Towards Bayesian System Identification: With Application to SHM of Offshore Structures

A Thesis submitted to the University of Sheffield

for the degree of Doctor of Philosophy in the Faculty of Engineering

by

T. J. Rogers

Department of Mechanical Engineering

University of Sheffield

October 2018

Acknowledgements

It is impossible to properly thank all the people who have helped and supported me through the course of this PhD in this short section. They know who they are and I am truly grateful. There are, however, a few people whose support I would like to highlight and to whom I would like to extend my thanks.

First and foremost, I could not have wished for a better place to conduct this research than as part of the Dynamics Research Group. The DRG is a very special place, full of inspiring people; who, every day, push me to be better than I thought I could be. I owe a special thanks to Dr Lizzy Cross, who has been a wonderful supervisor, who has stopped me getting lost in many rabbit holes, and who has taught me how to write. I wish to thank Dr Graeme Manson for his advice and guidance, and for many interesting conversations about bread. Finally, I count myself very lucky to have had input from Prof. Keith Worden; who has been a solid sounding block, on a number of occasions.

I am also grateful to Ramboll Energy for their support of this work. In particular my thanks go to Ulf Tygesen who has been able to offer fantastic guidance regarding the aims of the offshore industry and insight into current industry practice. I wish to thank Prof. Thomas Schön for hosting me at Uppsala university for a very productive two weeks. Along with the rest of his team, especially Andreas Lindholm; his input was invaluable in completing the work found in Chapter 5.

My thanks go to Paul for being a great friend, for deciphering papers with me, and for helping generate more ideas than we can work through. Among the many friends I have in the DRG, I am especially lucky to have spent time with and worked with Ramon, Nikos, and Lawrence; for each of whom I am very grateful.

Finally, to Ruth, you are my best friend; I could not have done this without you. You have kept my priorities straight, encouraged me more than I deserve, and always been there for me. For all of this, and more, I am eternally thankful.

Highlights:

- A number of Bayesian approaches to SHM for offshore are presented and discussed
- The problem of system identification for SHM is handled in a Bayesian manner within both Gaussian Process, state-space models, and Bayesian clustering frameworks
- The best practice for Gaussian Process models is discussed with reference to their application in SHM
- It is shown how the kernel of a GP can encode prior belief about the physical process
- Novel use of population based optimisation of Gaussian Process hyperparameters is shown to outperform the usual gradient based approach
- The importance of model choice in Gaussian Processes is shown for an SHM example
- The Gaussian Process NARX model is presented, highlighting some of the challenges in its implementation
- A novel comparison of uncertainty propagation techniques for GP-NARX is made
- The problem of lag selection in GP-NARX is discussed and contrasted with parametric NARX models
- The modelling of wave loading — a key unknown — on offshore structures is attempted
- The current standard model, the Morison equation, receives a Bayesian treatment and is contrasted with a black-box approach
- The applicability of Gaussian Process models to wave loading (including GP-NARX) is explored and discussed

- The novel use of particle Gibbs is shown for system identification problems in structural dynamics
- It is shown how ancestor sampling and particle rejuvenation can aid Particle Gibbs methods in structural dynamics
- The identification of a Duffing oscillator highlights the effectiveness of the method — especially in the presence of high noise
- The Gaussian Process Latent Force Model is introduced for the problem of load estimation in structural dynamics
- It is shown that this can be efficiently rewritten as a linear state-space model
- This is used to perform joint input-state-parameter estimation in an operational modal analysis setting
- A Dirichlet process model is introduced for online Bayesian clustering of SHM data
- The technique removes the need to pre-collect a training dataset or add information on expected damage conditions
- Random Projection is exploited to allow online unsupervised dimensionality reduction
- A framework which allows for incorporation of prior knowledge of damage states leads to a flexible and practical model

Abstract

Within the offshore industry Structural Health Monitoring remains a growing area of interest. The oil and gas sectors are faced with ageing infrastructure and are driven by the desire for reliable lifetime extension, whereas the wind energy sector is investing heavily in a large number of structures. This leads to a number of distinct challenges for Structural Health Monitoring which are brought together by one unifying theme — uncertainty. The offshore environment is highly uncertain, existing structures have not been monitored from construction and the loading and operational conditions they have experienced (among other factors) are not known. For the wind energy sector, high numbers of structures make traditional inspection methods costly and in some cases dangerous due to the inaccessibility of many wind farms. Structural Health Monitoring attempts to address these issues by providing tools to allow automated online assessment of the condition of structures to aid decision making.

The work of this thesis presents a number of Bayesian methods which allow system identification, for Structural Health Monitoring, under uncertainty. The Bayesian approach explicitly incorporates prior knowledge that is available and combines this with evidence from observed data to allow the formation of updated beliefs. This is a natural way to approach Structural Health Monitoring, or indeed, many engineering problems. It is reasonable to assume that there is some knowledge available to the engineer before attempting to detect, locate, classify, or model damage on a structure. Having a framework where this knowledge can be exploited, and the uncertainty in that knowledge can be handled rigorously, is a powerful methodology. The problem being that the actual computation of Bayesian results can pose a significant challenge both computationally and in terms of specifying appropriate models. This thesis aims to present a number of Bayesian tools, each of which leverages the power of the

Bayesian paradigm to address a different Structural Health Monitoring challenge.

Within this work the use of Gaussian Process models is presented as a flexible nonparametric Bayesian approach to regression, which is extended to handle dynamic models within the Gaussian Process NARX framework. The challenge in training Gaussian Process models is seldom discussed and the work shown here aims to offer a quantitative assessment of different learning techniques including discussions on the choice of cost function for optimisation of hyperparameters and the choice of the optimisation algorithm itself. Although rarely considered, the effects of these choices are demonstrated to be important and to inform the use of a Gaussian Process NARX model for wave load identification on offshore structures.

The work is not restricted to only Gaussian Process models, but Bayesian state-space models are also used. The novel use of Particle Gibbs for identification of nonlinear oscillators is shown and modifications to this algorithm are applied to handle its specific use in Structural Health Monitoring. Alongside this, the Bayesian state-space model is used to perform joint input-state-parameter inference for Operational Modal Analysis where the use of priors over the parameters and the forcing function (in the form of a Gaussian Process transformed into a state-space representation) provides a methodology for this output-only identification under parameter uncertainty. Interestingly, this method is shown to recover the parameter distributions of the model without compromising the recovery of the loading time-series signal when compared to the case where the parameters are known.

Finally, a novel use of an online Bayesian clustering method is presented for performing Structural Health Monitoring in the absence of any available training data. This online method does not require a pre-collected training dataset, nor a model of the structure, and is capable of detecting and classifying a range of operational and damage conditions while in service. This leaves the reader with a toolbox of methods which can be applied, where appropriate, to identification of dynamic systems with a view to Structural Health Monitoring problems within the offshore industry and across engineering.

Table of Contents

Chapter 1

Introduction

As the title of this thesis reflects, this document and body of work represents a stepping stone in an ongoing pursuit by some (the author included) to revolutionise the way engineering is done in the 21st century. Society is in the midst of what some would term the "data revolution" [1], where the ongoing digitisation of everyday life is giving rise to ever larger collections of data. This is a key factor in driving what has been dubbed the "fourth industrial revolution" [2] in an attempt to characterise the rapid changes happening in the way the world operates. This transformation could be considered by some — possibly correctly — to be merely the next set of buzzwords driving this generation's "bubble". However, this new-found availability of large datasets and (possibly more importantly) computing power is changing how engineering problems are tackled.

Classical engineering, where experiments are performed on a small scale in a laboratory and empirical relationships are deduced (more often than not linear or log-linear relationships), is no longer the primary focus of research. Nor is the application of these simple models the main driver in engineering design and production, instead the use of Finite Element or Multi-body Physics models has become the mainstay of engineering design. More recently, developments in the application of machine learning models, in which the need for physical insight is often removed (or possibly ignored), have led to new approaches to solving engineering problems.

A field which has been heavily influenced by the explosion in machine learning seen over the last two/three decades is that of Structural Health Monitoring (SHM).

SHM is the process of collecting data from a structure of interest; in some manner using this to learn about the condition of that structure; and to aid and inform the process of making an engineering decision regarding its operation. Although the use of "physical" models to understand the condition of structures has been and continues to be an active area of research in the SHM community [3, 4], the use of machine learning or statistical methodologies has become a mainstay of many SHM analyses [5, 6].

It is the author's opinion that the engineering community is faced with a philosophical dilemma; will, as ever more data become available, research be able to uncover a hidden order in systems of interest, allowing perfect prediction of the behaviour of engineering structures? The experience of the author is that, rather than this being the case, data only highlights the disorder in the world around us. That is, the future of engineering will not be defined by the removal of uncertainty but rather defined by how it is understood and handled. This motivates the need to investigate methodologies which are capable of representing and incorporating uncertain data. Invariably, this involves the adoption and adaptation of methods, developed by statisticians and later machine learning researchers as a groundwork for rigorous handling of uncertainty.

1.1 Structural Health Monitoring: A Probabilistic Challenge?

The task of SHM is often approached by introducing Rytter's Hierarchy [7], which breaks down the overarching question of what is the condition of a structure into a number of sub-tasks, of increasing challenge. These being,

1. Detection
2. Localisation
3. Assessment
4. Prediction

This was extended in Worden and Dulieu-Barton [8] to a five level problem which is used now:

1. Detection — is there damage present?

2. Localisation — where has damage occurred?

3. Classification — what type of damage is present?

4. Quantification — what is the severity/extent of the damage?

5. Prognosis — what is the remaining useful life of the structure?

Hopefully, some of the advantages of changing the mindset of an engineer, when approaching these problems, will become clear as they are discussed here and throughout this thesis.

The problem of detection is a good place to start, not only is it the first challenge in the hierarchy but it is the simplest to understand from a probabilistic perspective. Detection is concerned with identifying whether or not a structure is damaged; see the fundamental axioms of SHM for a definition of this [9]. This has commonly been approached as a problem with a binary outcome, "yes there is damage" or "no there isn't". Here the task of damage detection becomes a deterministic one-class classification problem [10]. This can be thought of as giving the outcome as a 'crisp' or 'hard' label. There are many examples of this type of damage detection approach within SHM [11–17]. The alternative view on this problem would be to consider, not whether a structure is damaged or not, but instead to attempt to quantify a probability of damage which can be propagated into a decision framework, for example see [18, 19].

This assessment of different approaches can be continued up the hierarchy, for instance, considering the classification problem as estimating the likelihood of membership to each class [20–23]. Although, the added information from the probabilistic estimation of classes can often be neglected by condensing this to the single most likely label. Likewise, the localisation problem can be considered in terms of identifying the most likely place damage may have occurred [24] or even a distribution over possible damage locations. The quantification problem can be considered to be a problem of estimating the distribution over possible damage extents [25]. Finally, this approach to the prognosis problem becomes to estimate the distribution over the remaining useful life of a given structure [26–29].

1.2 The Bayesian Approach

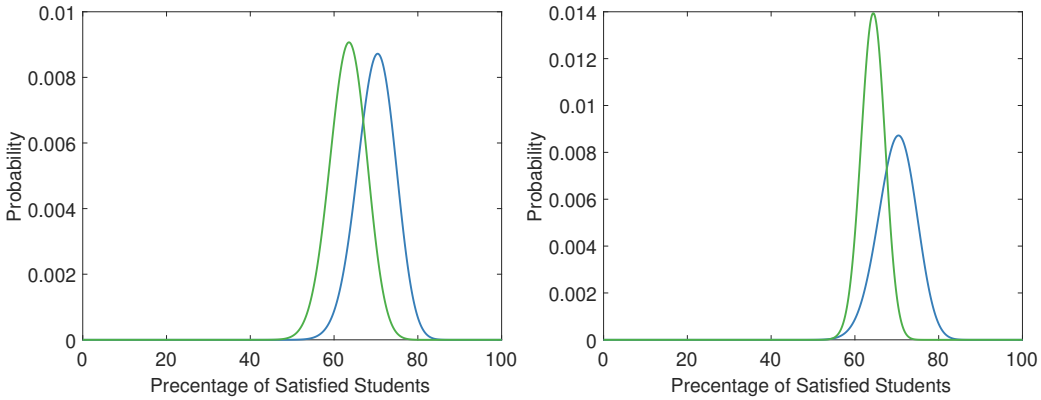
Since Rev. Thomas Bayes introduced a new manner in which to view problems of probability in the late 18^{th} century, the Bayesian approach has been a competitor to traditional frequentist statistics. Before considering the mechanics of a Bayesian approach to uncertain problems, it is useful to consider an example which highlights some of the benefits of a Bayesian outlook. Say a teacher decides to conduct a survey of a subset students in a class of two hundred in order to receive feedback. They ask twenty students if they are happy or unhappy with the course. Only six of these students reply that they are happy with the course and the teacher is put out — as any good instructor would be — assuming (as a frequentist) that the expected satisfaction in the class is only 30%. Over a coffee, they are advised by their Bayesian colleague to revisit their analysis from a different standpoint.

The colleague explains that, as a Bayesian, they think it is dangerous to accept only a small sample as representative of a population. They suggest that instead it might be useful to consider including any other knowledge that might be available. It transpires (this being a somewhat manufactured example), that there has recently been a survey of student satisfaction across the whole course (this class included). From this, it was possible to build a probability distribution of what percentage of students would be happy in a randomly selected class[1]. The two teachers now have everything needed to conduct a Bayesian analysis of the student satisfaction[2].

Breaking this down, the building blocks of Bayesian inference are having some model of the likelihood of the object of interest. In this case, the likelihood is simply the ratio of satisfied students to the total number of students surveyed. This is combined with some prior information which is encoded as a probability distribution. When choosing a prior it is important that it can accurately reflect *expert* beliefs; or, if not possible to do this, encode a lack of belief — in other words it formalises uncertainty before any data are collected. It is important that the prior has support over an appropriate domain, this means that samples from the prior will exist only in the domain (and across the entire domain) where the variable of interest exists. For this problem it is known that the object being modelled is the fraction of the students that are satisfied with the course, therefore, it does not make sense to have an outcome

[1] Formally, this prior is described by a Beta distribution with parameters $a = 70$ and $b = 30$

[2] Since the yes/no survey is a Bernoulli process and the prior is a Beta distribution, the posterior distribution can be computed in closed form.



(a) Sample of twenty (b) Sample of 180

Figure 1.1: Plots of the prior (blue) and posterior (green) distributions for student satisfaction given the survey of twenty students (a) and 180 students (b).

less than zero or greater than one. The Beta distribution is appropriate here since its support is $[0, 1]$ — i.e. all numbers drawn from it will lie in the interval $[0, 1]$.

By following a Bayesian analysis, it was possible for this teacher to compute the distribution over the student satisfaction given this survey of a subset of students. In Figure 1.1(a), the prior distribution is shown in blue and the posterior distribution is shown in green. The posterior encodes the distribution over the percentage of satisfied students, given the prior knowledge, which comes from knowing the overall course satisfaction, and the information from the survey of the subset. It can be seen that, since the survey includes only a small number of students it is not able to move the posterior very far from the prior — certainly it would fill the teacher with more confidence than accepting the frequentist result of 30% satisfaction!

Still concerned, the first teacher decides to extend the survey to 180 students (a much larger proportion of the class). Out of these students 110 indicate that they are satisfied with the class, which equates to $\sim 61\%$ in the frequentist viewpoint. If a Bayesian analysis is followed, the new posterior distribution (based on the same prior as before) is shown in Figure 1.1(b). Here, it can be seen that the prior belief has been able to moderate the effect of the small sample size from before and the results are remarkably similar. It should also be noted that as more data are collected the posterior distribution will tend towards the likelihood, that is that the frequentist solution is recovered in the limit of infinite data — which is where it is guaranteed.

But, how is the object of interest, the posterior distribution, computed? It is known

that the posterior distribution is a probability distribution over the variable of interest (in this case the proportion of students who are satisfied). The posterior distribution is best thought of as the updated belief after observing some data. This is just the rigorous combination of what a modeller believes before any observations are made (the prior) and the information contained in the observed data (the likelihood). Computing this requires the use of Bayes' theorem which is simple to derive given two key identities in probability theory.

