding influence temperature in the training data space. Because the behaviour of the trend is rather mild and fluent, not all the 3277 samples are needed to conduct the learning. Therefore the data space is reduced to 820 data points through removing points uniformly across the data space. It is clear to see the data space presents a bilinear behaviour. The two linear trends could be presented in an explicit way through putting down a partition approximately at $0C^o$. The switching behaviour is expected at the prior awareness of the physical change in the asphalt. As the temperature decreases, the asphalt becomes stiffer, which consequences the increase in the natural frequency according to $f = \sqrt{\frac{k}{m}}$. One should bear in mind that, after the switching point, the stiffness of the asphalt will maintain at a generally steady level. This is a crucial requisite premise to the validity of the whole methodology. Because the relation between the natural frequency and the temperature will generally remain unaltered, the modelled predictive curve could be eligibly used to make predictions on the data outside of the range of the training data. Therefore in this sense, the TGP model is used for the extrapolation of the data with high prior predictability.

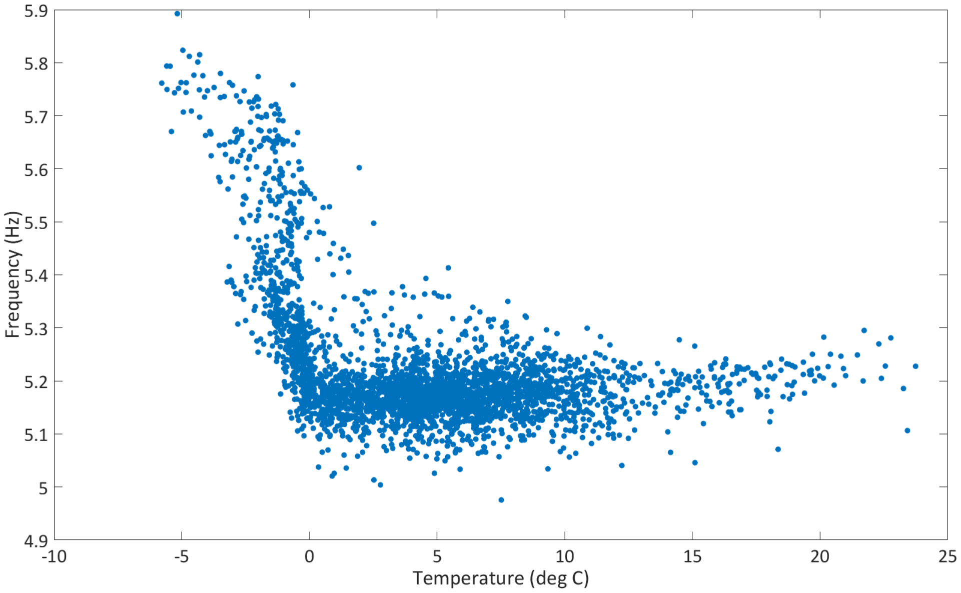


Figure 7.2: Z24 second natural frequency as a function of temperature.

As a customary way of unfolding the performance of the CTGP, the single GP will take a preliminary analysis on the data space giving Figure 7.3.

The prediction given by the GP, as can be seen in Figure 7.3, has made a seemingly perfunctory capture of the general trend of the data. Although the majority of the data points are well modelled by such a predictive curve, there are decisive deficiencies that could invalidate such prediction. The first problem is a comparatively minor one, taking place at the left end of the data space. The GP fitted an arch there attempting to explain the slight curved feature in that area. However it is an erroneous treatment to the data, superficially due to the lack of local data to reference. But more deeply it transpires that the GP has underestimated the length scale l^2 in the SE kernel. Such a fact could also be testified by the mildly undulating curve in the region above 0^oC, where a straight fitting line is more appropriate at the absence of informative traces from the data. Such an exaggerated interpretation by the l^2 connotes a deeper relation to a more severe problem in the confidence intervals (CI). The CI in the context of SHM is often used for detecting aberrant behaviours which can be the signs of damage. Thus to model the CI correctly is paramount. In

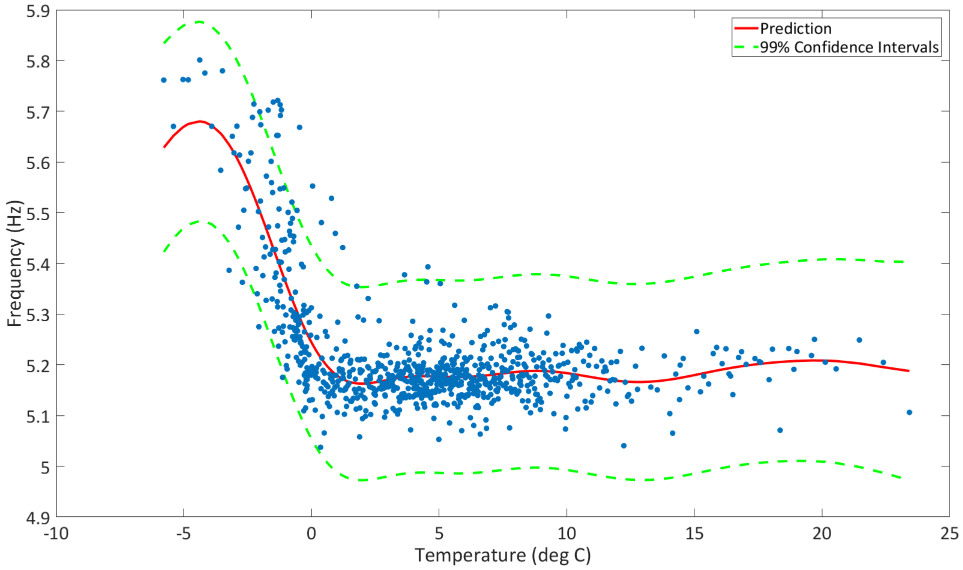


Figure 7.3: Z24 second natural frequency data with the GP in training temperature domain.

the figure, it can be perceived that the variance in the hotter region is higher than the colder region below 0^oC. The GP modelled the variance between [-3,0] well, but failed to remain confident in the left end and the hotter region. The inadequate CI at the left end is not as important as the one in the hotter region. However, the CI in the hotter region will directly influence the CI in the extrapolation of the test data which directly follows the training data after 25^oC.

Figure 7.4 shows the prediction from the GP on the test data in comparison with the measured data. The onset of the damage occurs at point 945 (the lowest spike in between [800,1000]). In the figure, the red predictive line represents the ideal relation between the frequency and the time without the presence of damages. Any excursions outside of its CI will carry the suspicion of being associated with damage. Although the development of the damage caused a gradual declination in the frequency, which eventually dropped outside of the CI, the onset of the damage still remains undetected at point 940. The reason is as addressed in the paragraph above, the CI is inappropriately modelled.