The first is the relationship between joint and conditional probabilities, the chain rule of probability,

$$p\left(a, b\right) = p\left(a \mid b\right)p\left(b\right) = p\left(b \mid a\right)p\left(a\right)$$

the second is the law of marginalisation which is written

$$p\left(a\right) = \int p\left(a \mid b\right)p\left(b\right) \, \mathrm{d}b$$

Importantly, the chain rule for probabilities can be applied in either order, this makes the derivation of Bayes theorem trivial:

$$p\left(a, b\right) = p\left(b, a\right) \tag{1.1a}$$

$$p\left(a \mid b\right)p\left(b\right) = p\left(b \mid a\right)p\left(a\right) \tag{1.1b}$$

$$p\left(a \mid b\right) = \frac{p\left(b \mid a\right)p\left(a\right)}{p\left(b\right)} \quad \blacksquare \tag{1.1c}$$

This shows that Bayes' theorem is just an application of the chain rule of probability where, sometimes, this will be written with the denominator as an integral using the law of marginalisation property. Introducing some terminology, the distribution $p\left(a \mid b\right)$ is referred to as the posterior; $p\left(b \mid a\right)$, the likelihood; $p\left(a\right)$, the prior; and $p\left(b\right)$, the marginal. In terms of computation it can be more helpful to consider Bayes theorem as written below.

$$p\left(a \mid b\right) = \frac{p\left(b \mid a\right)p\left(a\right)}{\int p\left(b \mid a\right)p\left(a\right) \, \mathrm{d}a} \tag{1.2}$$

The problem in practical application of Bayesian methods is also more apparent when the theorem is written in this form. It is usually possible to write down a

likelihood for the data, this involves deciding on a probabilistic model for the system. It is also possible to elicit a user's prior belief, although care should be taken on how to do this [30, 31]. The hard part of the implementation is computing the marginal integral that is the denominator of Equation (1.2). In many cases this will not be available in closed form, this is why many Bayesian analyses use a seemingly very restricted set of prior distributions — usually in the exponential family. The choice of priors where the marginal, and therefore the posterior, can be computed in closed form for the given likelihood allows much faster analysis. These priors are referred to as being conjugate to the likelihood.

In a large number of cases, it will not be possible to ensure conjugacy in the model — even when solving a simple problem such as linear regression in a fully Bayesian manner it is not possible to obtain a closed form solution. In these situations it is possible to target the posterior directly via some numerical approximation, popular tools include Markov Chain Monte Carlo (MCMC) [32, 33] where the posterior is approximated by point masses generated from a Markov Chain or variational inference where the form of the posterior is approximated by a known distribution which can be computed in closed form [33].

The real beauty in the Bayesian analysis is that it conforms to a very human understanding of the world. That is, the belief that events don't happen purely in isolation. When encountering new situations, humans can draw on knowledge that they already have to help guide their belief given a small amount of available information. It is this ability to incorporate prior belief that helps humans generalise problem solving tasks across different domains (one of the great challenges in machine learning). It also helps explain how, as humans, two people can witness the same event but come to very different conclusions. To use a slightly topical example, when there is a change of government followed by an increased economic output, it is prior belief that leads one group of people to say this is a long term effect from the previous administration compared to another group which accredit this to the fresh policies of the new administration. As more data become available the strength of a given person's prior belief is actually revealed through their posterior (current) belief. Clearly, this discussion is in some ways facile, but an exposition of the philosophical basis for a Bayesian vs. frequentist standpoint is beyond both the scope of this thesis and quite possibly the capabilities of the author! If this discussion is of interest, most good texts introducing Bayesian inference include at least a passing discussion on the topic, see [33–35].

As becomes apparent when using these methods, the stumbling block in Bayesian inference is not normally a philosophical one but a practical one. Despite the theory existing to build complicated hierarchical models which can represent multi-layered engineering systems, lack of available knowledge and computing power can restrict the usability of these. As greater amounts of computing power become readily available it opens up new possibilities for the use of these highly complex Bayesian models — the challenge is not to become absorbed by the modelling and lose sight of what the model is trying to represent. It is important, therefore, to continue discussions about how this approach can add benefit to engineering analyses and be applied in a rigorous and usable manner.

1.3 Structural Health Monitoring as System Identification

The task of Structural Health Monitoring is closely related to that of system identification — if a system could be fully identified and modelled then prognosis within a Structural Health Monitoring framework would be possible. A perfect model of a system is not attainable; yet, through more rigorous and powerful system identification, many of the challenges in Structural Health Monitoring can be reduced. Here the task of system identification is considered in its broadest sense, to understand the behaviour of dynamical systems given some observations — and of course, being Bayesian, any available prior knowledge. This can include the task of parameter estimation but also extends to building models which can make predictions either over unknown outputs or to recover unknown inputs.

The central aim of system identification is to be able to recreate the behaviour of a system through some model which can be implemented and from which inferences can be made about the system. For uses in SHM by inferring the parameters of the model (more usefully the distributions over the parameters) changes in the system can be observed — this is the main tenet of model updating approaches to SHM [4]. In a data driven manner, the ability to predict the behaviour of a system accurately and with quantified uncertainty allows the use of these predictions for the modelling of damage mechanisms or for detection of changes in the system behaviour without the requirement for a valid physical model.

As a simple example, one of the hardest challenges in SHM is that of prognosis. A user may be interested in the prognosis for a structure given its fatigue damage accrual. If the response of the structure could be identified and predictions could be made regarding the expected strain across the structure then this could be used in a predictive fatigue analysis. Additionally to this, it would be necessary to identify the loading on the system which gives rise to this response in order to quantify the amount of expected damage in the future. This is obviously a significant undertaking in practice but, assuming that the system could be fully identified in terms of its response and prediction of expected loading, the task of prognosis for SHM becomes merely inspecting the outputs of the model. Hopefully this highlights the intrinsic link between the desire to implement robust SHM systems and the task of performing system identification.

1.4 Structural Health Monitoring for Offshore

The offshore energy industry was one of the earliest investigators into the application of Structural Health Monitoring on commercial structures. The nature of the offshore environment led to a need for structural assessment when traditional inspections were either not possible, or were very costly. Early examples of the use of vibration measurements include [36, 37]. As part of this, output only techniques for understanding a structure's behaviour were also developed from a system identification standpoint and operational modal analysis [38] remains a key tool for offshore.

The increased interest in SHM was motivated by the need to prove the structural integrity of these structures as they age and to do so in a cost effective manner. This has led to the development and integration of Risk & Reliability based inspection models where the inspection schedule is driven (at least in part) by measured data from the structure. The approach of Risk & Reliability attempts to model the risk as a probability of failure and when this exceeds a certain level to drive a cost effective intervention, see [39, 40]. Bayesian approaches to the problem have also been considered, for example pre-posterior decision theory [41, 42], or the use of Bayesian networks [43]. Although presented in literature, the current uptake of these methodologies in industry is unclear.

Although many of these technologies were developed for the offshore oil & gas industry, the growth of the offshore wind energy market is also driving development

of monitoring systems [44] — and related technologies. Due to the obvious influence of the condition of rotating machinery on the operation of the wind turbine, there has been investigation into the application of Condition Monitoring for offshore wind [45], this is a closely related field to that of SHM and it shares many of the challenges. In fact, there has been a substantial body of research into monitoring of wind turbine bearings and gearboxes [46–50].

The question is, what makes offshore a challenging (and therefore more interesting) environment in which to perform SHM? The nature of working with offshore structures is such that, many of the open problems in SHM are encountered at one time. These are systems in environments with severely changing operational conditions, which operators are unable to quantify [17, 51–53]; where the structural models are uncertain [54, 55] and based upon output only analyses [56] and modal expansion techniques [57, 58]. Alternatively, direct fatigue load analysis could be considered which requires estimation of the input loads [59–63]. These problems are further complicated by the inherent nonlinearity in the system, both from the nonlinear fluid structure interaction [64] and from foundation conditions [65–68]. These factors contribute to a highly uncertain environment with significant challenges for both system identification and for SHM due to the high number of unknowns and confounding influences. It is also an environment where there is tangible cost benefit for implementation of SHM strategies due to the high cost of maintenance and inspection. This motivates continuing research into the application of emerging technologies for SHM of offshore structures.

1.5 Contribution of This Thesis

The work of this thesis aims to introduce and utilise some potentially powerful technologies which, having been developed in the statistics and machine learning communities, can now bring significant value to engineering. The work focuses on the application of these methods to SHM problems. Before explaining these individual methods it has been necessary to present a short introduction to SHM from a probabilistic perspective and to introduce the Bayesian paradigm on which the methods used in this thesis are built. Chapter 2 introduces the Gaussian Process model as a Bayesian nonlinear regression tool for SHM. Following this Chapter 3 contains research into the practical application of these models, how the Gaussian

Process can be used robustly within an engineering context. This includes the use of the Gaussian Process NARX model for handling dynamic data. Chapter 4 contains a demonstration of the application of these Gaussian Process models for the problem of load identification on offshore structures — this is dominated by the wave loading.

Chapters 5 and 6 present the use of Bayesian state-space models for both nonlinear system identification and then the use of a Latent Force Model as an alternative approach to the load estimation problem. Finally, an online Bayesian approach to damage detection and classification is introduced in Chapter 7. This approaches the problem of SHM from a different perspective to the preceding chapters — that is treating it as a classification problem rather than a regression problem. This is followed by a number of conclusions which can be drawn from this work and suggestions on how the results seen here might motivate future research.

Chapter 2

Implementation of Gaussian Process Models for SHM

Highlights:

- *The best practice for Gaussian Process models is discussed with reference to their application in SHM*
- *It is shown how the kernel of a GP can encode prior belief about the physical process*
- *Novel use of population based optimisation of Gaussian Process hyperparameters is shown to outperform the usual gradient based approach*
- *The importance of model choice in Gaussian Processes is shown for an SHM example*

Within SHM, a large number of problems can be considered to be a regression task. This can be used in an indirect way, for instance, predicting some unknown variable in a system such as loading or response — a popular subset of this is sometimes referred to as virtual sensing [69, 70]. Alternatively, regression can be used directly within an SHM system to predict damage location or damage extent, e.g. the length of a fatigue crack. Various approaches are available to do this; for many years the use of linear regression models has been prevalent. This includes models that are

"linear in the parameters", such as a linear regression (e.g. the Morison equation for simplified calculation of wave loading) which can occur on a log-log scale as in common in fatigue analysis. Linear regression models can be insufficient to fully capture the behaviour of physical systems, in fact, within engineering many processes are inherently nonlinear, e.g. the true wave loading on offshore structures.

When the physical process of a system is known, it can be possible to define a nonlinear model parametrically. These models are normally computationally more demanding since many systems do not have closed form solutions — a pertinent example is the Navier-Stokes equations which would allow calculation of wave loading on a structure, but are famously not solved in closed form, for more information see Temam [71]. The approximation of these systems numerically is possible and is considered viable in certain circumstances [72]. The major challenge is when it is not possible to write down, exactly, the equations which govern a given system. This may be due to lack of specific knowledge about some aspect of the system, such as the boundary conditions. It could, however, be due to lack of understanding of the physics driving the problem. The alternative being that, the system is comprised of very many, individually complicated, systems such that it would be impossible for a user to fully describe its physical behaviour.

When this is the case, a model form has to be chosen which will be used for the regression. The problem is that, while there is a single truly linear model for a given set of inputs and outputs, there is an infinity of nonlinear models — that is the set of all possible basis functions. The model selection problem remains one of the biggest challenges in engineering where the physics of the system is not fully known. Many approaches restrict the model form to a set of basis functions — transformations of the input variables. For example, these bases could be chosen to be polynomial or Fourier [20]. The problem remains to choose the appropriate bases from a given set, "which polynomial terms will give the best model?". If fitting a polynomial regression model to a finite number of points, one will notice that increasing the number of bases will reduce the error when minimising the sum square error to fit the parameters (or if there are more parameters than there are data points the model is unsolvable). However, when new data arrive, this model will perform very badly. This is due to the function in the model being far more complicated than the function driving the process. This is commonly referred to as *overfitting*.

These types of basis function models are referred to as parametric models. This reflects the fact that, as more nonlinear functions are modelled, the number of bases

increases and with that, the number of parameters. There is one parameter associated with each basis, e.g. in a quadratic polynomial model there are three parameters, one for the quadratic term, one for the linear term, and one for the constant. It is this growing number of parameters that causes problems in *overfitting* and model selection. To combat this, approaches have been developed that broadly fall into two categories:

1. Cross-validation — leaving out subsets of data from the parameter estimation process and minimising the error on these sets, the limit of this being leave-one-out cross-validation where the size of the set is one data point.

2. Regularisation — introducing additional terms to the cost function used for parameter estimation which penalise more complex models to introduce a trade off between model fit and model complexity.

Both of these approaches are valuable in fitting parametric models but require comparison of many model forms and different model orders. This can be a difficult and computationally expensive task. It is desirable, therefore, to look for models where the model order (number of bases) does not need to be specified *a priori*. Such a model exists in the Gaussian Process (GP).

Gaussian Processes (GPs) present a flexible non-parametric Bayesian prior over functions [73, 74]. The GP model can be explained in a number of ways, the two most popular approaches are to either motivate the extension of the problem of Bayesian linear regression or to consider the GP as the limit of a multivariate Gaussian distribution as the dimensionality tends to infinity. The view of the GP as an extension of Bayesian linear regression is a more intuitive starting point in most cases, and this will be presented here briefly. More thorough discussions can be found in a number of textbooks [74–76].

The Gaussian process model as already achieved some attention in the engineering community. Particularly within SHM [53, 77–79]. This includes the investigation into the use of a GP as an emulator for a more expensive computer model [80, 81]. Most of this work has been to directly apply models developed within the machine learning community, for wider adoption, consideration of their implementation specifically for engineering warrants further investigation. This thesis aims to frame many of the results from the GP community within engineering problems and also to consider

additional difficulties that an engineer might encounter and how these might be overcome.

2.1 The Gaussian Process Model

When introducing the Gaussian process model, it is useful to start with the simplest Bayesian regression model — a linear regression model with noise only present on the output. For this, a parametric equation can be written down; where the aim is to determine the distribution over the weights, $\mathbf{w}$, which best describe the relationship between the column vector of inputs, $\mathbf{x}$, and the output, y. In the case of a standard one-dimensional linear regression, the weights represent the slope and offset of the line.

$$\begin{aligned} f(\mathbf{x}) &= \mathbf{x}^{\mathsf{T}}\mathbf{w} \\ y &= f(\mathbf{x}) + \varepsilon \end{aligned} \tag{2.1}$$

The model is formed such that it aims to model some target y as a function of an observed certain set of inputs $\mathbf{x}$ corrupted by an additive noise ε. The function is assumed to be an additive linear combination of the inputs weighted by some vector of weights $\mathbf{w}$. As is commonplace, the noise in the model can be assumed to be Gaussian distributed with zero mean and a known variance, σ_n^2, such that $\varepsilon \sim \mathcal{N}(0\,,\ \sigma_n^2)$. This assumption ensures that a solution to the model is available in closed form — this approach is coincidently equivalent to a Tikhonov or L_2 regularisation [20] when computing a deterministic model. By construction, it is possible to write down the likelihood of the output for a given input.

$$p(y \,|\, \mathbf{x}, \mathbf{w}) = \mathcal{N}\left(\mathbf{x}^{\mathsf{T}}\mathbf{w}\,,\ \sigma_n^2\right) \tag{2.2}$$

Usually, there will be more than one data point available so the vector of inputs, $\mathbf{x}$, becomes a matrix of inputs, X, where this matrix is assembled by stacking observations across the rows, $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, for N observed points. Therefore, the outputs for each observation, $\mathbf{y} = \{y_1, \ldots, y_N\}$, have a joint likelihood, $p(\mathbf{y} \,|\, X, \mathbf{w})$. Since the noise is generated i.i.d. (independently identically distributed) this likelihood can be computed as the product of the individual likelihoods. It should be noted at

this point that the offset in the linear model is introduced by extending the input matrix with a column of ones, this provides a more compact notation by not needing to state this offset explicitly.

$$\begin{aligned} p\left(\mathbf{y} \mid X, \mathbf{w}\right) &= \prod_{i=1}^{N} p\left(y_i \mid \mathbf{x}_i, \mathbf{w}\right) \\ &= \mathcal{N}\left(X^{\mathsf{T}}\mathbf{w}\,,\ \sigma_n^2 \mathbb{I}\right) \end{aligned} \tag{2.3}$$

In the case of the Bayesian linear regression, the first object of interest is the posterior distribution over the weights given the observed data, $\mathcal{D} = \{X, \mathbf{y}\}$, which is the distribution $p\left(\mathbf{w} \mid X, \mathbf{y}\right) = p\left(\mathbf{w} \mid \mathcal{D}\right)$. The likelihood has already been defined by the form of the model, Equation (2.3). It only remains to specify a prior distribution as the marginal can be computed via the integral of the likelihood and the prior. Since inference is performed over the weights in the model, $\mathbf{w}$, the prior must be specified for these. It is sensible to choose this prior to be a multivariate Gaussian since this ensures conjugacy with the likelihood (which allows a closed form solution) and has the required support for the weights (i.e. the distribution admits all real numbers as samples). The prior is parameterised such that it is zero mean with a specified covariance, $p\left(\mathbf{w}\right) \sim \mathcal{N}\left(0\,,\ \Sigma_w\right)$. Using these two definitions and the marginalisation property, the posterior distribution over the weights can be written down.

$$\begin{aligned} p\left(\mathbf{w} \mid X, \mathbf{y}\right) &= \frac{p\left(\mathbf{y} \mid \mathbf{w}, X\right) p\left(\mathbf{w}\right)}{p\left(\mathbf{y} \mid X\right)} \\ &= \frac{p\left(\mathbf{y} \mid \mathbf{w}, X\right) p\left(\mathbf{w}\right)}{\int p\left(\mathbf{y} \mid \mathbf{w}, X\right) p\left(\mathbf{w}\right) \mathrm{d}\mathbf{w}} \end{aligned} \tag{2.4}$$

If this is calculated, due to the choice of prior and likelihood, the posterior distribution of the weights is also Gaussian and the mean and covariance can be expressed in a closed form, by completing the square.

$$\begin{aligned} p\left(\mathbf{w} \mid X, \mathbf{y}\right) &\sim \mathcal{N}\left(\sigma_n^{-2} Z^{-1} X \mathbf{y}\,,\ Z^{-1}\right) \\ Z &= \sigma_n^{-2} X X^{\mathsf{T}} + \Sigma_w^{-1} \end{aligned} \tag{2.5}$$

This gives the distribution of the weights in the model based on the data which has currently been observed, both inputs and outputs, assuming that there is noise only present on the output and that noise is Gaussian distributed, zero mean with

known variance. This is normally, however, not the final quantity of interest. Instead, models are used to perform inference on new data; to make some prediction of a new output value given a new input value based on the data that has previously been observed. To do this, the posterior predictive distribution, $p\left(y_\star \,|\, \mathbf{x}_\star, X, \mathbf{y}\right)$ needs to be calculated. This is the distribution over a new test output $y_\star$ given a new test input point $\mathbf{x}_\star$ and the previously observed data. Again, the marginalisation identity allows this to be computed, the weights of the model are marginalised out leaving the distribution over this new output based only on the previously observed data.

$$\begin{aligned} p\left(y_\star \,|\, \mathbf{x}_\star, X, \mathbf{y}\right) &= \int p\left(y_\star \,|\, \mathbf{x}_\star, X, \mathbf{y}, \mathbf{w}\right) p\left(\mathbf{w} \,|\, X, \mathbf{y}\right) \mathrm{d}\mathbf{w} \\ &= \mathcal{N}\left(\sigma_n^{-2} \mathbf{x}_\star^\mathsf{T} Z^{-1} X \mathbf{y}\,,\; \mathbf{x}_\star^\mathsf{T} Z^{-1} \mathbf{x}_\star\right) \end{aligned} \tag{2.6}$$

This may appear to be a restrictive model; it is known that many functional relationships exist, in SHM and across engineering, which are not linear — in fact it is often hypothesised that nothing is truly linear. This, however, is not as significant a barrier as might be expected. It is possible to transform the inputs using a basis function expansion, here instead of only considering an input $\mathbf{x}$, some transformation, $\phi\left(\mathbf{x}\right)$, of this input can instead be fed into the model. A common basis, which could be used, is a polynomial basis where the inputs are generated as polynomials transforms of the input, for example if a one dimensional input, x, is observed this can be related to the output via a quadratic rule where $\phi\left(x\right) = [x^0, x^1, x^2]$. Here there is a separate weight over each term in the basis function expansion, and the transform fits into the Bayesian linear regression model as before if we assume that $\mathbf{x} = \phi\left(x\right)$. Likewise, for multiple observations the same basis function expansion can be applied giving a matrix of transformed inputs, $\Phi\left(X\right)$, where the expansion of every input is stacked row-wise. Analysis proceeds as before and the predictive posterior can be written down, substituting $\phi\left(\mathbf{x}_\star\right)$ for $\mathbf{x}_\star$ and $\Phi\left(X\right)$ for X.

$$\begin{aligned} p\left(y_\star \,|\, \phi\left(\mathbf{x}_\star\right), X, \mathbf{y}\right) &= \int p\left(y_\star \,|\, \phi\left(\mathbf{x}_\star\right), \Phi\left(X\right), \mathbf{y}, \mathbf{w}\right) p\left(\mathbf{w} \,|\, \Phi\left(X\right), \mathbf{y}\right) \mathrm{d}\mathbf{w} \\ &= \mathcal{N}\left(\sigma_n^{-2} \phi\left(\mathbf{x}_\star\right)^\mathsf{T} Z^{-1} \Phi\left(X\right) \mathbf{y}\,,\; \phi\left(\mathbf{x}_\star\right)^\mathsf{T} Z^{-1} \phi\left(\mathbf{x}_\star\right)\right) \\ Z &= \sigma_n^{-2} \Phi\left(X\right) \Phi\left(X\right)^\mathsf{T} + \Sigma_w^{-1} \end{aligned} \tag{2.7}$$

After some rearrangement, it can be shown that this distribution can be expressed only in terms of dot products of the input space, $\phi\left(\mathbf{x}\right)$. This allows the relationship

to be expressed through a reproducing kernel Hilbert space — that is the covariance kernel. The covariance kernel (also covariance function) is a function which computes the covariance between any two inputs, $k(\mathbf{x}, \mathbf{x}')$. This is commonly referred to as the 'kernel trick' and allows definition of the inner product matrix directly through the Reproducing Kernel Hilbert Space. The kernel trick appears across machine learning and statistics literature, for example, it is also used in the support vector machine [20].

Moving to a kernel representation of the relationship, of the inputs; under certain conditions, is equivalent to using an infinite basis function set, thereby removing the need to specify a basis function set. The GP can be interpreted as a prior over a family of functions where that family is the one described by the covariance kernel, this will be discussed further in Section 2.3. From this is it possible to define the GP fully in terms of its mean and covariance function despite being a flexible nonlinear model. One powerful interpretation of a GP is as the infinite dimensional extension of a multivariate Gaussian distribution.

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \tag{2.8}$$

When a finite number of training data pairs are observed, $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$, as before the inputs can be stacked into a matrix, X, and the outputs into a vector, $\mathbf{y}$. At this point the data is distributed according to a multivariate Gaussian, where the mean is given by the mean function of the GP and the covariance is defined by the covariance function between every data point, $K(X, X)$ — the elements of this matrix $K_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$. Since the observations are corrupted with some Gaussian white noise of a known variance a diagonal term must be added to the covariance matrix, the magnitude of which is the noise variance σ_n^2. The distribution is as shown in Equation (2.9).

$$\mathbf{y} \sim \mathcal{N}\left(m(X)\ ,\ K(X, X) + \sigma_n^2 \mathbb{I}\right) \tag{2.9}$$

When a new point is observed, where the inputs are known, but inference needs to be made over the output, this forms a joint distribution with the training data.

$$\begin{bmatrix} \mathbf{y} \\ f_\star \end{bmatrix} = \mathcal{N}\left(\begin{bmatrix} m\left(X\right) \\ m\left(\mathbf{x}_\star\right) \end{bmatrix}, \begin{bmatrix} K\left(X,X\right) + \sigma_n^2 \mathbb{I} & K\left(X,\mathbf{x}_\star\right) \\ K\left(\mathbf{x}_\star,X\right) & K\left(\mathbf{x}_\star,\mathbf{x}_\star\right) \end{bmatrix} \right) \tag{2.10}$$

Here, inference is made over the underlying latent function, $f_\star$, although by adding noise this can be extended to inference over values which will be observed, $y_\star$. This can be useful if using the prediction in a damage detection scenario. In order to recover the predictive distribution over $f_\star$, $p\left(f_\star \,|\, \mathbf{x}_\star, X, \mathbf{y}\right)$, it is necessary to condition $f_\star$ on the training data — this is equivalent to the conditioning of a joint multivariate Gaussian distribution — which gives rise to the following equation.

$$\begin{aligned} p\left(f_\star \,|\, \mathbf{x}_\star, X, \mathbf{y}\right) &= \mathcal{N}\left(\mathbb{E}\left[f_\star\right],\ \mathbb{V}\left[f_\star\right]\right) \\ \mathbb{E}\left[f_\star\right] &= m\left(\mathbf{x}_\star\right) + K\left(\mathbf{x}_\star,X\right)\left(K\left(X,X\right) + \sigma_n^2\mathbb{I}\right)^{-1}\left(\mathbf{y} - m\left(X\right)\right) \\ \mathbb{V}\left[f_\star\right] &= K\left(\mathbf{x}_\star,\mathbf{x}_\star\right) - K\left(\mathbf{x}_\star,X\right)\left(K\left(X,X\right) + \sigma_n^2\mathbb{I}\right)^{-1} K\left(X,\mathbf{x}_\star\right) \end{aligned} \tag{2.11}$$

Equation (2.11) is the key equation in GP regression models which allows the expectation and variance of the prediction to be expressed in closed form based solely on the relationship between the training and testing data through the kernel and the mean functions. Commonly, the mean function is considered to be zero so the equations can be rewritten removing these.

$$\begin{aligned} p\left(f_\star \,|\, \mathbf{x}_\star, X, \mathbf{y}\right) &= \mathcal{N}\left(\mathbb{E}\left[f_\star\right],\ \mathbb{V}\left[f_\star\right]\right) \\ \mathbb{E}\left[f_\star\right] &= K\left(\mathbf{x}_\star,X\right)\left(K\left(X,X\right) + \sigma_n^2\mathbb{I}\right)^{-1}\mathbf{y} \\ \mathbb{V}\left[f_\star\right] &= K\left(\mathbf{x}_\star,\mathbf{x}_\star\right) - K\left(\mathbf{x}_\star,X\right)\left(K\left(X,X\right) + \sigma_n^2\mathbb{I}\right)^{-1} K\left(X,\mathbf{x}_\star\right) \end{aligned} \tag{2.12}$$

Presented with the maths, it can be easy to miss the simplicity of what the GP is trying to achieve. Here, only the GP with a zero mean function is considered but, as shown, the extension to a parametric mean is trivial. Under the GP prior, before any data have been observed, the predictions of the model will be zero for every input with a variance equal to the signal variance of the kernel this is demonstrated in Figure 2.1.

In this plot, the variance is summarised by a confidence interval, however, interpreting

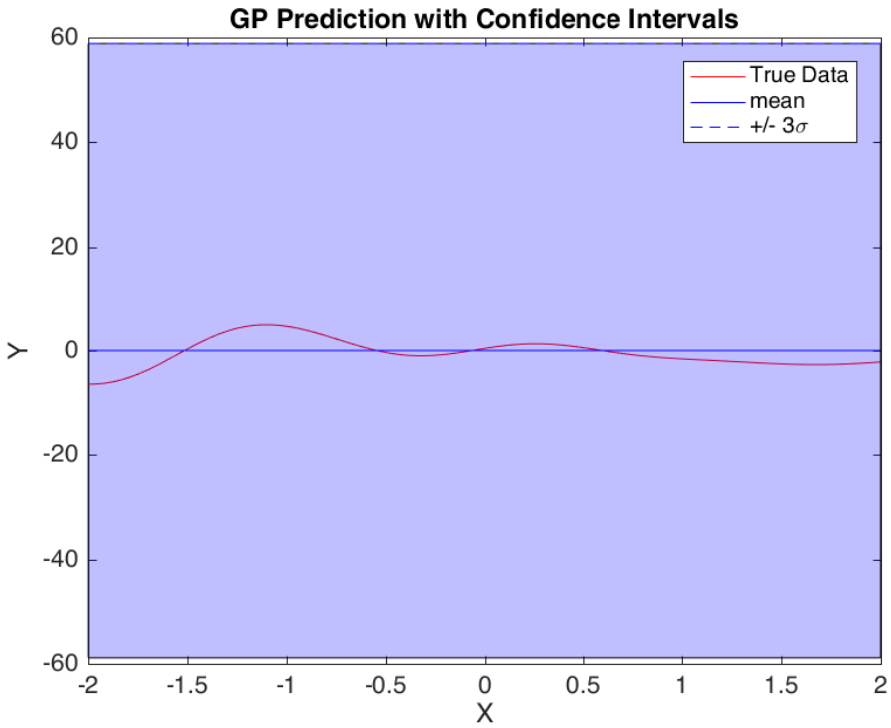


Figure 2.1: The Gaussian Process prior with a zero mean function is a Gaussian distribution across the output at every input with zero mean and variance equal to the signal variance of the kernel, σ_f^2

the GP in terms of the confidence intervals can obscure the actual understanding of what these represent. Since the GP is a distribution it is possible to draw random samples from it, as shown in Figure 2.2. These random draws, instead of being merely random variables, are in fact random functions over the input space. It can be seen that, under the prior, different functions drawn from the GP will be similar in terms of the functional form, which is defined by the kernel, but may not be good representations of the data which is being modelled, shown in red. It is also possible to make draws from the posterior distributions to create realisations of the possible functions that could have given rise to the observed data or possible functions for some new set of inputs.