When the CTGP is applied to the data, it has been found 100 MCMC rounds are sufficient to produce the very partitions that indicate the physical switch. Two partitions are generated in the proximity of 0 ^oC, which yields three regions with distinctive variance levels. The left region is modelled with a curve exhibiting a

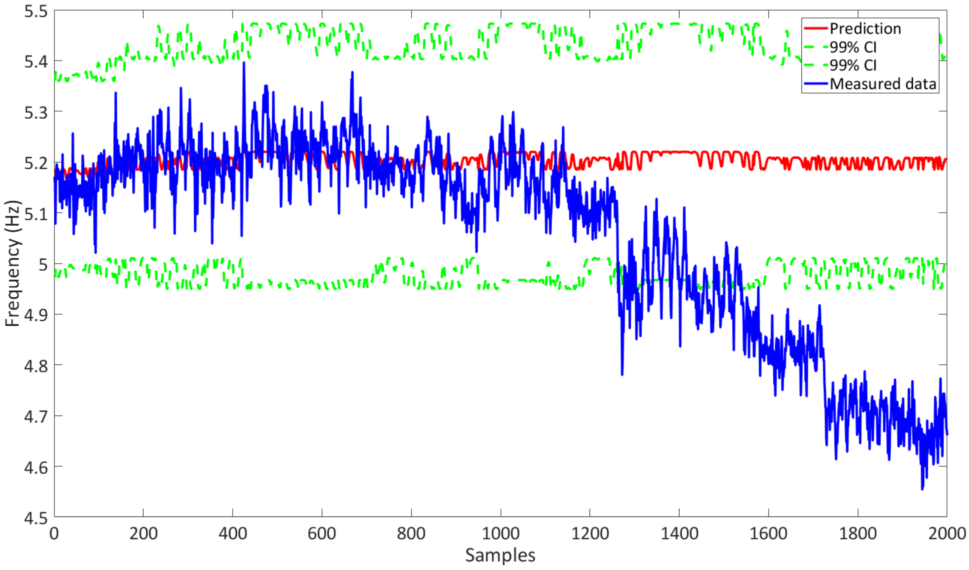


Figure 7.4: GP prediction on the test data in comparison with the measured data.

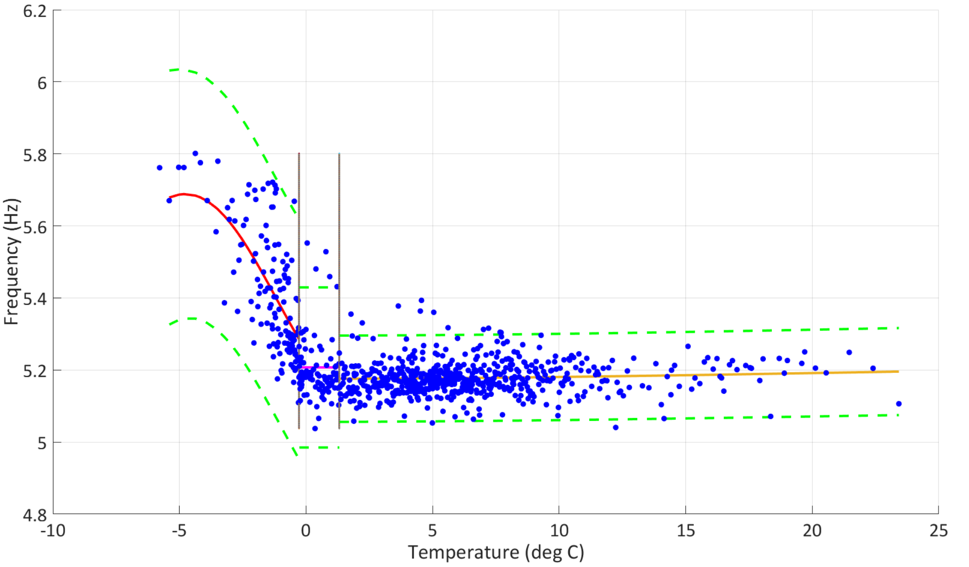


Figure 7.5: TGP prediction in the training temperature space.

gradual departure from the linearity from left to right. The curvature at the left end shows a partial agreement with the prediction given by the GP. However the curvature in the TGP prediction is very much moderated, which shows the advantage of segregating the data space to make the local data more informative. The middle region is considered to be a transient region featuring a short flat line prediction with a comparatively wide CI. From the standpoint of the statistical behaviour of the data, it could be sensed that between [-1,3], there features a gradual decrease in the variance level, which suggests a scenario of continuous heteroscedasticity. Although the decision tree model is not a natural fit to modelling continuous heteroscedasticity, to be able to identify the transient part still has reflected the effectiveness of the CTGP. The right region features a strict linear prediction of the data. Its contrast in the variance with the other two regions is rather distinguishable. Compared to the CI given by the GP in the same region, the considerably tightened CI encompasses the data in a more compatible fashion. It is definitely a better modelling of the local variance than the one the GP presented, as manifested by the CI's local consistency. Also, as benefited from the partitions, the undulating behaviour in the GP is banished completely in the results from the CTGP. Using such a modelling to predict the test data, the Figure 7.6 arrives,

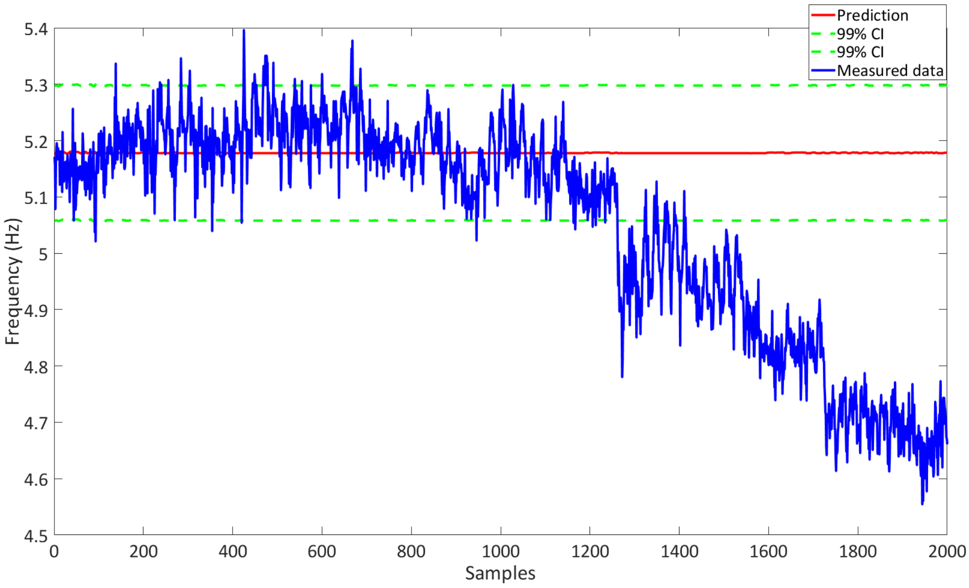


Figure 7.6: CTGP prediction on the test data in comparison with the measured data.

One can easily pick out that the predictive CIs in the test data domain have shrunken as a result of the same shrinkage in the last region of the training data space. Because the predictive model produced by the CTGP for the temperatures above 1.3^oC is a perfect flat line, that means the model holds a theoretical belief that the natural frequency is invariant with the temperature in the higher temperature domain. The 2000 data points in the test set are all measured outside of winter, hence the prediction is served as a flat line. The effectiveness of the CTGP is manifested at the damage initiating point 940, which has been successfully identified by narrowed CI produced by the CTGP. However, there is one problem one should bear in mind. Because the CI has been shrunken, that means the chance of false detections will rise. From the figure, it could observed before point 940, there are several times when the undamaged data fell outside of the CI, which shows an excessive sensitivity from such a model to the damage.