The Bayesian nature of the GP is revealed as more data are added to the training process — if it was not obvious in the construction of the model! Figure 2.3 shows the effect of adding progressively more points to the training dataset. As more points are added the posterior (prediction) is more influenced by previously observed points and less resembles the prior. The effect could be described by the prediction being pinched around or pinned to the training data points. As the model is used to predict

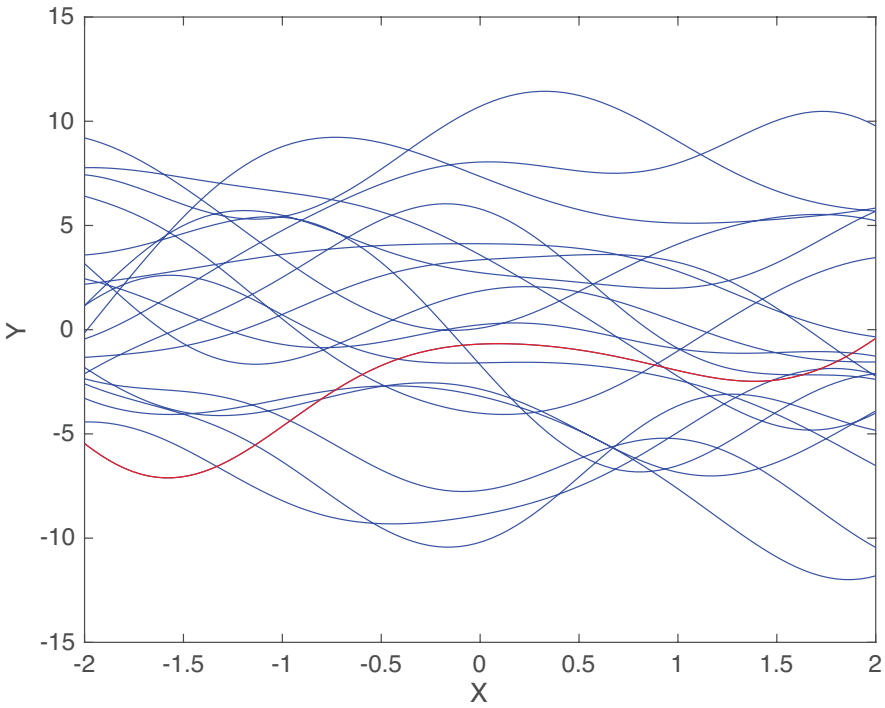
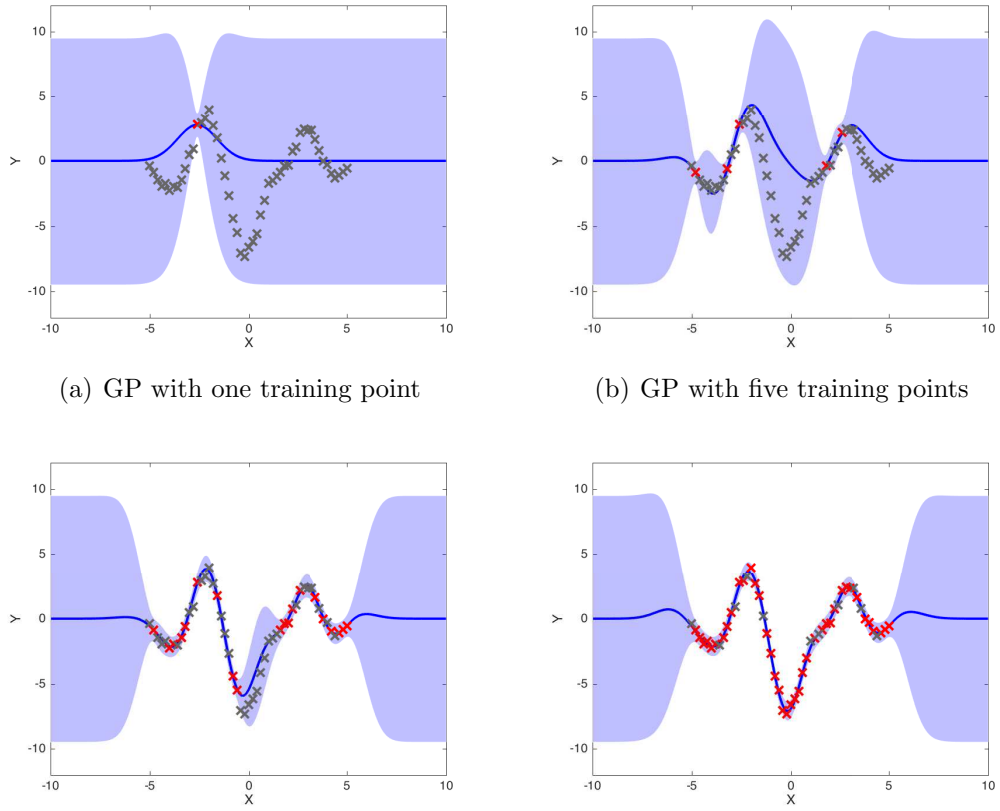


Figure 2.2: The interpretation of the GP as a prior over functions becomes clear when draws are made from the prior, here the lines is blue are all possible functions under the prior in Figure 2.1 with the red line being the true function.

further away from the observed training points the prediction returns to the prior and the confidence intervals increase, indicating that the model is less certain making predictions at those points.

One might find it helpful to imagine the prior distribution as a large (infinite) number of snakes wriggling around in a child's play tunnel (which represents some confidence interval), at points where data are observed it can be thought of as restricting the diameter of this tunnel to force all of the snakes together. Here the snakes are analogous to the potential latent functions, where the metaphor falls down is that the support of the predictive distribution at every point is across the whole real line and, therefore, the snakes should not be strictly constrained by the tunnel. Therefore, any real value is a potential sample from the distribution, however, many values become highly unlikely quickly as is normal in a Gaussian distribution. Likewise the functions exist over the whole real line — in the standard GP — for the inputs and (thankfully...) no snake is infinitely long.



(a) GP with one training point

(b) GP with five training points

(c) GP with twenty training points

(d) GP with forty training points

Figure 2.3: Figures showing posterior predictions of the GP as more training points are added. Discrete data observations are shown by crosses where the ones highlighted in red are used in the training process.

In the example shown, by the time forty training points are added to the GP, Figure 2.3(d), the functional form of the training data is very well captured and the model can successfully interpolate across that part of the input space, X. It is clear that the model is still unable to extrapolate; since there is no data to guide the GP the prediction returns to the prior. In fact, even if the model is interpolating, e.g. Figure 2.3(b) around $X = 0$, if the input space is not well covered in the training set, the GP is unable to make a confident prediction as to the expected output.

2.1.1 Assessing GP Performance

It is important to consider what is meant by a "good fit" in the context of a GP model. The general approach to quantifying model fit in SHM has been to consider some form of mean-squared error. This works well in a deterministic setting but

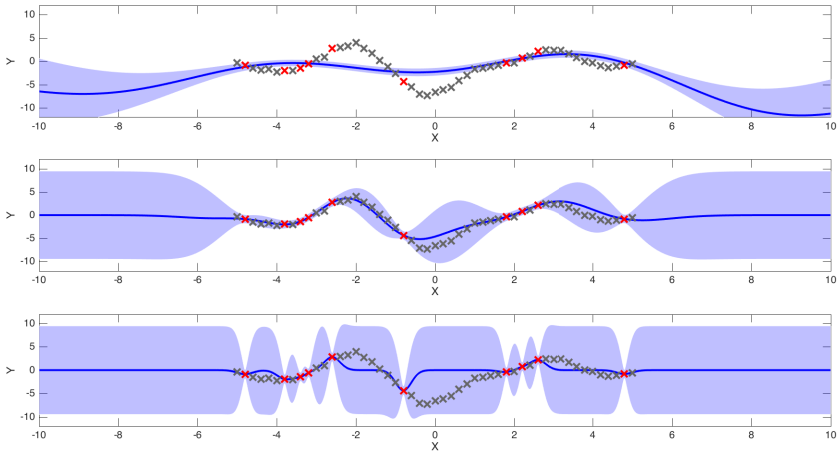


Figure 2.4: Plots of Gaussian process predictions for a function with training points shown in red and test points in grey, the prediction both in the mean, solid blue line, and the 3σ confidence intervals shaded in blue. Although all solutions it can be seen that the fit of some models is more desirable.

fails to quantify the quality of a probabilistic fit. Fuentes [50] discusses the uses of likelihoods as a model quality metric in SHM, here, the author agrees this is a sensible approach. It is possible to shown that the GP can be given a universal kernel such that it can fit a target function arbitrarily well given a prescribed error $\sigma_n^2 > 0$ [82]. However, this guarantee exists in the limit of infinite data — a condition which is not encountered in engineering! The problem remains then, how can a user determine if the model will return appropriate results for the function being modelled. Figure 2.4 shows three different predictions made by a Gaussian process on the same set of training and test data. It is clear that these fits are of varying quality, it will turn out that this is due to the optimisation of the hyperparameters which is discussed further in Section 2.5.

Although, heuristically, one could consider Figure 2.4 and choose a "best" model, by eye, it is useful at this stage to consider how to assess the quality of a prediction from a GP in the engineering context. It is important to be able to make quantitative comparison between different models which can inform future decision making.

$$\text{NMSE} = \frac{100}{N\sigma_{\mathbf{y}}} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2 \tag{2.13}$$

The use of variants of the mean squared error is a common choice for comparison of models, it is also useful to normalise this value to make it more interpretable. This can be done as in Equation (2.13), which causes the value of the normalised mean squared error (NMSE) to equal zero in the case of a perfect prediction and increase as the quality of the prediction decreases. If the value is equal to 100 then the prediction is comparable in quality to just taking the mean of the observations.

However, in the move towards probabilistic models, it is important to revisit the manner in which these models are assessed. For example, the NMSE is a comparison tool that relies only on the quality of the pointwise prediction, this way of assessing models fails to account for the uncertainty prediction in the model, as will all the mean squared methods.

By construction of the GP predictive equations, Equation (2.11), the likelihood of the measured data is available in closed form. For an individual prediction, the likelihood of the measured observation at that point is the likelihood of the normal distribution defined by the expectation and variance of that prediction.

$$p\left(y_\star \,|\, \mathbf{x}_\star, X, \mathbf{y}, \theta\right) = \mathcal{N}\left(\mathbb{E}\left[f_\star\right] \,,\, \mathbb{V}\left[f_\star\right] + \sigma_n^2\right) \tag{2.14}$$

It is worth highlighting that this likelihood includes the measurement noise added to the predictive variance of the GP. This assumes that the observation of the new test output $y_\star$ occurs under the same noise conditions as the training data. The likelihood of the prediction is bounded $(0, \infty)$ and allows for comparison of predictions between probabilistic models, it is not, however, immediately interpretable. It is a valuable method for comparing the quality of models and can be easily extended where there is more than one prediction point by considering the joint likelihood which is the product of Equation (2.14) for each predictive point $y_\star$. This is operating under the assumption that each prediction of the GP is independent of the others made in the same set, although it is possible to compute the cross covariances which give rise to other metrics.

Since the form of Equation (2.14) is Gaussian, this calculation is akin to using the Mahalanobis distance metric [12, 83] which is proportional to the log likelihood of the Gaussian distribution. For numerical reasons it is normally more sensible to work in the log space when computing likelihoods. For this reason using the Mahalanobis distance to assess the fit of the GP can be seen as very closely related to using the

posterior probability in Equation (2.14). The joint likelihood when computing a number of predictions can then be seen as related to the sum of the Mahalanobis distances. The combination of these tools allows for quantitative comparison of the quality of Gaussian Process model fits.

2.2 Learning Gaussian Processes for SHM

The GP has been presented as a powerful and flexible tool for solving regression tasks in general which can be readily applied SHM. The model has already begun to be used within SHM, for example see [53, 77]. At the base level of Rytters hierarchy [7], a key interest is minimising the number of false positives (alarms when a structure is undamaged) and false negatives (cases where the structure is damaged but the SHM system doesn't indicate this). Moving up the hierarchy, it is desirable to have accurate, low variance predictions from a regression model for the location or extent of damage, for example. This allows decisions to be made with confidence — for example, in a localisation task, knowing the location of damage with more accuracy has a significant cost benefit for the end user. The question is how does a user ensure this behaviour when using a GP in an SHM system?

Despite the GP being a non-parametric approach to regression modelling, there are a number of user choices which must be made and hyperparameters which must be determined. These include the selection of the covariance function, which governs which functional family the process is drawn from. Although work has previously discussed kernel selection, for example [84–86], this has not been framed in an engineering context and discussion regarding the optimisation of hyperparameters is normally avoided. The author believes — and aims to show — that these are key parts of the modelling process which should be considered a matter of practice. This chapter discusses and demonstrates the effect and importance of these two issues.

From Equations (2.11) and (2.12), it is clear that the kernel function plays a key role in the ability of the GP to make meaningful predictions. As discussed, the choice of kernel encodes the user's prior belief as to which functional family the data is drawn from, which includes belief about smoothness, periodicity, linearity, and other properties. For example, if the kernel function is chosen to be linear, the solution for Bayesian linear regression is recovered exactly, this is further explored in Section 2.3.

In addition to this, each kernel function is controlled by one or more hyperparameters which alter the behaviour of that kernel. It is necessary to determine the "correct" hyperparameters for any given model as these control key behaviours of the function, such as the frequency of repeating behaviour, in a periodic kernel, or the total magnitude of the function. This is the commonly adopted view of the hyperparameter estimation problem as a parameter estimation. The usual manner in which this is approached is as a Type-II maximum likelihood estimate, which is to maximise the model evidence $p(\mathbf{y} \mid X)$. A Bayesian approach to this problem is possible (and arguably preferable) through the application of hyperpriors, as will be discussed. Consider the typical kernel which is used when introducing GP models [74], the squared exponential (SE) kernel — also called Gaussian or exponentiated quadratic — which has the following form.

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left\{-\frac{\|\mathbf{x} - \mathbf{x}'\|_2^2}{2\ell^2}\right\} \tag{2.15}$$

where ℓ is the length scale of the process which governs the smoothness of the function. Since choosing the kernel is equivalent to making some prior assumptions about the type of function you are trying to model, the choice of an SE kernel encodes the belief that the function being modelled is smooth, infinitely differentiable, and nonlinear. Following the choice of kernel, the specific function being modelled is conditioned on two things, foremost the observed training data in a Bayesian manner, but also the hyperparameters of the kernel.

As an example, the SE kernel is defined by two hyperparameters, the signal variance, σ_f^2, and the length scale, ℓ. The signal variance affects the total scaling of the kernel, a larger signal variance is associated with signals which have higher variance, in fact this hyperparameter is actually the prior variance over the signal. That is, if no data are available for training, what the expected variance of the output would be around the mean of the process (usually zero).

Interpreting the length scale is slightly more difficult. In the case of the SE kernel it moderates the region of influence of each data point, or how close one data point needs to be to another in order to have an effect on the output. In Figure 2.5, the covariance $k(\mathbf{x}, \mathbf{x}')$ is plotted against the input distance for different length scales. It can be seen that by increasing the length scale parameter, points which are further away in the input space can still influence the outputs. This has the effect of

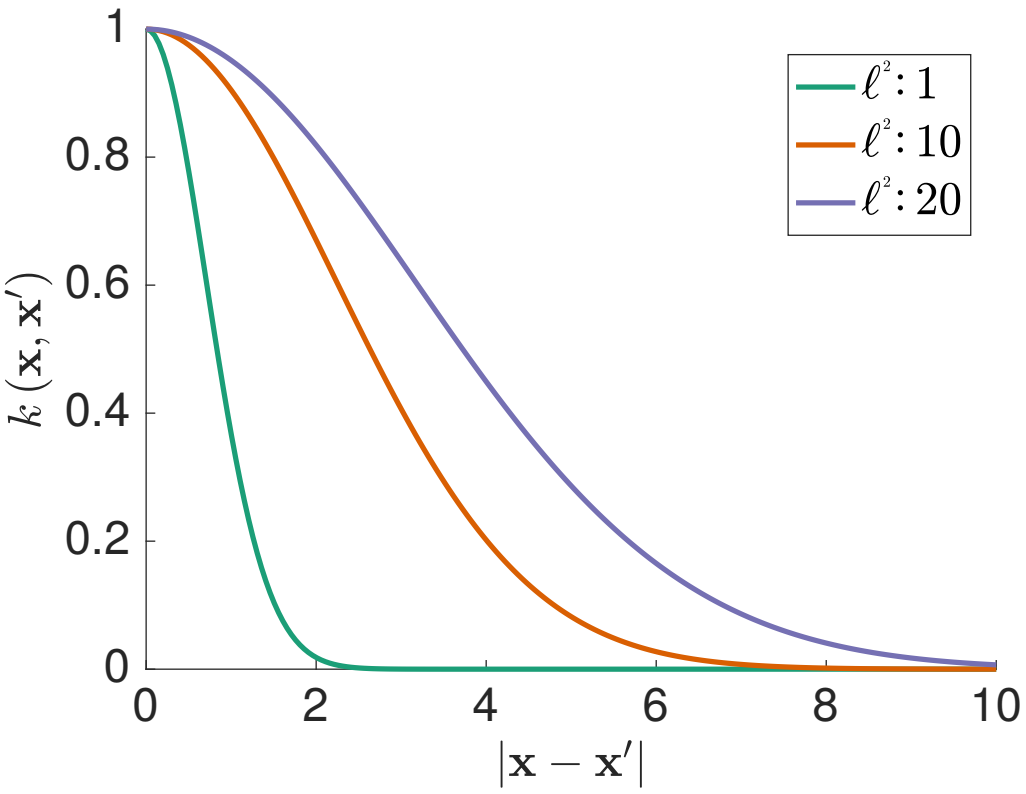


Figure 2.5: Increasing the length scale in the squared exponential kernel is shown to increase the region of influence. There is a higher covariance $k(\mathbf{x}, \mathbf{x}')$ at greater input distance $|\mathbf{x} - \mathbf{x}'|$ as the length scale increases

'smoothing out' the function and removing short scale variations.

Practically, this means that, if the lengthscale parameter is large, the function being modelled will be smoother and if the lengthscale is shorter, the process will be less smooth. This is demonstrated in Figure 2.6 where the function is defined by a GP where three different length scales are used but other hyperparameters are fixed. If the length scales are set too long, as in the top figure, then all the observed data is considered as noise and the GP will not make meaningful predictions as well as being overconfident when predicting far from observed data. If the length scale is set too short, as in the bottom figure, the prediction of the GP quickly returns to the mean of the process, in this case zero. The means that there is only a very limited range of values for which the GP will return a meaningful prediction. The question remains as to how to specify the 'optimal' set of hyperparameters.

Before discussing this selection of hyperparameters, it is worthwhile considering one more commonly used technique when specifying the kernel function. In cases where there exists more than one dimension in the input data, it can be useful to place

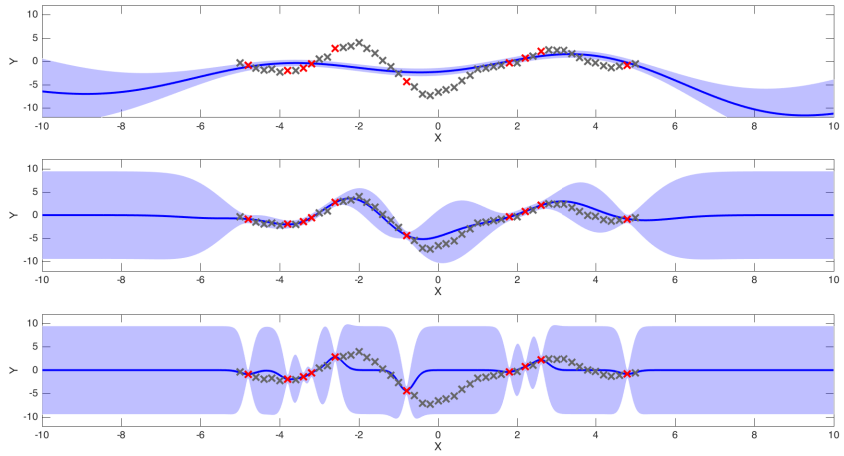


Figure 2.6: Predictions from GPs with varying length scales, training points shown in red and test points in grey, the prediction both in the mean, solid blue line, and the 3σ confidence intervals shaded in blue. It can be seen that changing the lengthscale alters the smoothness of the process.

separate lengthscales over each of these dimensions. This is usually referred to as an Automatic Relevance Determination (ARD) kernel where the single hyperparameter, ℓ, is replaced with a matrix, Λ, which is a diagonal matrix with the individual length scales along the diagonal. This allows different dimensions of the input to operate over different scales and also provides information as to which of the input dimensions are contributing to which behaviours in the output. It is important also to consider that the noise variance of the process is rarely known *a priori*, meaning it is normally included as an additional hyperparameter.

In the context of emulators — where a computationally cheaper model is used to replace an expensive simulator for calibration or Monte-Carlo analysis — Andrianakis and Challenor [87] discuss the role of the "nugget" — which replaces the noise term in a noise free emulator setting. For emulators (at least of deterministic simulators) it is assumed that the output of the simulator can be modelled as a noise free function — Gaussian Processes are one powerful approach to building these emulators. Within the Gaussian Process, the noise variance helps to stabilise the model numerically which poses a problem in the noise free emulator setting. For this reason a "nugget" term is usually set by the modeller. The paper of Andrianakis and Challenor [87] discusses how there can be two "modes" of solution (not related to dynamic modes) corresponding to either a Type-I or Type-II solution. The Type-I solution occurs

in the case of very low noise solutions to the GP (or when the nugget is set to be very small) in this mode the latent function passes through all the observed training points and leads to short lengthscales. This behaviour can be associated with "fitting the noise" in a model where the true underlying function is not captured in favour of attempting to explain all of the observed variability. This can be viewed as analogous to overfitting in a parametric model if the noise in the function is not actually very low. The alternative, Type-II solutions correspond to higher noise solutions where more of the function behaviour is explained as noise (or by the nugget in the emulator). In measured data, this can be a more realistic behaviour. However, the extreme of this is that the GP fails to model the underlying function instead modelling the functional behaviour as noise. This is a less dangerous misspecification of the model than fitting a Type-I solution when it is inappropriate. However, these solutions will lead to greater predictive uncertainty which can be just as costly if the models are used as part of a Risk & Reliability based inspection system through an increased need to inspect.

The question remains, how does a user ensure appropriate behaviour, for the available dataset. In this work, the author proposes two things, the first being to consider if a fully Bayesian solution is viable since understanding the full posterior over the hyperparameters will allow more informed decision making. Usually this is not possible practically. In that case, the second approach must be taken; to find an 'optimal' set of hyperparameters via optimisation. It is important to ensure robustness in the solution in one of a number of ways:

1. Begin gradient descent optimisation close to hyperparameters where there is good prior belief, i.e. low or high noise assumptions

2. If this cannot be done run multiple restarts of gradient descents to minimise the risk of local minima

3. Consider the use of a "global" optimisation scheme — such as the Quantum Particle Swarm Optimisation (QPSO)

4. Consider modifying the cost function of the optimisation

Within the machine learning community, most focus has been on achieving computationally feasible solutions to the Bayesian estimation problem for the hyperparameter

and not on modifying the optimisation of the parameters in the Type-II estimation. For examples of this see [88–90] The effect of these decisions will now be demonstrated.

2.3 Kernel Selection in Engineering

It has already been discussed that that the choice of covariance function/kernel plays a central role in the performance of the GP — since this and the mean function fully define the GP. It is not just the optimisation of the hyperparameters that plays a role in this, although this will be shown to be a key ingredient in the use of the GP for SHM. Specifically, the choice of optimisation scheme can be shown to be important especially as the dimension of the hyperparameter vector increases. It remains to address the problem of choosing which covariance function should be used for a given problem. This choice, as mentioned, encodes the prior belief of the modeller regarding the functional family from which the data are drawn. For instance the choice of a linear kernel restricts the GP to modelling only linear functions — in fact the solution to a Bayesian linear regression is recovered identically.

It may be considered a simple task then to inspect a function and decide if the data appear to be generated by a linear process, periodic process, nonlinear process, etc. However, this is complicated somewhat by the realisation that most of the time in SHM the function being considered is modelled as the effect of a number of input variables. For example, it may be the case that the output data are periodic, however, if this is modelled as a function of some other periodic signal, the relationship may in fact be linear or some other relationship entirely. This requires a change in mindset to perform kernel selection — one should not consider the properties of the observed function in time (unless fitting a temporal GP!) but instead in the input-output/regression space where the regression is being performed. A key example of this is when using a Gaussian Process Nonlinear Auto-Regressive model with eXogeneous inputs (GP-NARX) model, the regression space here becomes the correlation between the lags being considered and the output.

It may be clear at this point that this is a tall order. How does an engineer visualise relationships in these, often, high dimensional regression spaces? If it is not possible to visualise the correlations, it becomes exceedingly difficult to make meaningful modelling choices regarding the covariance function. Here, a number of common

choices for covariance functions are presented, alongside some discussion regarding their combination and approaches for automating the model selection problem in the GP.

Choices of Covariance Function

There are a number of conditions that must be fulfilled to produce a valid covariance function, which must give rise to a symmetric positive semi-definite covariance matrix — this can also be shown by proving that the covariance function satisfies Mercer's condition [74]. The use of covariance functions and corresponding covariograms has been well established within the geostatistics kriging community [91, 92] which the GP is closely linked to. A complete survey is excessive for this work but a number of the most common covariance functions are presented here.

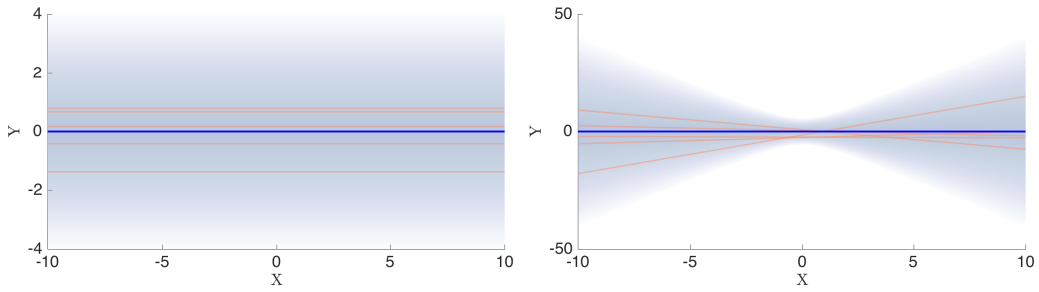
The logical starting place is the linear covariance function defined as below.

$$k\left(\mathbf{x}, \mathbf{x}'\right) = a \ \ \mathbf{x}^{\mathsf{T}} \cdot \mathbf{x}' + b \tag{2.16}$$

The affine form is shown here which contains two hyperparameters the slope a and offset b. It is also possible to define the kernel without the offset hyperparameter, and to use addition with a constant kernel to achieve the same result, where the constant kernel would be defined as,

$$k\left(\mathbf{x}, \mathbf{x}'\right) = c \tag{2.17}$$

As will be shown, combinations of kernels via products or sums are also valid kernels and this is a useful property to create more expressive covariance functions. A helpful way to understand what the kernel is doing is to leverage the generative nature of a Gaussian Process model. That is, it is possible to sample from the prior over the function space an plot potential outputs from that kernel before any data have been observed. The following plots of the priors for different kernels have the mean highlighted as a blue line with the distribution over the possible function values shown up to 4σ by the shaded region and 5 possible realisations of the function shown in orange. Figure 2.7 shows the result of sampling from the prior distribution over a Gaussian Process with a constant kernel and linear kernel respectively. The



(a) Constant kernel (b) Linear kernel

Figure 2.7: Realisations from the prior of the Gaussian Process for the constant kernel in (a) and the affine linear kernel in (b) the shaded area indicates the distribution, the blue line the zero mean prior, and the orange lines are individual realisations (samples) from the prior.

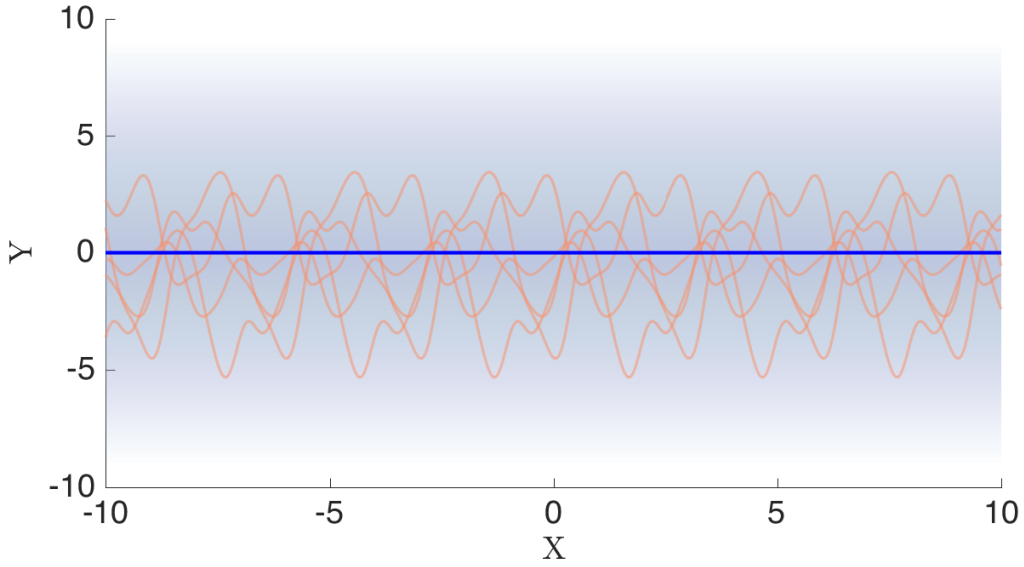
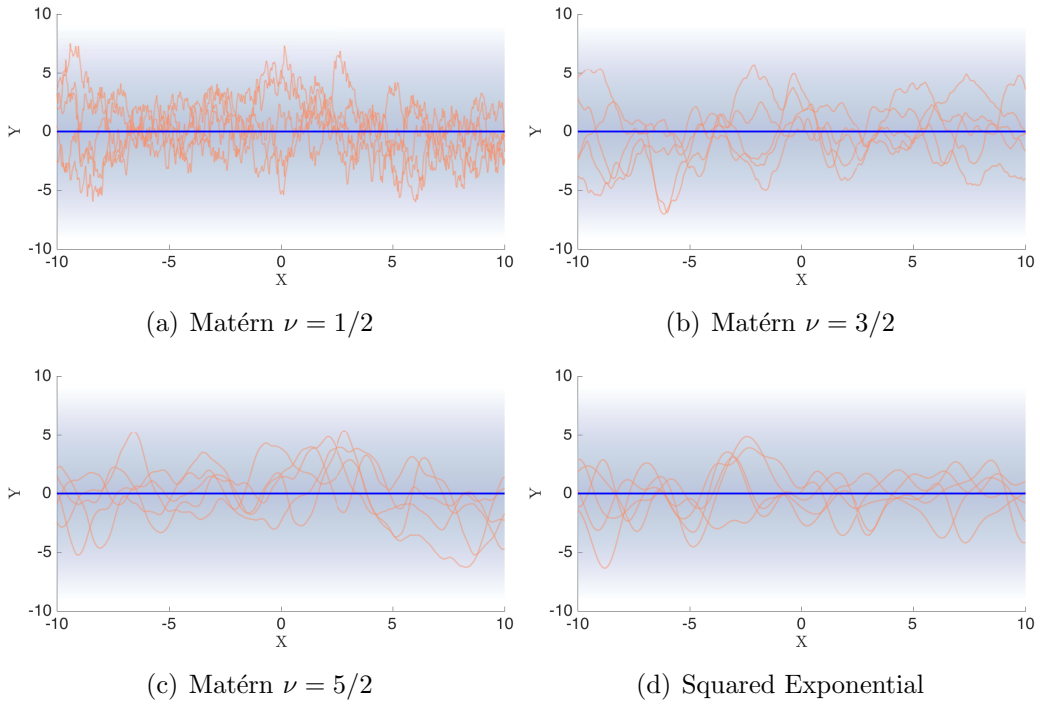


Figure 2.8: Realisations from the prior of a Gaussian Process with a periodic kernel.

constant kernel in (a) of Figure 2.7 is seen to only generate constant values of the output function, the exact value of which is conditioned on any observed data. This on its own is not a very useful kernel but when combined with other kernels can allow a richer set of functions to be modelled. The affine linear kernel also behaves very much as expected and can generate linear output functions with an offset which recovers exactly the solution from a standard Bayesian linear regression.

Analogously to the Bayesian linear regression, it is simple to extend this type of model to higher order polynomials. The polynomial model, however, can be too restrictive. Remember this is what motivated the adoption of Gaussian Processes



(a) Matérn $\nu = 1/2$ (b) Matérn $\nu = 3/2$

(c) Matérn $\nu = 5/2$ (d) Squared Exponential

Figure 2.9: Realisations from the prior of the Gaussian Process for radial basis kernels. Matérn kernels of increasing smoothness parameter are shown in (a), (b), and (c); the infinitely smooth squared exponential is shown in (d)

initially! Additionally, signals encountered within SHM can be dynamic, in this case it is common to see periodic behaviour for example in the impulse response of a resonant structure. It is useful to be able to prescribe periodicity in the prior form of the model. Again this is possible to do through the choice of a periodic kernel defined as,

$$k\left(\mathbf{x}, \mathbf{x}'\right) = \sigma_f^2 \exp\left\{\frac{-2\sin\left(\pi \cdot |\mathbf{x} - \mathbf{x}'|/p\right)^2}{\ell^2}\right\} \tag{2.18}$$

where p is s hyperparameter related to the frequency of the periodic signal and ℓ to the smoothness of the periodic component. Realisations from a Gaussian Process prior with this kernel specification are shown in Figure 2.8. The uncertainty shown is seen to be very similar to that for the constant kernel. However, when viewing realisations of the prior process, this prior is shown to generate realisations of periodic functions. This can be especially useful as it allows extrapolation beyond the training data range if the model is known to be strictly periodic.