7.3 Introducing the CTGP with Pre-Partitions

The GP is notorious for its computational expensiveness. It is strongly discouraged to still apply the GP when the data space consists of over 10^4 data points, unless some processing have been done on the data. Because the GP is adamant on the matrix inversion which will cost n^3 operations to compute. In the CTGP, things get even worse, because the CTGP is a sampling process, where each sampling step requires a new set of matrix inversion involved in the hyperparameter optimisation. In this case study of the Z24 data with 820 data points, it has been found at least 200 MCMC rounds are required to optimise the hyperparameters for each MCMC step in the Markov space of the tree. At least 100 MCMC steps are required to search the MAP tree in the Markov space. Thus it means there are at least 20000 times of matrix inversion required for a 820×820 matrix, not mention that some other minor matrix inversions are required along the process. In the MATLAB, it takes averagely 1.5 hours to solve the case.

There are a number of ways to alleviate the computation burden. Here the strategy of pre-partitioning is introduced, which is innovatively introduced and developed by the author. The pre-partitioning, as the name suggests, will make pre-partitions in the data space before the real CTGP process starts. Fundamentally, each matrix inverse cost n^3 operations, if the full dataset is separated evenly into two $n/2$ sub datasets, the total computation cost will become $2 * (n/2)^3 = n^3/4$. If the dataset could be evenly separated into k subsets, the computational cost will be ideally reduced to n^3/k^2. At this point, one with an experience in statistics or familiar with the idea of Fast Fourier Transform (FFT) would probably think of partitioning the data space by orders of odd and even. However, this is not what is introduced here, instead the pre-partitioning strategy does not take a FFT-style partitioning scheme, rather conversely, it partitions the data space into k evenly spaced regions. These pre-partitioned regions rather than impairing or disrupting the data space, actually will be strongly beneficial. Because the treed model is a partitioning model, there ought to be partitions. Postulating that some pre-partitions exist in the data space, if that pre-partition is benign, it is good to have it. If it tends out to be a malignant one, it can still be pruned out through computation.

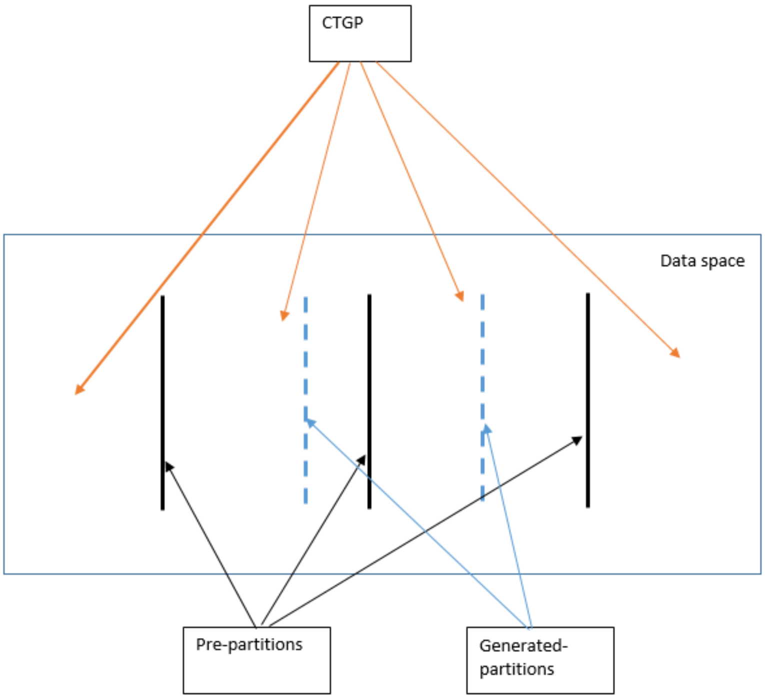
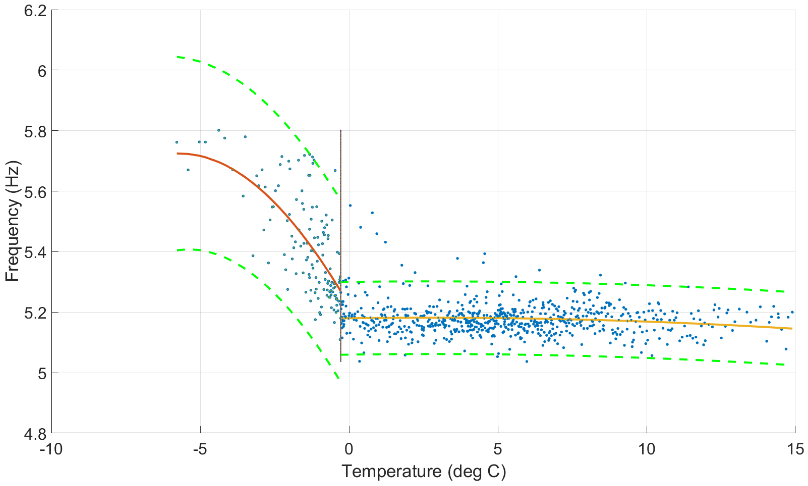


Figure 7.7: Illustration of the pre-partitioning strategy.

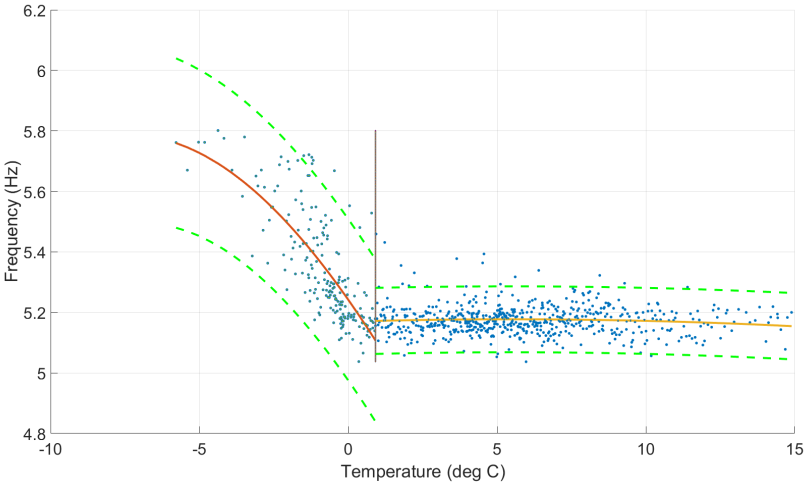
All the pre-partitions will stay fixed before the pruning stage, and the CTGPs will be applied to each individual pre-partitioned region to produce results there. In this way, not only the number of matrix inverse-related operations is reduced, but also the number of MCMC steps as well, because each pre-partitioned region contains less data. After all these have been done, there comes the procedure of diagnosing the bad partitions which will be pruned out from the space. The Figure 7.7 illustrates the concept of the pre-partitioning strategy.