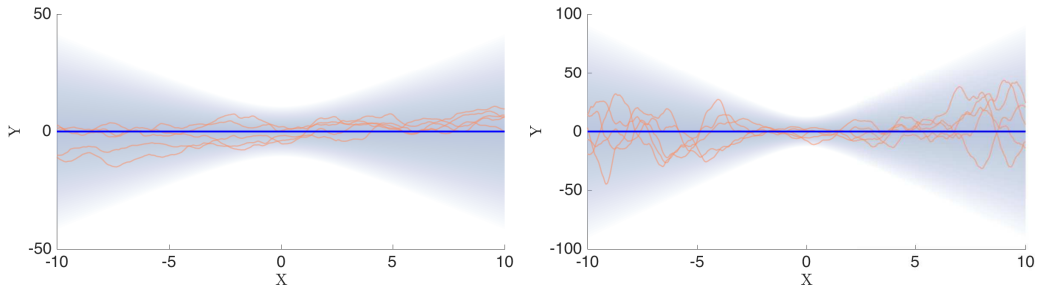
Unfortunately, this is normally too strict an assumption to make and priors are chosen based on less definite assumptions about the functional form than those of choosing a polynomial or periodic kernel. For this a general nonlinear kernel can be chosen which imposes far weaker assumptions about the functional form. A popular family of kernels for this type of function are the radial basis function family. To describe the radial basis family it is useful introduce two pieces of terminology, the first is that a kernel is said to be stationary when the covariance is a function of the difference of two inputs not their absolute values. The second is isotropy, a kernel is isotropic when it is invariant to translation or rotation of the inputs, i.e. the order of the inputs is unimportant. A general radial basis kernel is the Matérn kernel which is defined as,

$$k\left(\mathbf{x},\mathbf{x}'\right)=\sigma_f^2\frac{2^{1-\nu}}{\Gamma\left(\nu\right)}\left(\sqrt{2\nu}\frac{r}{\ell}\right)^{\nu}\mathcal{K}_{\nu}\left(\sqrt{2\nu}\frac{r}{\ell}\right) \tag{2.19}$$

This covariance function is governed by a number of hyperparameters: the signal variance σ_f^2, the lengthscale ℓ, and the smoothness parameter ν. For notational convenience the distance between the two inputs has been simplified to $r=|\mathbf{x}-\mathbf{x}'|$. The covariance function contains two additional functions the gamma function $\Gamma\left(\cdot\right)$ and the modified Bessel function of the second kind $\mathcal{K}_{\nu}\left(\cdot\right)$ which has ν degrees of freedom. This is a some what complicated covariance function but with appropriate choice of the smoothness parameter *a priori* the equation simplifies significantly. An added advantage of this is removing the need to compute the costly Gamma function! This happens when setting $\nu=p+1/2$ for positive integer values of p or zeros — that is $p\in\{\mathbb{Z}^+,0\}$.

If $p=0$ the Matérn kernel generates realisations from an Ornstein-Uhlenbeck process [74], which is the continuous time equivalent to a random walk. This kind of very rough random walk behaviour can be seen in the realisations of the GP prior shown in Figure 2.9 (a). Not many signals encountered in SHM can be well described by this rough a kernel — indeed, this can lead to modelling the noise process in the measured data rather than the underlying functional form.

By increasing the value of p and consequently ν, the smoothness of the function increases, and as $\nu\to\infty$ the kernel tends towards the squared exponential kernel. This progression is shown in the other three frames of Figure 2.9. The squared exponential kernel is defined by the covariance function,



(a) Additive kernel (b) Product kernel

Figure 2.10: Realisations from the prior of a Gaussian Process where the kernel is defined by the combination of a Matérn kernel with $\nu = 3/2$ and a Linear kernel. The additive kernel is shown in (a) and the product kernel in (b).

$$k\left(\mathbf{x}, \mathbf{x}'\right) = \sigma_f^2 \exp\left\{-\frac{\|\mathbf{x} - \mathbf{x}'\|_2^2}{2\ell^2}\right\} \tag{2.20}$$

The squared exponential is a popular kernel since it imposes the prior of a smooth function that is infinitely differentiable. Its form is very similar to that of the Gaussian distribution and, therefore, many of the same mathematical manipulations can be conducted. That being said, Stein [92] argues that the squared exponential will impose an unrealistic smoothness assumption for naturally occurring functions. The experience of the author falls in line with this analysis and the use of the Matérn 3/2 function is a personal preference but one which is based in experience and this important result.

It is also possible to combine kernels in such a way that the combination is also a valid kernel. This allows the use of the basic kernel forms as a number of building blocks when wanting to express more complicated functional forms or impose more nuanced constraints. There are two methods in which the kernels can be combined; the first to add kernel functions and the second to take the product of two kernel functions. Since these produce new valid kernels, they can of course be repeatedly applied, for example the covariance function can be defined as the summation of two different products. Interpreting the effect of these combinations is a little more difficult than understanding the behaviour of the individual kernels and is best achieved pictorially.

Figure 2.10 shows the prior of a GP given combination of a linear kernel and a Matérn 3/2 kernel, both additively and as a product kernel. Consider the realisations

shown in frame (a) for the additive kernel which is defined by,

$$\begin{aligned} k\left(\mathbf{x}, \mathbf{x}'\right) &= k_{\text{Linear}}\left(\mathbf{x}, \mathbf{x}'\right) + k_{\text{Matérn 3/2}}\left(\mathbf{x}, \mathbf{x}'\right) \\ &= \left(a\, \mathbf{x} \cdot \mathbf{x}' + b\right) + \sigma_f^2 \left(\left[1 + \frac{\sqrt{3}r}{\ell} \right] \exp\left\{ -\frac{\sqrt{3}r}{\ell} \right\} \right) \end{aligned} \tag{2.21}$$

Inspecting the realisations from the prior, it is clear that this combination of the kernels results in the function being modelled as the nonlinear function on top of the linear trend. This can be useful if it is known, for example, that the process being modelled is broadly linear with some deviations from that line. Turning attention to frame (b) the product kernel is defined as,

$$\begin{aligned} k\left(\mathbf{x}, \mathbf{x}'\right) &= k_{\text{Linear}}\left(\mathbf{x}, \mathbf{x}'\right) \times k_{\text{Matérn 3/2}}\left(\mathbf{x}, \mathbf{x}'\right) \\ &= \left(a\, \mathbf{x} \cdot \mathbf{x}' + b\right) \times \sigma_f^2 \left(\left[1 + \frac{\sqrt{3}r}{\ell} \right] \exp\left\{ -\frac{\sqrt{3}r}{\ell} \right\} \right) \end{aligned} \tag{2.22}$$

The realisations seen are seen to be very different from the additive case. The product kernel is seen to behave like an amplitude envelope. In this case, the variance of the nonlinear function described by the Matérn kernel grows according to the linear covariance. The use cases for this combination are more difficult to imagine, but can allow modelling of complexly varying signals.

As previously mentioned, although it is relatively simple to consider the prior implications for the choice of covariance function, the difficultly is normally in understanding the high dimensional input/output spaces that can occur. The main problem then is, can the need for human intervention be removed from the kernel selection process? If so then the need to understand these complicated spaces is removed. If not then, in the absence of a complete picture of the space, can the relationships be either visualised in a lower dimension, understood via physical knowledge of the process, or both?

2.3.1 Kernel Selection Techniques

First the problem of automated kernel selection is considered. Since the marginal likelihood of the process is implicitly conditioned on the hyperparameters it is also

implicitly conditioned on the model choice — in this case the kernel. That is, it can be written, $p(\mathbf{y} \mid X, \theta, \mathcal{M})$, where the kernel choice is encoded in the choice of model $\mathcal{M}$. This notation is normally omitted for neatness when writing down the equations of the GP model but is useful when considering how to conduct kernel choice. Of course, calculation of this value is identical to the formula presented in Equation (2.26) except here the conditioning is explicit.

Using this knowledge it is possible to set up the model selection task as a Bayesian hypothesis test using Bayes factors [93, 94]. This approach has received some criticism [35, 95]. The main charge levelled against it is the departure from the Bayesian philosophy of inferring posterior distributions in favour of an absolute model selection — the hypothetico-deductive method. These abstract arguments regarding semantics and philosophy, although important (as anyone who has discussed Bayes with the author will know), are often not key drivers for engineering problems. In fact, for the engineering use case, a more pressing issue with automated model selection occurs — computational expense. The fully Bayesian solution is not considered here at all due to the impossibility (in closed form or due to the massive computational load) of marginalising all possible model forms.

In order to conduct model selection via the Bayes factor one has to calculate the marginal likelihood for all possible models, which here is every kernel and combination of kernels which could be valid. This set becomes large very quickly. Additionally, to compute this factor one should marginalise out the model parameters which, in this case, would be the hyperparameters of each kernel. This makes sense since the aim is to compare the posteriors of the models conditioned only on the data. However, as discussed this is normally a costly exercise, where the posteriors must be approximated.

Bearing this in mind, currently, the most feasible method for kernel selection in the Gaussian Process model remains to rely on expert knowledge/intervention. The primary criteria for this should be to choose kernels which encode at least approximate belief regarding the physical processes. However, the kernels should also be chosen to be flexible enough to capture fully the information contained in the data. For instance, if a trend is known to be approximately linear it would be prudent to use a more flexible kernel such as the sum of a linear kernel with a Matérn kernel to capture any nonlinearity which may not be obvious. It can be useful to attempt to visualise pairwise relationships between each input and the output of the model to help with this choice. In addition, another useful trick is to separate different input

dimensions into their own kernels. If one variable as a strong correlation with a particularly clear form, it can be useful to separate it into its own kernel. It can also help generalisation of the model to separate inputs whose effects on the output are not coupled into their own kernels. This formalises the knowledge that the output might depend on both input one and input two but the combination in which they appear does not matter. That is, the effect of one output is not related to the value of the other.

2.4 Choice of Cost Function

Within engineering it has been common to fit models based on minimising an error of some type, normally related to the mean-squared error in some way, for example see the learning of the Bouc-Wen system in [96]. If the model form is known to be correct, this works well in some circumstances. It would seem credible that models with lower error perform better. Of course, it is worth bearing in mind the problems with overfitting that can occur if the model form is incorrect. In an effort to combat this (especially when fitting black-box models not based on physical understanding) regularisation techniques are used to develop a *trade-off* between model fit and model complexity [20]. The choice of the cost function for an optimisation can, therefore, have a significant effect on the quality of the model fit and its ability to generalise. An extreme example is, if the cost function chosen is not sensitive to the (hyper)parameters being learnt then it will be impossible to perform the optimisation. This will be the case when the cost function has no dependence on those (hyper)parameters being learnt. It will be shown that this would be the case if learning a Gaussian process via a simple mean-squared error cost. It is important, therefore, to consider also what available cost functions are available when training Gaussian Processes. In literature, although alternatives have been discussed [74], it appears that most Gaussian proceess models are trained based upon the negative log marginal likelihood.

2.4.1 Mean-Squared Error

Traditionally, minimising the pointwise error has been a popular way to learn parametric models of systems — using the mean squared error is the popular least

squares approach, for example see [20]. The standard least squares error is written as,

$$error = (\hat{\mathbf{y}} - \mathbf{y})^{\mathsf{T}} (\hat{\mathbf{y}} - \mathbf{y}) \tag{2.23}$$

where, $\hat{\mathbf{y}}$ is the model prediction for some measured "true" values $\mathbf{y}$. It is also popular to use variants of this such as the mean squared error by taking the mean of the above or a normalised version such as the one presented in Equation (2.13). Which particular variety of error used is, generally, not important except to improve numerical stability and help with interpretation of results. The weakness in all these approaches is the deterministic nature of the cost function.

Classically, this has not been a problem since many engineering models have been deterministic. However, it is the opinion of the author that moving to a probabilistic paradigm requires a shift in how the quality of models is understood. Since the Gaussian Process returns predictions over the distributions of possible function values, it is reductive to consider only the mean fit when assessing the model. It is true that this mean value is still an important method for communicating the predictive capability of the model but it should not be used in isolation and its use as a cost function in optimisations may lead to poor models as the cost function is not sensitive to the variance of the model. Therefore, it is not optimal to use a mean-squared error (or variant thereof) as the cost function for a Gaussian Process — or indeed any probabilistic model optimisation. Thankfully, the author has not seen this approach in literature although there might be a temptation apply these techniques if one were moving from a deterministic model to the GP for regression modelling.

2.4.2 $z-$ Score

Consider instead the use of a z-score as a cost function for optimising Gaussian Process hyperparameters. It is known that the posterior predictive distributions from the GP regression model will be Gaussian and that the z-score can be used to calculate how close a given point is to a pre-defined Gaussian distribution — i.e. how many standard deviations from the mean a given value is. The calculation of the z-score is a simple formula,

$$z = \frac{y - \mu}{\sigma} \tag{2.24}$$

here z is calculated as the residual between the measured point y and the mean, μ which is the expectation of the predictive posterior of the Gaussian process, and the standard deviation of the Gaussian process predictive posterior as defined in Equation (2.11) or Equation (2.12). This is valid since the posterior of the Gaussian Process is a Gaussian distribution by construction. Here a cross-validation scheme has to be adopted where certain points are predicted from other available data — the gold standard being a leave-one-out cross validation. However, usually a grouped cross validation is more feasible due to the high computational cost of leave-one-out. This can be achieved via a hold-out validation set or through a k-fold cross validation which allows use of all the available data in training.

The score currently is only defined for a single predictive point; it is desirable to extend this to multiple predictive points. The simplest extension would be to consider taking individual z-scores for each predictive point and then combining these, for example by taking the mean. A number of alternative approaches are presented in [97], the first of which is moving to the multivariate analogue to the z-score which is the Mahalanobis distance. This has already found extensive use within SHM for outlier/novelty detection [5, 6, 13, 16, 51, 98, 99]. Here the full predictive covariance between all of prediction points is used to calculate the distance of the measured data from the predicted ellipsoid. Bastos and O'Hagan [97] suggests that using a pivoted Cholesky decomposition of the covariance will be of more benefit by decoupling the errors of individual points to allow for more informative diagnoses. Approaches based on Mahalanobis type assessments move towards assessing a Gaussian log likelihood of the predictions, it is sensible in a probabilistic model to take this further and to consider the likelihood of the process. Since the use of a Mahalanobis distance as a cost function is somewhat of a 'halfway-house' between the deterministic approach of the mean-squared error and the probabilistic approach of considering likelihoods, it is not used for learning in Gaussian process literature. Instead its use is restricted to assessment of the model fit as in [97].

2.4.3 Negative Log Marginal Likelihood

The cost function recommended in most introductions to the Gaussian Process including that of Rasmussen and Williams [74] is the negative log marginal likelihood and this convention has been followed thus far in the work presented here. The argument for using this cost function is that the marginal likelihood provides a trade-off between the model complexity and the model fit, referred to as the Bayesian Occam's Razor [100–102]. Consider the terms in Equation (2.26) which is reproduced here in the negative log form, this form is the cost function for the minimisation.

$$-\log p\left(\mathbf{y} \mid X, \theta\right) = \underbrace{\mathbf{y}^{\mathsf{T}}\left(K + \sigma_n^2 \mathbb{I}\right)^{-1}\mathbf{y}}_{\text{Model Fit}} + \underbrace{\frac{1}{2}\log\det\left(K\left(X, X\right) + \sigma_n^2\mathbb{I}\right)}_{\text{Model Complexity}} + \underbrace{\frac{N}{2}\log\left(2\pi\right)}_{\text{Constant}} \tag{2.25}$$

The equation shown in Equation (2.25) has the terms of the negative log marginal likelihood annotated. There is a trade-off when learning the hyperparameters between the model fit and the model complexity which exists by construction. The term relating to model fit will be familiar to readers in SHM as having the same form as a Mahalanobis distance where the covariance is defined by the covariance matrix of the GP with the noise added to the diagonal. The term moderating the model complexity is governed by the determinant of the covariance matrix. It is worth noting that if the size of the training set changes it is difficult to us this likelihood value as the cost function for optimisation since the constant term (and in fact the other two terms) is related to the number of training points. This is a subtlety that may be lost if one were to attempt to perform for example lag selection where the number of training points will change with differing scenarios.

It is worth nothing that this approach is an example of what is referred to as a Type-II maximum likelihood solution. That is that the hyperparameters are learnt by maximising the *marginal* likelihood of the process. This is where the latent function values of the model have been marginalised out of the cost function analytically, this type of maximum likelihood should be more robust to overfitting. For this reason, the use of this type of cost function should, using the Bayesian Occam's Razor, give rise to the minimally complex model regardless of the prior. However, this is still dependent on two important assumptions. The first that the model selection is correct, i.e. the choice of kernel; secondly that the dataset which is being used

to calculate the marginal likelihood is representative of the full dataset which is being modelled. That is, that the data are drawn from the same distribution as the function being modelled and that there are sufficient data to fully cover the space over which the function needs to be modelled.

2.4.4 Predictive Probability

The choice of the marginal likelihood seems like a sensible cost function for learning Gaussian Process hyperparameters — it is useful to have the Bayesian Occam's Razor to automatically choose the minimally complex model. However, this assumption is predicated on the belief that the training dataset is a representative sample from the full distribution. It has been established that the GP will only be able to make predictions 'close' to where there is observed training data (mediated by the lengthscale of the process). Unfortunately, within engineering, it is rarely possible to guarantee that sufficient data has been collected to cover the input space of the function in an engineering context. For instance if trying to regress data where the relationship is governed by a sine wave, if the sample frequency of the data acquisition is equal to the frequency of the sine wave then the Gaussian Process will model the function as a constant value with high confidence. This is as much a consideration in design of data collection systems for SHM but raises an important point. This is by no means a new problem within SHM (or engineering in general), although applying this type of model can highlight some inadequacies in a collected dataset. This is especially true if modelling dynamic processes in a GP-NARX scheme, it is possible to fit models with very well model the observed training data but fail to generalise to additional data. There is a subtle different between this and overfitting behaviour. In overfitting, a function which has more parameters than necessary is misidentified during training. Here, the Gaussian Process has learnt a minimally complex model for the data observed but this data is not representative of the true process being modelled.

In an effort to overcome this it is possible, rather than considering the marginal likelihood of the process, to consider the predictive probability of the process. That is the posterior likelihood of some unused data given a model trained on the rest of the training set. Again this can be done in a leave-one-out methodology or using hold out sets [74]. The author has found this can help, especially when fitting dynamic models this can help to understand how well the training data actually describes the

underlying function being modelled. The predictive probability is given by calculating $p(\mathbf{y}_\star \mid X_\star, \mathcal{D})$ which is the Gaussian posterior likelihood given by Equation (2.11) or Equation (2.12). Practically, the author has observed that the risk of not fully covering the input space is greater when fitting dynamic models. It is recommended, therefore, that if this is a worry the predictive posterior likelihood may perform better than the marginal likelihood as the cost function for optimisation of the GP hyperparameters. However, care should be taken if it is believed that the training data does not fully represent the function being modelled. In this case it would be better to attempt to collect a more representative training set before modelling the data or pay further care to the model selection to reduce the limitations of an incomplete training set.

2.5 Hyperparameter Optimisation

The GP hyperparameters are most easily and commonly determined via a Type-II Maximum Likelihood problem[1] where the marginal likelihood of the observed training outputs ($\mathbf{y}$) conditioned on the observed inputs (X) and hyperparameters ($\Theta = \left[\sigma_f^2, \operatorname{diag}(\Lambda), \sigma_n^2\right]$) is optimised with respect to the hyperparameters[2]. Perhaps the most desirable property of this optimisation problem is the automatic balancing of the model complexity and model fit that is expressed in the marginal likelihood, this is usually referred to as the Bayesian Occam's Razor [100]. In fact, the optimisation of the GP model in this way ensures that the minimally complex model is used to explain the data [101]. Like many problems in engineering this reduces the learning process to a multi-dimensional optimisation problem.

The marginal likelihood of the Gaussian process is defined as the likelihood of the process marginalised with respect to all model predictions as shown in Equation (2.26). This is the likelihood of the outputs of the model given the inputs and hyperparameters but not conditioned on the latent function values. Here $\mathbf{f}$ represents the noise free training outputs, which are the underlying latent function being modelled by the GP.

[1] The Type-II maximum likelihood refers to a maximisation of the model evidence term in Bayes' theorem which is the denominator. This is equivalent to maximising the integral of the likelihood and the prior which marginalises out the model outputs/parameters and is more robust to overfitting [100].

[2] The operator $\operatorname{diag}(\cdot)$ corresponds to taking the diagonal terms of a matrix.

$$p\left(\mathbf{y} \mid X, \Theta\right) = \int p\left(\mathbf{y} \mid \mathbf{f}, X, \Theta\right) p\left(\mathbf{f} \mid X, \Theta\right) \mathrm{d}\mathbf{f}$$
$$\ln p\left(\mathbf{y} \mid X, \Theta\right) = -\frac{1}{2}\mathbf{y}^{\mathsf{T}}(K\left(X, X\right) + \sigma_n^2 \mathbb{I})^{-1}\mathbf{y} - \frac{1}{2}\ln|K\left(X, X\right) + \sigma_n^2 \mathbb{I}| - \frac{n}{2}\ln 2\pi \tag{2.26}$$

The log likelihood is normally used as it can be more numerically stable. The problem is normally posed as a minimisation problem over the negative of this log marginal likelihood, so that the optimal set of hyperparameters, $\hat{\Theta}$, is found as:

$$\hat{\Theta} = \arg\min_{\Theta} \left\{ -\ln p\left(\mathbf{y} \mid X, \Theta\right) \right\} \tag{2.27}$$

It is worth noting, however, that the 'true' problem of interest (in a Bayesian sense) is determining the posterior distribution over the hyperparameters, $p\left(\Theta \mid \mathbf{y}, X\right)$, given the data. Using Bayes theorem, it can be seen that optimisation of the cost function shown in Equation (2.26) is equivalent to the maximum likelihood solution but inference could be extended to determine the *maximum a posteriori* (MAP) estimate or indeed the true posterior. For the case of determining the true posterior, exact inference is not possible so approximate methods must be employed, such as Markov-Chain Monte-Carlo [32], Sequential Monte-Carlo [88, 103], Adaptive Importance Sampling [89] or variational Bayes [33].

However, by far the most common mechanism for training GPs is the use of the minimisation of the negative log marginal likelihood, in fact, this is the only approach used in engineering literature both within SHM and beyond. The author does not feel that adequate attention has been given to the effect of different learning procedures for GP hyperparameters and this is explored here. Namely the choice of optimisation scheme as the dimension of the hyperparameter space grows. It is known from general optimisation problems that as dimensionality increases the issue of getting 'stuck' in local minima increases — i.e. it becomes harder to find the global minimum of the objective function.

The novel contribution here is to highlight that the current use of gradient based optimisation algorithms can lead to sub-optimal solutions since the cost function has many local minima. Also, to explore the dependency of the performance of the GP on the optimisation procedure as dimensionality increases, three experiments are

conducted and the predictive performance of the GP is considered following training with each strategy. Also investigated is the differences in computation time for each method. This investigation focuses on three datasets, low and high dimensional input synthetic datasets are tested, then a data set from a Tuncano TMK1 Trainer aircraft is considered. This represents a more realistic SHM example — although not an example from offshore, this case study highlights some important general points for engineering applications.

2.5.1 Effect of Optimisation Scheme

So far, the two key considerations in applying a GP to an engineering problem have been introduced. The first of these, the optimisation of the covariance function hyperparameters is now explored in more detail. Specifically, the effect of the choice of optimisation scheme on the predictive performance of the GP is considered. When optimising over the marginal likelihood of the process, Equation (2.26), maximisation of this value should lead to the 'best' fit of the model.

The field of optimisation is established and varied, a full survey of all possible algorithms and techniques exceeds the scope of this thesis and probably the attention of the reader! It is useful, though, to discuss broadly some of the different overarching themes of this body of work. The first consideration in optimisation is if the approach aims to solve the global or local optimisation problem. The local problem aims to find the optimal solution (minimum cost) in the region close to some specified starting point — in many cases this will not lead to the global minimum of the cost function (the global minimum being the point that minimises the cost across the entire domain of the function). In this case there remains a solution that is better — i.e. will give a lower cost. In contrast to this, global methods aim to find the single minimum value of the cost function. There are also two main approaches to the optimisation problem, the first more suited to the local problem, and the second to the global problem.

This first approach is gradient based optimisation. Broadly speaking, in a gradient based optimisation, the minimum value is found by moving in the direction of negative gradient. This is most simply understood in an example as, imagine one was out walking, possibly around Sheffield's seven hills, searching for the lowest point. To find this point it is intuitive to try and always walk downhill and, if there

is no direction in which you can head downhill, you must be at a minimum. In this scenario it is easy to see how this would not necessarily find the absolute lowest point since the only information available is the current location and the gradient at that location. More sophisticated techniques modify the direction in which steps are taken to achieve two things, either faster convergence or to avoid becoming 'stuck' in local minima. For example, this could be the addition of momentum terms or making conjugate steps [104, 105].

The most common alternative to this is to use a population-based optimisation method. Returning to the analogy of finding the lowest point in Sheffield; one could survey the height in a number of locations and the combine all of these spot heights to try and choose new points closer to the minimum. For example, it is sensible to believe that if a point has a small value then it is likely that the area around it will also have small values — i.e. the function is continuous, which in the case of the terrain of Sheffield is generally true! Within this class of optimisation algorithms exist those based on genetic operations; the genetic algorithm or differential evolution [106] and their derivatives [107, 108]; the particle swarm family [109]; many nature inspired algorithms [110–113].

Gradient-Based Optimisation

By far the most popular choice for the optimisation of GP hyperparameters is a conjugate gradient method, probably owing to the recommendation of this approach in [74]. These methods are a variant on simple gradient descent, where the optimisation step direction is chosen to be a conjugate direction; this should lead to faster convergence to the local optimum. Since the method is gradient-based, it is susceptible to converging to local minima and exhibits poor exploration properties, but will potentially lead to faster convergence times and good exploitation behaviour. That is, it will quickly find the local minimum in few steps.

Conjugate gradient methods comprise of the repeated application of three substeps. The first being the determination of the current function value and gradient, in the case of GP with respect to the kernel hyperparameters. This is the negative log marginal likelihood and all its partial derivatives. The second stage is to choose the conjugate direction; there are a number of formulations of the conjugate direction, the method proposed by Dai and Yuan [105] is used for experiments here although others include the traditional [104]. The approach of Dai and Yuan [105] is shown in

Equation (2.28).

$$\beta = \frac{|\nabla \mathbf{f}|^2}{\nabla \mathbf{f}^\mathsf{T} \mathbf{y}} \tag{2.28}$$

where $\mathbf{y}$ is the function value and $\nabla \mathbf{f}$ is the vector of partial derivatives. Thirdly a line search is conducted along this conjugate direction to find the one dimensional minimum, for example Brent's line search [114]. No more iterations of the search are performed when either: the gradient of the cost function with respect to every dimension ($\nabla \mathbf{f}$) is approximately zero; the change in the cost function value ($\mathbf{f}$) is close to zero; or a predefined maximum number of iterations is reached. It should also be noted that within the family of gradient decent methods if the function value or gradient are undefined at any point (e.g. an asymptote in the cost function) the method cannot be applied. Fortunately, this is not the case when optimising the Gaussian process, but to improve numerical stability it is worth calculating both the marginal likelihood and its derivatives in the log space — that is to operate with the log values of the hyperparameters. The calculation of the log derivative can be achieved via the chain rule which is shown below.

$$\frac{\partial K}{\partial \log \theta_i} = \frac{\partial K}{\partial \theta_i} \frac{\partial \theta_i}{\partial \log \theta_i} = \frac{\partial K}{\partial \theta_i} \theta_i \tag{2.29}$$

for i indexing each of the hyperparameters of the kernel. This is important since the calculation of the gradient of the negative log marginal likelihood requires the derivative of the covariance matrix with respect to the hyperparameters.

Population-Based Optimisation

An alternative to using a gradient-based method such as conjugate gradient descent, is to employ a population-based search, where the costs at a number of randomly initialised starting locations are combined to converge towards an optimal solution. These methods include: evolutionary algorithms including the genetic algorithm, more sophisticated techniques such as differential evolution (DE) [106], and adaptive methods such as self-adaptive differential evolution (SADE) [107] or JADE [108]. There also exist a variety of methods with inspiration from physics [115], or the smörgåsbord of animal-inspired algorithms [110–113].

An example of this form of approach to the optimisation problem is the Quantum Behaved Particle Swarm (QPSO) [116, 117]. The QPSO is a variation on the traditional particle swarm optimisation where each particle is modelled as having quantum behaviour. The dynamics of the particle are governed by the wave function of the imagined quantum state of each particle. One of the key advantages of the QPSO algorithm is that it is proven to be globally convergent [117]. There are a number of additional variants of this specific algorithm which are discussed further in [118–120].

The details of the QPSO algorithm are laid out in [116] and not reproduced here. The only modification made is to have the expansion-contraction coefficient α be a function of the number of generations to create an annealing like behaviour. The choice of optimisation scheme is famously impossible to objectively justify — there is *no free lunch* [121]. This theorem states that considering all optimisation problems, there is no one algorithm that will always perform better. The author has, however, found that the QPSO algorithm has performed well in the optimisation of GP hyperparamters and other engineering optimisation problems — for example see the results presented in [122].

2.5.2 Results

The effect of the optimisation scheme will be shown on three different datasets which are described here. The first synthetic dataset is taken as a random draw from a GP model which is a single realisation from a GP prior with a known set of hyperparameters. A test function is drawn from a one dimensional GP prior using the Matérn 3/2 kernel[3] with inputs generated in the range $x = [-10, 10]$, a signal variance of 20, length scale of 15, and a noise variance of 0.01. In order to generate a higher dimensional nonlinear dataset a similar procedure was followed, first forty draws are made from the GP specified as above. These forty draws are then used as inputs to a forty-dimensional input GP, with hyperparameters defined as $\sigma_f^2 = 20$, $\lambda_i \sim \mathcal{U}(0, 100)$, $i = 1, \ldots, 40$, and $\sigma_n^2 = 0.001$. The notation $\mathcal{U}$ is used to denote a uniform distribution bounded by the two arguments. A single realisation of this high dimensional GP is used as the outputs for the dataset.

Finally, the performance of each optimisation scheme is tested on an SHM-focussed

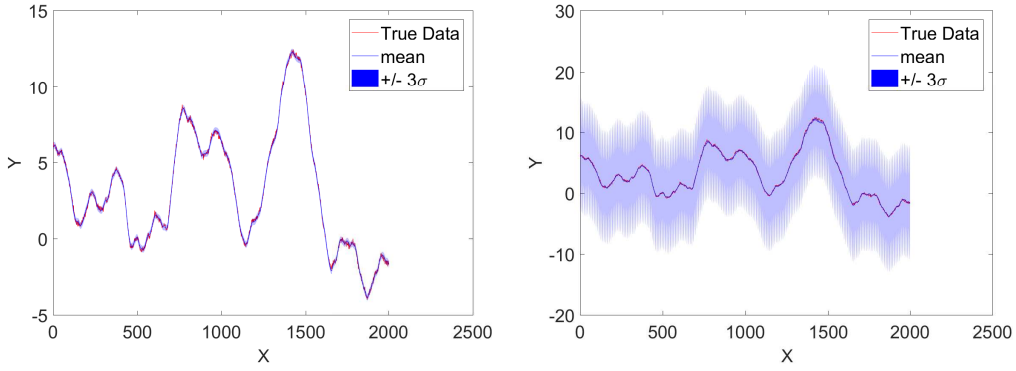
[3]See Section 2.3 for a thorough introduction.

dataset. A Tucano TMk1 Trainer aircraft was instrumented with the aim of applying techniques for prediction of strain histories for assessing fatigue damage. The dataset has been studied in more detail in [77, 123, 124] where application of ANNs and GPs have already shown very promising results. The full dataset contains one hundred flights across two different aircraft, with 14 input variables. For training, a restricted dataset is chosen, where the first sortie from one aircraft is chosen as training data and a test sortie is randomly-chosen — ensuring the test sortie is taken from the same aircraft. This may well not be the optimal training set for this process as it is likely it does not cover the input space sufficiently, resulting in errors above that which would be expected, the use of a sparse GP implementation such as such as those described in [125–128] with a larger training set would likely improve predictions without large increases in computational expense.