To illustrate the performance of the pre-partitioning, the second natural frequency data against the temperature again is used (Figure 7.8). But a small modification is made to the data that the temperature is cut to be less than 15^oC for the purpose of better illustration (in this case there will only be one split which is easier to



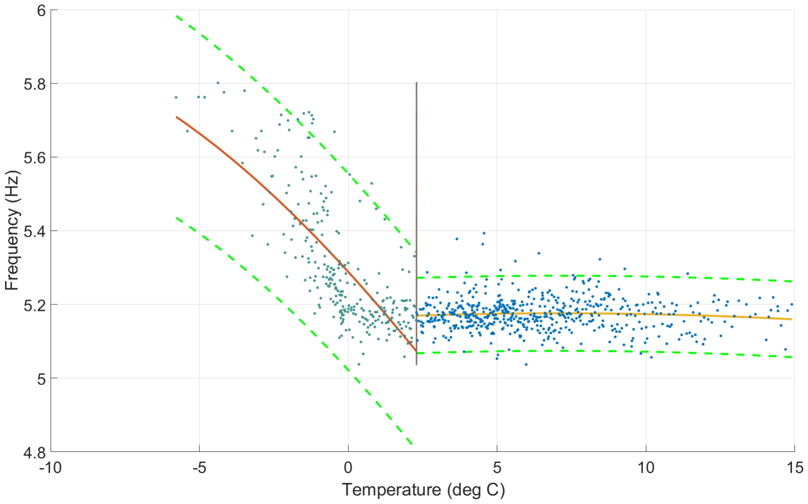
(a)

Figure 7.8: Pre-partitioning results: (a) result at 1PP



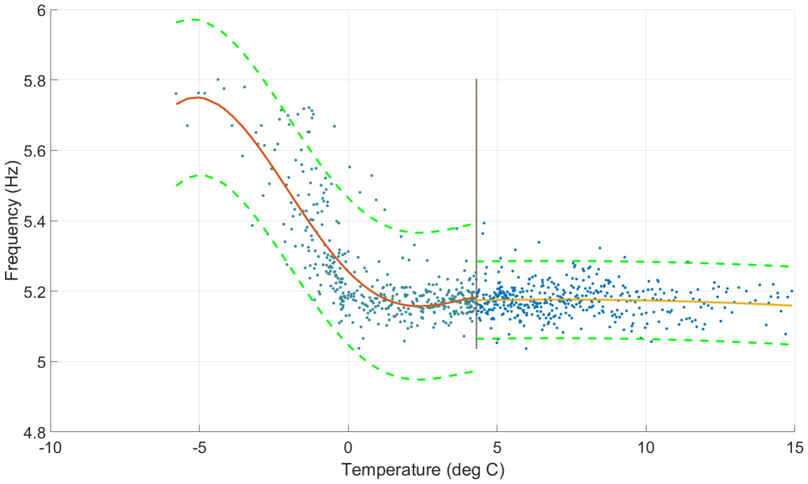
(b)

Figure 7.8: Pre-partitioning results: (b) result at 3PPs



(c)

Figure 7.8: Pre-partitioning results: (C) result at 5PPs



(d)

Figure 7.8: Pre-partitioning results: (d) result at 7PPs

explain). Ideally, there is one or two partitions to make in the interval [-1,1]. Figure 7.8 displays the four final partitioning results at four different initial pre-partitioning specifications. It is easy to observe that, although all the four results do only produce one partition in each, the location of the partition varies. It is easy to identify if this partition is the pre-partition or the generated partition through referring to the number marks on the x-axis (data are in [-5,15], the grid indicates four intervals, one can easily locate where the pre-partitions are accordingly). In fact, apart from the last result at seven pre-partitions, all the final partitions that finally stand are all generated partitions, which definitely shows the effectiveness of the strategy. The results at seven pre-partitions failed to identify the physical switch. It is because too many pre-partitions are allocated, the prediction in each pre-partition region is also worsened due to the local shortage of training data. Therefore, these pre-partitions or generated partitions are prone to be pruned off. For the first three results, the variation of the location of the final partition reveals a major drawback from such a strategy, partitions can only be approximately located close to their optimal locations depending on the initial pre-partitioning. It is not a huge problem for those cases featuring transient regions whose variance is continuously changing, because there is inherently no best partitioning location but an approximate one. However, if the case is like the Motorcycle accident data in Figure 4.1, such a strategy cannot be applied, simply because the miss of the partition at 15 can be legitimately considered as an error. The advantage of the pre-partitioning strategy is summarised in Table 7.1 shown below:

No. of PPs	total time(s)	Time per region (s)	Pruning time (s)
0	6672	6672	0
1	2278	1011	256
3	984	172	296
5	856	56	520
7	898	24	706

Table 7.1: Table illustrating the time consumption at different number of PPs.

From Table 7.1, one can see a massive drop in the total computation time at the presence of one pre-partition. By further introducing more pre-partitions, the algorithm can achieve a further reduction in the time before seven pre-partitions are allocated. At seven pre-partitions, the total computational time rose again. This phenomenon could be explained by observing the pruning time. When more pre-partitions are introduced, indeed the computational time for each region will drastically reduce, but more partitions will be subject to the pruning process. The pruning process is not cheap, because it is an exhaustive process, where the removal of a partition will merge two adjacent regions into one whose boundary partitions will be subject to further pruning as well, and it will be more costly since the merged region is more sizeable and requires more computation. Besides considering the performance, as illustrated in Figure 7.8, starting from five pre-partitions, the model has generally lost a good tracking on the optimal partitioning location. Therefore in this particular case study with one split involved, the three pre-partitions produced the optimal balance between the computational cost and inference accuracy.

7.4 Conclusion

The application of the CTGP is certainly an effective way to detect the onset of the damage in the Z24 study. It could be considered as a successful showcase of the generalisable applicability of the CTGP. Speaking of the case study itself, although the CTGP shows an advantage over the single GP in being more sensitive to the damage, the predictive result is still not unanimously flawless. The problem with the CTGP in the study of the Z24 bridge is that it is tends to be over-sensitive and normal fluctuations of the frequency due to temperature sometimes could be misdiagnosed. It is not in the same vein with saying the CTGP is incapable of distinguishing the confounding influences, since the CTGP did manage to subtract the confounding influence from the data. The white-noise behaviour in the residual approved the assumption of the theory. The problem seems a bit hard to explain at the current stage. At last the short introduction of the pre-partitioning strategy has proved to be useful at substantially reducing the computational cost. But it has a rather limited applicability. Future works are on the way to fully exploit its capability.

Chapter 8

Conclusion and future works

Referring back to the objectives of the thesis in the introduction, it could be concluded that the goals of all the objectives have been well met.

This thesis has successfully presented a new variation of the treed Gaussian processes, the CTGP, based on a more genuine and straightforward idea from the classic Bayesian classification and regression tree. Specifically four CTGPs are developed, but only two (stochastically-optimised CTGP and the Newton's method-optimised CTGP) are adopted, and the other two (the Quasi-Newton method-optimised CTGP and the Nonlinear Conjugate gradient method-optimised CTGP) are abandoned. The failure of the later two is due to the failure of Wolfe conditions which are practically indispensable and irreplaceable in these two models.

To fully understand the present TGP developed by Gramacy, the GTGP, is not a simple task. It has been shown in the chapter theory that the GTGP is a fully stochastic model subject to perfect randomness for both the tree structure and the hyperparameters. Its hyperparameter space is highly complex, which naturally shapes the complexity of the sampling system. It is definitely one of the essential successes in this thesis to replicate the GTGP in MATLAB in the absence of any detailed insight into Gramacy's code in R.

The benchmarking on the CTGP against the GTGP has successfully demonstrated that the power of the CTGP chiefly lies on the grounds of predictive accuracy (or reliability). It has been shown in the benchmarking chapter that the partitions produced by the CTGP evinces a higher sensitivity to the difference in variance than the GTGP which shows a general inclination towards putting partitions at turning points. Such a fact indicates that the CTGP performs better than the GTGP in countering the heteroscedasticity of the data. It is an advantage that could connect to a wide range of practical applications. The later two case studies on the particle damping and the Z24 bridge are typical examples where the heteroscedasticity is encoded with physical meanings. Apart from this, it cannot be predicated that the CTGP transcends the GTGP in every way in terms of the performance, because the sensitivity towards trend turning can be an advantage in some different applications. In terms of the computational cost, the GTGP leads by a large margin against the stochastically-optimised CTGP. The convergence rate of the posterior also shows the stochastically-optimised CTGP has potentials to further improve its efficiency considerably. The NM-optimised CTGP has a running speed almost at the same level as the GTGP, and its posterior inference quality is only a touch behind the stochastically-optimised CTGP, and its performance on the Motorcycle Accident Data (MAD) is highly commendable. However, when the CTGP is applied to the particle damping (PD) data, its drawbacks finally transpire. The NM-optimised CTGP could be highly dependent on the initial conditions for the optimisation, and the scale of the data space also could be vitally consequential. Without a comprehensive knowledge of the data space, the application of the NM optimised CTGP must be handled with caution.

The experiment on the PD has successfully generated valuable data in both respects of presenting the PD's physical characters and data's statistical peculiarities. It has been found for the current PD setup, that the damping loss factor is one-sidedly dependent on the excitation amplitude over the frequency. It entitles the problem of performing a 2D regression model on the PD data to be reduced to a 1D application of the model concerned with the single input variable, the excitation amplitude. Speaking of the process of the experiment, it could be said that the experiment is well designed to meet a high standard with the incorporation of the control loop, but is fulfilled at a lower level due to the failure of the control loop. It has been shown by the results after the post-processing from the filter that the acquired data still upholds a certain good level of quality in the absence of a functional control loop. But one should bear in mind that the filter is only a makeshift for the missing of a proper control of the system; it has irremediable drawbacks. First the filter will induce inevitable residual loss spanning the entire frequency range, thus causing undesirable damage to those good frequencies. Although the filter can clear the high frequency noisy signal, it cannot be used to adjust the deviations from the sinusoid shape which is at times happening when the excitation level is too high.

The application of the CTGP on the PD data managed to identify the three phases of the PD as described by the former researchers. The confidence interval in the prediction proved the statistical heteroscedasticity is an indicator for the physical change in the PD. The CTGP is the correct tool to use for the modelling of the PD.

The application of the CTGP on the Z24 bridge data in the context of SHM has demonstrated the versatility of applying the model. In the case study, it has been shown that the variance could be used as the quantity that indicates the onset of the damage. This naturally requires the accurate modelling of the variance, which stressed the importance of applying the CTGP again. Because the CTGP partitions the input space into regions with local consistent variance level which is more reliable and accurate.

8.1 Future Works

The future works could be generally directed towards two respects: namely, the future work on the PD and the future work on the CTGP.

8.1.1 Future works on the Particle damping experiment

The future work on the PD experiment will establish its interest on the amelioration and refinement of the current experimental defects. There are a number of things that could fit into this bracket. As has been addressed with emphasis in the former content of the thesis, the primary defect that depreciates the quality of the measured data is the absence of an effective control loop. The main issue with the current functionless PID control loop is that the manual tuning of the PID is either impractically inefficient or egregiously complicated against the some time turbulent dynamics of the PD system. In fact this issue belongs to a rather definitional domain in the control theory, namely the control of nonlinear systems. The PID is established on the basis of linear control theory, but it does not necessarily mean the PID cannot perform well in a nonlinear control environment. The studies on the nonlinear control with the PID are bounteous in the area of the automatic control. The existent solution could be as simple as linearising the system by approximation

[75] or as complicated as designing a module integrated with multiple controllers [76]. However, in either the simple or complex way, it is no longer within the scope of mechanical engineering, but a classic study object in the scope of the automatic control. Apart from the major flaw in the control loop, the experiment did not cover the lower end of the frequency, which is the frequency under 50Hz. The current testing frequency range has shown the variation of frequency has insignificant effect on the loss factor. However within the extremely low frequency range, the behaviour of the PD can change drastically. Because in that frequency range the PD could experience a new vibration mode associated exclusively with the impact between the particle bed and the damper's wall. A sudden switch of behaviour is almost guaranteed to be observed there, which means the power of the TGP model could be released more completely. Another potential improvable issue is quite similar to the one with the frequency. At the lower frequencies (e.g. 100Hz), the power supply from the shaker is not enough to drive the PD system into the stage of decreasing loss factor. A simple and direct solution is to use a more power shaker.

8.1.2 Future Works on the CTGP

The future work on the CTGP could be massive. The current version of the CTGP is more or less a prototype. Multifarious features could be developed and embedded into the CTGP. The CTGP is a highly robust model compared to the GTGP, because it is straightforward amalgamation between the generic CART and the GP. There is no complexity in either the hyperparameter space and the sampling scheme, which makes the CTGP a portmanteau receptacle to various different regression models. For example, if one wants to perform a piecewise generalised linear model (PGLM), one just needs to replace the current GP module with the PGLM, while the GTGP has to do massive adaptations to make it happen. Further down this route, one great potential held by the CTGP could be conjectured: A mixed piecewise regression model is made possible by the framework of the CTGP. It is definitely worthy of the investigation, considering that the GTGP is in incorporated with a similar idea, the limiting linear model (LLM). But the GTGP can only switch its inference method between the LLM and the GP as a result of being restricted by its prior specifications. If the CTGP could switch its inference method among a group of models, undoubtedly it will benefit both the performance and the computational cost.

Apart from changing the regression models, it is also feasible for the CTGP to change the regression kernel inside the GP. The current choice of the kernel is the standard SE kernel. It has been stated by some researchers that the SE kernel is unrealistically smooth for modelling real world events [31]. Thus it would be highly intriguing to change the SE covariance function to the Matern class (more realistic for real world data modelling), and then apply it to real world problems. The covariance function also determines the reasoning factors of the GP inference. For example, in the SE kernel, there are only three hyperparameters, σ_f^2, l^2 and σ_n^2 according to which three the GP can generate different predictions. In the benchmarking chapter, the partitions are reasoned also based on these three hyperparameters so to produce regions containing homogeneous local hyperparameters. Therefore, it will be highly exploitable to develop new covariance functions to guide the partitioning in different ways based on the user's preference.

The current CTGP is only capable of dealing with 1D problems. To make it functional for higher-dimensional problems is not difficult. Because, during the development of the TGP, the treed linear regression model is built as well, which features the multi-dimensional applicability. The reason why it is not yet introduced into the CTGP is generally due to two reasons. The first reason is the CTGP for the 1D application has already been considered to be cumbersome in terms of the computational cost. For higher dimensions, more data points are needed to specify the distinctiveness of each region. The 1D application of the CTGP has already been arduously struggling for a reduction in the computational cost, the 2D implementation will be unreachable from the grasp of practical applications. The second point shares the same concern, the computational cost. Although the computational cost of inferring the tree's posterior does not vary much with the dimensionality (the cost is $N^3 + DN^2$, where D is the dimensionality), the cost will be aggrandized immensely in the sampling space of the tree. It means the introduction of a new dimension will substantially raise the complexity in the partitioning results, thus the sampling size will grow almost explosively. Therefore, before making the CTGP functional for higher dimensional implementations, the foremost task is to reduce the computational cost.

In general, the approaches to reducing the computational cost could be divided into two categories. The first category encompasses the approaches that retain exact inference, while the second category allows the approaches based on approximation of the inference.

For the first category, future works could be done on investigating and developing other tree construction methods that could accelerate the MCMC search to find the MAP tree. In the same vein, the stochastic optimisation process for the hyperparameters could also be accelerated through introducing more efficient sampling schemes. The case study of the MAD has shown the well-balanced nature of the NM-optimised CTGP between the predictive accuracy and the computational cost. However for wide range practical applications, it is heavily disfavoured due to its sensitivity to initial conditions as exposed in the case study on the PD data. Despite such a negative fact, the NM optimisation could be used in combination with the stochastic optimisation to form a hybrid optimisation approach. The hybrid optimisation theoretically could take advantages from both sides, where the stochastic system could be used to help the NM to escape from local minima.

Whatever method taken from the first category can only gain comparative improvements in the speed. The overall cost will be still expensive as long as the exact posterior inference remains unaltered. In order to achieve further significant reduction in the computational cost, approximate approaches must be considered. In Bayesian statistics, it is common to encounter the rise of non-trivial distributions from the marginal likelihood which cannot be modelled by any simple probabilistic density functions (PDF) that allow direct sampling. The current GP is a good example, where the hyperparameters in the marginal likelihood can only be sampled through guided sampling strategies such as MCMC. The involvement of MCMC sampling is almost an announcement that the inference will be computational expensive. In recent years, Variational Inference (VI) has been favourably studied by many researchers, and has proved to be effective to a wide selection of inferences based on the Bayesian framework [77]. Briefly say, the principle of the VI is to posit a family of approximate distributions to fit the intractable complex PDF. For example if the marginal likelihood of the GP can be approximated by a simple PDF such as a normal distribution, the optimal values of the hyperparameters can be easily obtained rather straightforwardly. Some authors have already developed VI for the GP [78], and proved its generalisability in such applications. It will be definitely worthwhile to investigate into its details in the future.

Apart from these two categories which both focus on making changes inside the algorithm, although on different grounds their focuses are laid; in fact, other methods can also exist outside of the algorithm itself. The pre-partitioning strategy introduced in the end of Chapter seven is an example, which shows that the data space could be studied and reasonably modified to lessen the computational burden. The pre-partitioning strategy definitely has a good potential, which is worthy of being studied along with other methods of its kind, in the future.

Appendix A

PD experiment-related mathematical derivation

Based on the properties of harmonic motions, then postulating that,

$$
\begin{gathered}
F_{rms} = |F_{rms}|(a_1 + a_2 i), \qquad a_1^2 + a_2^2 = 1 \\
V_{rms} = |V_{rms}|(b_1 + b_2 i), \qquad b_1^2 + b_2^2 = 1 \\
V_{rms}^* = |V_{rms}|(b_1 - b_2 i)
\end{gathered}
\tag{A.1}
$$

then,

$$
\begin{gathered}
P = F_{rms} V_{rms}^* = |F_{rms}||V_{rms}|[(a_1 b_1 + a_2 b_2) + (a_2 b_1 - a_1 b_2) i] \\
P_{real} = |F_{rms}||V_{rms}|(a_1 b_1 + a_2 b_2)
\end{gathered}
\tag{A.2}
$$

Now given,

$$
F_{rms} = V_{rms}(c + \frac{(\omega^2 m - k)}{\omega} i) = V_{rms}(c + \lambda i) \tag{A.3}
$$

where $\lambda = \frac{(\omega^2 m - k)}{\omega}$, one has,

pairing these two equations,

$$\begin{aligned} F_{rms} &= |F_{rms}|(a_1 + a_2 i) \\ F_{rms} &= V_{rms}(c + \lambda i) \end{aligned} \tag{A.4}$$

Now, because $V_{rms} = |V_{rms}|(b_1 + b_2 i)$,

$$F_{rms} = V_{rms}(c + \lambda i) = |V_{rms}|[(b_1 c - b_2 \lambda) + (b_2 c + b_1 \lambda) i] \tag{A.5}$$

Equalising the coefficients with respect to the real and imaginary parts; the following relations arrive,

$$\begin{aligned} |F_{rms}|a_1 &= |V_{rms}|(b_1 c - b_2 \lambda) \\ |F_{rms}|a_2 &= (b_1 c - b_2 \lambda) + (b_2 c + b_1 \lambda) \end{aligned} \tag{A.6}$$

Now, a_1 and a_2 can be represented in terms of b_1 and b_2; substituting them in $P_{real} = |F_{rms}||V_{rms}|(a_1 b_1 + a_2 b_2)$, the following relations can be obtained,

$$\begin{aligned} P_{real} &= |F_{rms}||V_{rms}|(a_1 b_1 + a_2 b_2) \\ = |F_{rms}||V_{rms}| &\times \frac{|V_{rms}|}{|F_{rms}|}[(b_1 c - b_2 \lambda) b_1 + (b_2 c + b_1 \lambda) b_2] \end{aligned} \tag{A.7}$$

The λ terms are all cancelled out, also with $b_1^2 + b_2^2 = 1$ a simple expression is achieved,

$$\begin{aligned} P_{real} &= |V_{rms}|^2(c(b_1^2 + b_2^2)) \\ &= c|V_{rms}|^2 \end{aligned} \tag{A.8}$$

Appendix B

CTGP Pseudo-code

In this Appendix, the Pseudo-code for the CTGP is presented.

```
Initial condition specifications
for Main_loop_number < specified_number do
    Tree_alter_op = Randomise(GROW, PRUNE, CHANGE, ROTATE);
    Tree_proposed = Construct_tree(Tree_alter_op, Tree_current);
    Posterior_proposed = Posterior_eval(Tree_proposed, Tree_alter_op);
    Posterior_ratio = $\frac{Posterior\_proposed}{Posterior\_current}$;
    if Posterior_ratio < Rand(1) then
        Tree_current = Tree_proposed;
        Posterior_current = Posterior_proposed;
        Parameter_history(Main_loop_number) =
         Record(Tree_proposed, Posterior_proposed, parameters_proposed);
    else
        Tree_current = Tree_current;
        Posterior_current = Posterior_current;
        Parameter_history(Main_loop_number) =
         Record(Tree_current, Posterior_current, parameters_current);
    end
    Main_loop_number = Main_loop_number + 1;
end
Optimal_tree = Choose_max(Parameter_history);
plot&output
```

Appendix C

GTGP Pseudo-code

In this Appendix, the Pseudo-code for the GTGP is presented.

```
$Initial\ condition\ specifications$
for $Main\_loop\_number < specified\_number$ do
    $Tree\_alter\_op =$ Randomise($GROW, PRUNE, CHANGE, ROTATE$);
    $Tree\_proposed =$ Construct_tree_pra($Tree\_alter\_op,\ Tree\_current$);
    $[Posterior\_proposed, Parameters_{p}roposed] =$
     Posterior_eval($Tree\_proposed,\ Tree\_alter\_op$);
    $Posterior\_ratio = \frac{Posterior\_proposed}{Posterior\_current}$;
    if $Posterior\_ratio <$ Rand(1) then
        $Tree\_current = Tree\_proposed$;
        $Parameters\_current =$
         Parameters_Gibbs_update($Tree\_proposed, Parameters_{p}roposed$);
        $Posterior\_current =$
         Posterior_eval($Tree\_proposed, Parameters\_current$);
        $Parameter\_history(Main\_loop\_number) =$
         Record($Tree\_proposed, Posterior\_current, parameters\_current$);
    else
        $Tree\_current = Tree\_current$;
        $Posterior\_current = Posterior\_current$;
        $Parameter\_history(Main\_loop\_number) =$
         Record($Tree\_current, Posterior\_current, parameters\_current$);
    end
    $Main\_loop\_number = Main\_loop\_number + 1$;
end
$Optimal\_tree =$ Choose_max($Parameter\_history$);
plot&output
```

Bibliography

[1] Arthur L Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3):210–229, 1959.

[2] W Liu, GR Tomlinson, and JA Rongong. The dynamic characterisation of disk geometry particle dampers. *Journal of Sound and Vibration*, 280(3-5):849–861, 2005.

[3] P Veeramuthuvel, K Shankar, and KK Sairajan. Application of particle damper on electronic packages for spacecraft. *Acta Astronautica*, 127:260–270, 2016.

[4] Stepan Simonian. Particle damping applications. In *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*, page 1906, 2004.

[5] Clara Salueña, Thorsten Pöschel, and Sergei E Esipov. Dissipative properties of vibrated granular materials. *Physical Review E*, 59(4):4422, 1999.

[6] Paul Lieber and DP Jensen. An acceleration damper: Development, design, and some applications. *Trans. ASME*, 67(10):523–530, 1945.

[7] HV Panossian and DL Bice. Low frequency applications of nonobstructive particle damping (nopd). *Rocketdyne Corporation Document.*

[8] SF Masri. General motion of impact dampers. *the Journal of the Acoustical Society of America*, 47(1B):229–237, 1970.

[9] Yoshiaki ARAKI, Isao YOKOMICHI, and Junkichi INOUE. Impact damper with granular materials: 2nd report, both sides impacts in a vertical oscillating system. *Bulletin of JSME*, 28(241):1466–1472, 1985.

[10] Isao Yokomichi, Y Araki, Y Jinnouchi, and J Inoue. Impact damper with granular materials for multibody system. *Journal of pressure vessel technology*, 118(1):95–103, 1996.

[11] Randolph Danner Friend and VK Kinra. Particle impact damping. *Journal of Sound and Vibration*, 233(1):93–118, 2000.

[12] Chen Q and Worden K and Rongong J. Characterisation of particle dampers using restoring force surface technique. pages 1785–1790, 1 2005. URL . Accessed on 2018/02/04.

[13] CJ Wu, WH Liao, and Michael Yu Wang. Modeling of granular particle damping using multiphase flow theory of gas-particle. *Journal of vibration and acoustics*, 126(2):196–201, 2004.

[14] LS Fan and C Zhu. Principles of gas-solid flows, cambridge series in chemical engineering. *Cambridge University Press, United Kingdom*, 1998.

[15] TW Martin, JM Huntley, and RD Wildman. Hydrodynamic model for a vibrofluidized granular bed. *Journal of Fluid Mechanics*, 535:325–345, 2005.

[16] Peter A Cundall. A computer model for simulating progressive, large scale movement in blocky rock systems. In *Symp. ISRM, Nancy, France, Proc.*, volume 2, pages 129–136, 1971.

[17] CX Wong, MC Daniel, and JA Rongong. Energy dissipation prediction of particle dampers. *Journal of Sound and Vibration*, 319(1-2):91–118, 2009.

[18] Sean McNamara and WR Young. Inelastic collapse and clumping in a one-dimensional granular medium. *Physics of Fluids A: Fluid Dynamics*, 4(3): 496–504, 1992.

[19] Sean McNamara and WR Young. Kinetics of a one-dimensional granular medium in the quasielastic limit. *Physics of Fluids A: Fluid Dynamics*, 5(1): 34–45, 1993.

[20] Stefan Luding. Introduction to discrete element methods: basic of contact force models and how to perform the micro-macro transition to continuum theory. *European Journal of Environmental and Civil Engineering*, 12(7-8):785–826, 2008.

[21] Pieter A Vermeer, Stefan Diebels, Wolfgang Ehlers, HJ Herrmann, S Luding, and Ekkehard Ramm. *Continuous and discontinuous modelling of cohesive-frictional materials*, volume 568. Springer Science & Business Media, 2001.

[22] Nikolai V Brilliantov and Thorsten Pöschel. Granular gases with impact-velocity-dependent restitution coefficient. In *Granular Gases*, pages 100–124. Springer, 2001.

[23] Kuanmin Mao, Michael Yu Wang, Zhiwei Xu, and Tianning Chen. Simulation and characterization of particle damping in transient vibrations. *Journal of vibration and acoustics*, 126(2):202–211, 2004.

[24] Martín Sánchez and C Manuel Carlevaro. Nonlinear dynamic analysis of an optimal particle damper. *Journal of Sound and Vibration*, 332(8):2070–2080, 2013.

[25] M Saeki. Impact damping with granular materials in a horizontally vibrating system. *Journal of Sound and Vibration*, 251(1):153–161, 2002.

[26] Chian Wong, Michelle Daniel, and Jem Rongong. Prediction of the amplitude dependent behaviour of particle dampers. In *48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 2043, 2007.

[27] Prasanta S Bandyopadhyay and Malcolm R Forster. Philosophy of statistics: An introduction. *Handbook of the philosophy of science*, 7:1–50, 2011.

[28] M Bishop Christopher. *PATTERN RECOGNITION AND MACHINE LEARNING.* Springer-Verlag New York, 2016.

[29] Robert B Gramacy, Herbert KH Lee, and William G Macready. Parameter space exploration with gaussian process trees. In *Proceedings of the twenty-first international conference on Machine learning*, page 45. ACM, 2004.

[30] Daniel G. Krige. A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand. *Journal of the Chemical, Metallurgical and Mining Society of South Africa*, 52(6):119–139, December 1951. doi: 10.2307/3006914. URL `http://dx.doi.org/10.2307/3006914`.

[31] Carl E Rasmussen and Zoubin Ghahramani. Infinite mixtures of gaussian process experts. In *Advances in neural information processing systems*, pages 881–

888, 2002.

[32] William A Belson. Matching and prediction on the principle of biological classification. *Applied statistics*, pages 65–75, 1959.

[33] James N Morgan and John A Sonquist. Problems in the analysis of survey data, and a proposal. *Journal of the American statistical association*, 58(302): 415–434, 1963.

[34] Robert Messenger and Lewis Mandell. A modal search technique for predictive nominal scale multivariate analysis. *Journal of the American statistical association*, 67(340):768–772, 1972.

[35] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees.* CRC press, 1984.

[36] Hugh A Chipman, Edward I George, and Robert E McCulloch. Bayesian cart model search. *Journal of the American Statistical Association*, 93(443):935–948, 1998.

[37] John R Quinlan et al. Learning with continuous classes. In *5th Australian joint conference on artificial intelligence*, volume 92, pages 343–348. Singapore, 1992.

[38] Luís Torgo. Functional models for regression tree leaves. In *ICML*, volume 97, pages 385–393. Citeseer, 1997.

[39] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[40] Hyunjoong Kim and Wei-Yin Loh. Classification trees with bivariate linear discriminant node models. *Journal of Computational and Graphical Statistics*, 12(3):512–530, 2003.

[41] Yusuf Koc, Ecevit Eyduran, and Omer Akbulut. Application of regression tree method for different data from animal science. *Pakistan Journal of Zoology*, 49 (2), 2017.

[42] Jeppe Druedahl and Anders Munk-Nielsen. Higher-order income dynamics with linked regression trees. 2017.

[43] Andreas Hecker and Thomas Kurner. Application of classification and regression trees for paging traffic prediction in lac planning. In *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*, pages 874–878. IEEE, 2007.

[44] K Worden and EJ Cross. On switching response surface models, with applications to the structural health monitoring of bridges. *Mechanical Systems and Signal Processing*, 98:139–156, 2018.

[45] Probal Chaudhuri, Wen-Da Lo, Wei-Yin Loh, and Ching-Ching Yang. Generalized regression trees. *Statistica Sinica*, pages 641–666, 1995.

[46] George EP Box, Mervin E Muller, et al. A note on the generation of random normal deviates. *The annals of mathematical statistics*, 29(2):610–611, 1958.

[47] Milton Abramowitz and Irene.A. Stegun. *Handbook of Mathematical Functions.* Dover, 1965.

[48] Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in neural information processing systems*, pages 2933–2941, 2014.

[49] Kanti V Mardia and Roger J Marshall. Maximum likelihood estimation of models for residual covariance in spatial regression. *Biometrika*, 71(1):135–146, 1984.

[50] P Blomgren. Numerical optimization. quasi-newton methods, the bfgs method. *Computational Sciences Research Center, San Diego State University, San Diego*, 2013.

[51] David F Shanno. Conditioning of quasi-newton methods for function minimization. *Mathematics of computation*, 24(111):647–656, 1970.

[52] Philip Wolfe. Convergence conditions for ascent methods. *SIAM review*, 11(2): 226–235, 1969.

[53] Stephen J Wright and Jorge Nocedal. Numerical optimization. *Springer Science*, 35(67-68):7, 1999.

[54] Maurizio Filippone and Raphael Engler. Enabling scalable stochastic gradient-based inference for gaussian processes by employing the unbiased linear system solver (ulisse). *arXiv preprint arXiv:1501.05427*, 2015.

[55] Reeves Fletcher and Colin M Reeves. Function minimization by conjugate gradients. *The computer journal*, 7(2):149–154, 1964.

[56] B. W. Silverman. Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *Journal of the Royal Statistical Society. Series B (Methodological)*, 47(1):1–52, 1985. ISSN 00359246. URL `http://www.jstor.org/stable/2345542`.

[57] Robert B Gramacy and Herbert K. H Lee. Bayesian treed gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103(483):1119–1130, 2008. doi: 10.1198/016214508000000689. URL `http://dx.doi.org/10.1198/016214508000000689`.

[58] Alessio Sancetta et al. Greedy algorithms for prediction. *Bernoulli*, 22(2): 1227–1277, 2016.

[59] Richard Hamming. *Numerical methods for scientists and engineers.* Courier Corporation, 2012.

[60] John E Dennis Jr and Robert B Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*, volume 16. Siam, 1996.

[61] Jean Charles Gilbert. On the realization of the wolfe conditions in reduced quasi-newton methods for equality constrained optimization. *SIAM Journal on Optimization*, 7(3):780–813, 1997.

[62] Adrian S Lewis and Michael L Overton. Nonsmooth optimization via quasi-newton methods. *Mathematical Programming*, 141(1-2):135–163, 2013.

[63] Larry Armijo. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of mathematics*, 16(1):1–3, 1966.

[64] Don Coppersmith and Shmuel Winograd. Matrix multiplication via arithmetic progressions. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 1–6. ACM, 1987.

[65] Michael Yichung Yang. Development of master design curves for particle impact dampers. 2003.

[66] Monica Carfagni, Edoardo Lenzi, and Marco Pierini. The loss factor as a measure of mechanical damping. In *Proceedings-spie the international society for optical engineering*, volume 1, pages 580–284. SPIE INTERNATIONAL SOCIETY FOR OPTICAL, 1998.

[67] Benjamin Joseph Lazan. Damping of materials and members in structural mechanics. *PERGAMON PRESS LTD, OXFORD, ENGLAND. 1968, 317*, 1968.

[68] BRITISH STANDARD and BSEN ISO. Plasticsdetermination of dynamic mechanical properties. 2001.

[69] Gabriel Ziegler, Gerard R Ridgway, Robert Dahnke, Christian Gaser, Alzheimer's Disease Neuroimaging Initiative, et al. Individualized gaussian process-based prediction and detection of local and global gray matter abnormalities in elderly subjects. *Neuroimage*, 97:333–348, 2014.

[70] Charles R Farrar and Keith Worden. *Structural health monitoring: a machine learning perspective*. John Wiley & Sons, 2012.

[71] Charles R Farrar, WE Baker, TM Bell, KM Cone, TW Darling, TA Duffey, A Eklund, and A Migliori. Dynamic characterization and damage detection in the i-40 bridge over the rio grande. Technical report, Los Alamos National Lab., NM (United States), 1994.

[72] Hoon Sohn. Effects of environmental and operational variability on structural health monitoring. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 365(1851):539–560, 2007.

[73] Guido De Roeck. The state-of-the-art of damage detection by vibration monitoring: the simces experience. *Structural Control and Health Monitoring*, 10 (2):127–134, 2003.

[74] A.-M. Yan, Gatan Kerschen, P De Boe, and J.-C. Golinval. Structural damage diagnosis under varying environmental conditionspart ii: local pca for nonlinear cases. *Mechanical Systems and Signal Processing*, 19(4):865–880, 2005.

[75] WJ Rugh. Design of nonlinear pid controllers. *AIChE Journal*, 33(10):1738–1742, 1987.

[76] Shankar Sastry. *Nonlinear systems: analysis, stability, and control*, volume 10. Springer Science & Business Media, 2013.

[77] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112 (518):859–877, 2017.

[78] Dustin Tran, Rajesh Ranganath, and David M Blei. The variational gaussian process. *arXiv preprint arXiv:1511.06499*, 2015.