The first test is to consider the optimisation of a Gaussian Process with a low dimensional (one dimensional in fact) input space. Therefore, the optimisation task should be simplified since the hyperparameter space which requires searching is lower dimensional. The search space will be three dimensional since there are three hyperparameters — the signal variance σ_f^2, the lengthscale ℓ, and the noise variance σ_n^2 for the chosen Matérn 3/2 kernel. Since the function itself is drawn from the GP it is known that there isn't a model form problem — the kernel is known *a priori* to be correct.

It is possible to investigate the role of the starting point of the gradient descent in the optimisation since the 'true' hyperparameters are also known. Initially the model is trained via gradient descent from values known to be close to the 'true' hyperparameters, the result of this is shown in Figure 2.11. Clearly the GP is able to find a set of hyperparameters that lead to the function being modelled very well and the NMSE for this model is low at 0.1054 as shown in Figure 2.11(a). However, if the optimisation is started from values which are known to be far from the true hyperparameters, the result is very different. Consider the result shown in Figure 2.11(b), where an undesirably short lengthscale has been found as the 'optimum', leading to poor modelling of the true functional uncertainty and behaviour where the model fails to generalise. Even though the test points are very close to the training points, the variance of the prediction is seen to increase rapidly. This is again symptomatic of a very short length scale.

More realistically, however, it is very hard to tell *a priori* where a "good" or "bad" start point for the optimisation may be. It is sometimes possible to make prior



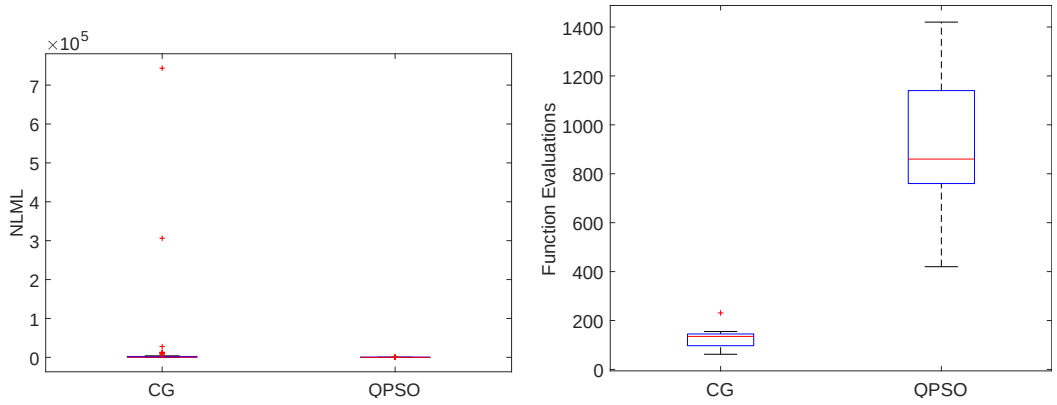
(a) (b)

Figure 2.11: Plots of the prediction on an independent test set for Gaussian Processes, modelling the low dimensional test function, trained via conjugate gradient descent from start points close to the known hyperparameters (a) and far from the known hyperparameters (b).

assumptions regarding the measurement noise and set a meaningful starting value for σ_n^2 or the total signal variance σ_f^2. However, usually making meaningful suggestions about the lengthscale is difficult and becomes more so as the dimensionality of the input space increases. It is possible to perform multiple gradient descents from randomised start locations to attempt to find multiple minima where hopefully one is the global minimum. It is the author's understanding that this is the accepted best practice in the community, this comes from discussions with other researchers in the GP community and cannot be found in the literature! This random restart methodology is compared to the use of the QPSO as a global optimiser. Each of the methods is run for fifty iterations so fifty gradient descents are performed from fifty randomised locations and the QPSO is also run fifty times with randomised initial population locations. These two approaches will be compared on three criteria, the first being the value of the negative log marginal likelihood that they optimise to, the second is the number of function evaluations before the optimisation converged, and the third is the normalised mean squared error of the prediction on an independent test set.

In Figure 2.12 boxplots[4] are shown comparing the distribution over the final best value of the optimisation of the negative log marginal likelihood, in the low dimensional

[4]The box-plot allows a summary of a distribution of values to be shown. The horizontal line indicates the median value with the box showing the interquartile range. The whiskers indicate the 2.7σ range which can be thought of as all data which is not considered outlying if the data were Gaussian distributed. "Outlying" data points — ones beyond the range of the whiskers — are shown with the red crosses.



(a) Negative Log Marginal Likelihood (b) Function Evaluations

Figure 2.12: Boxplots of (a) and the number of function evaluations before convergence is also shown (b) for the conjugate gradient and QPSO optimisations of fifty GPs.

test, for fifty runs of both the conjugate gradient descent and the QPSO. It can be seen that the conjugate gradient optimisation will converge faster in general (with notable outliers) in a single run than the QPSO. However, the variance in the optimal value of the marginal likelihood is far greater than for the QPSO method — the main reason for this is the appearance of several severe outliers. Even ignoring these outliers, optimisation using the QPSO consistently leads to slightly better (more minimal) values of the negative log marginal likelihood. This suggests that the models learnt via the QPSO can better (and more consistently) explain the observed training outputs given the training inputs and the hyperparameters found.

Figure 2.13 allows comparison of the predictive performance of the models summarised by the NMSE of predictions on an independent test set. The results seen here would indicate that the reduced optimisation performance of the conjugate gradient scheme seen in Figure 2.12(a) is mirrored in the prediction capability of the model. The variance in the NMSE of the optimised models reflects the variance seen in the marginal likelihood during training. It is expected that this difference will continue and could be exaggerated as the optimisation space in high dimensions may be more prone to local minima.

The same results are shown for a synthetic model that is high dimensional. Similar results are noted in Figure 2.14 as Figure 2.12 where the QPSO consistently optimises to better values of the marginal likelihood at the expense of an increased number of function evaluations.

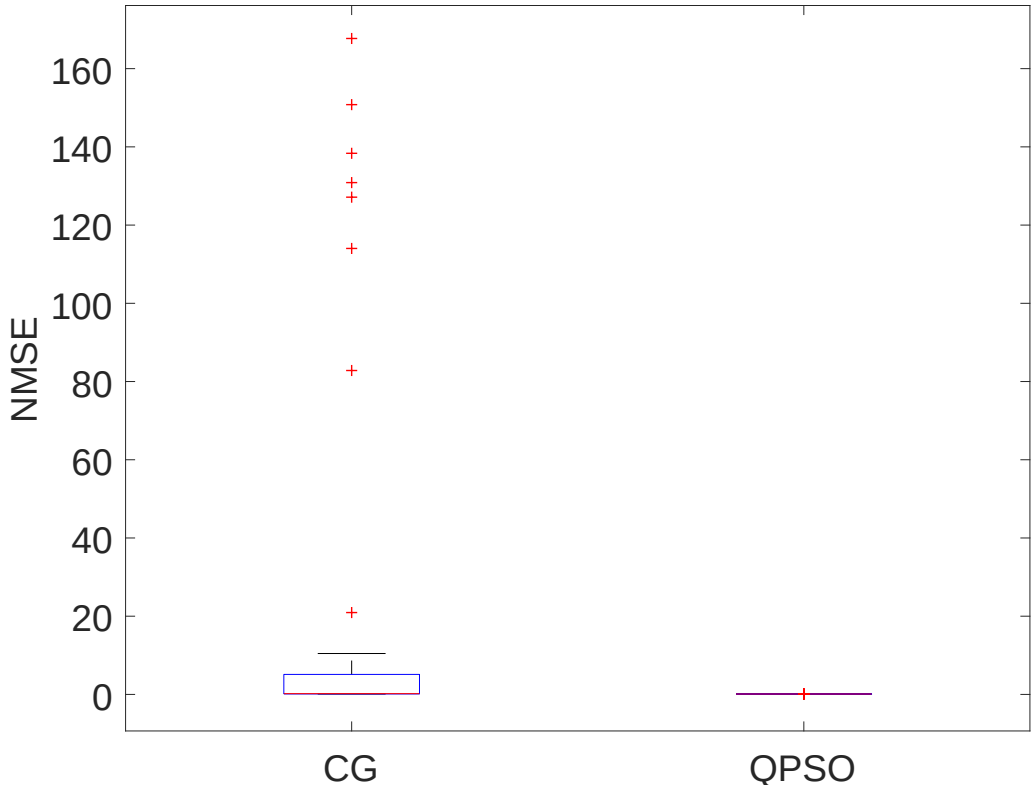
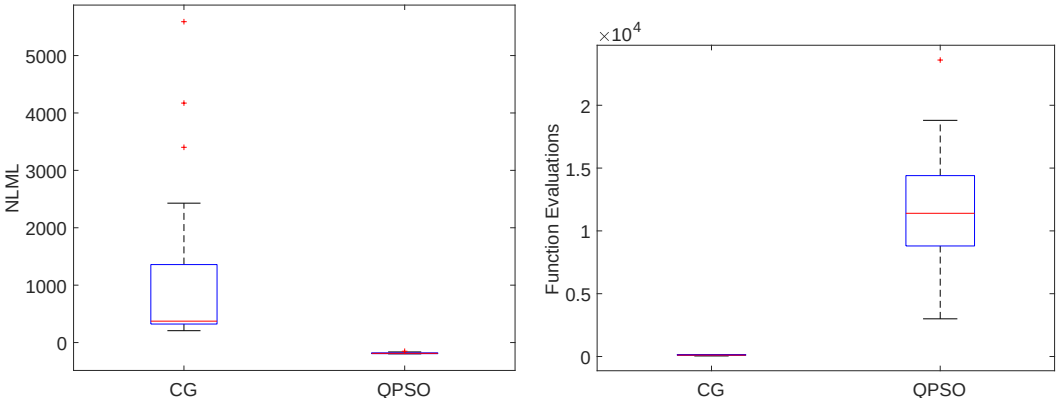


Figure 2.13: Comparison of normalised mean squared errors for models trained with conjugate gradient descent compared to QPSO for the low dimensional test function.



(a) Negative Log Marginal Likelihood (b) Function Evaluations

Figure 2.14: Boxplots are shown comparing the distribution over the final best value of the optimisation, using the high dimensional input space, of the negative log marginal likelihood for fifty runs of both the conjugate gradient descent and the QPSO (a) and the number of function evaluations before convergence is also shown (b).

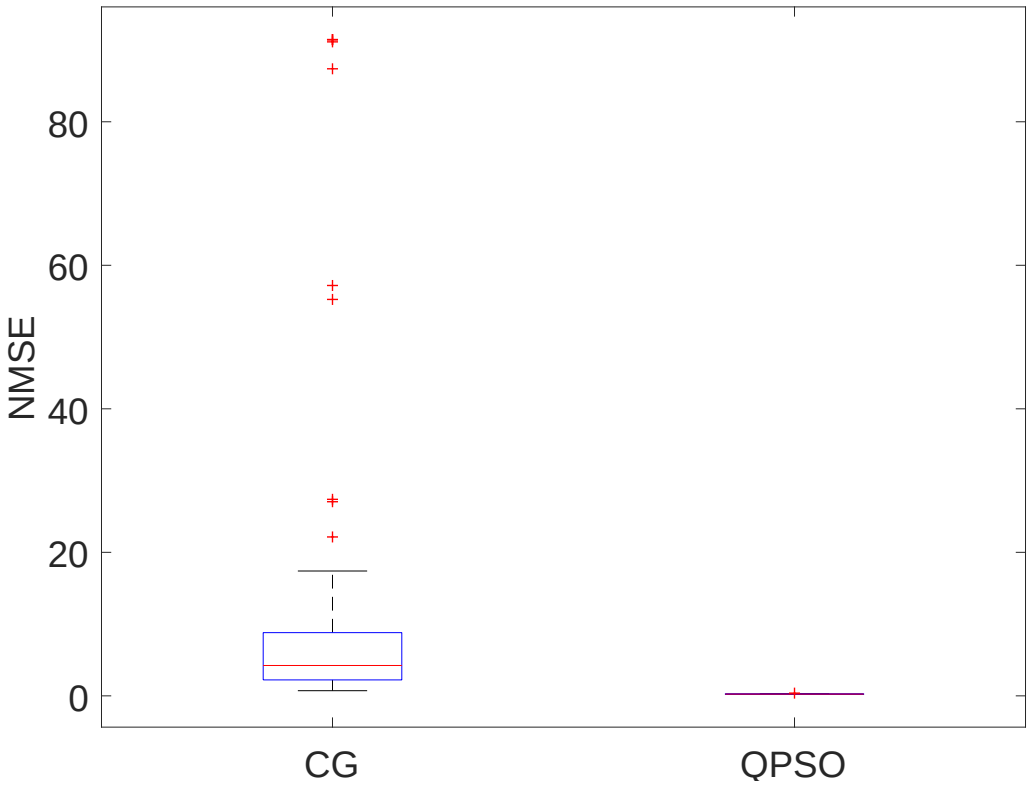
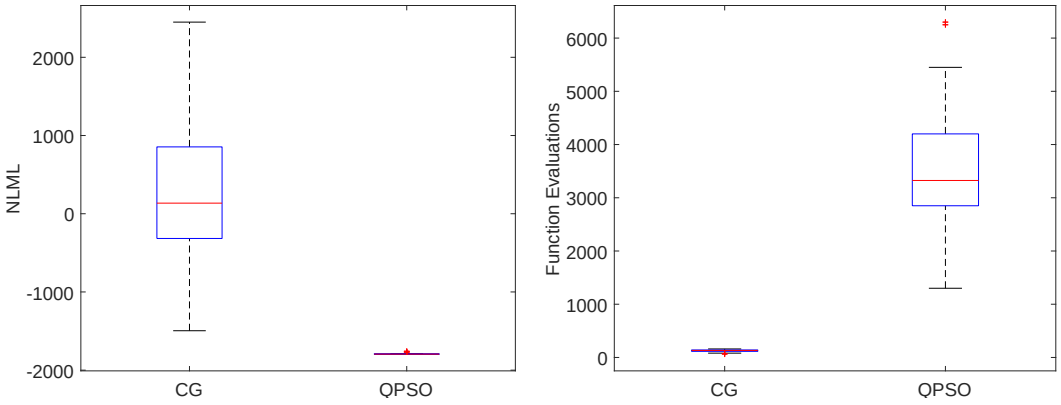


Figure 2.15: Comparison of normalised mean squared errors for models trained with conjugate gradient descent compared to QPSO for the high dimensional test function.

Considering the predictive performance of the models summarised by the NMSE in Figure 2.15, similar results are observed. The increased variance in the optimal cost found by the optimisation algorithm is translated into increased variance in the error of the predictions. The QPSO is again able to find consistently better marginal likelihoods which result in lower normalised mean squared error scores. This evidence would suggest that the QPSO may be a more robust training mechanism for Gaussian Process models. It is useful to test this hypothesis on data collected from an SHM setting to ensure that it isn't an artefact of using synthetic test functions.

To do this, Gaussian Process models are trained for the prediction of the port side inner wing bending strain on a Tucano MkII. Trainer aircraft using a dataset previously presented in [77, 123, 124]. The inputs used are:

1. Total fuel mass
2. Indicated airspeed
3. Barostatic altitude



(a) Negative Log Marginal Likelihood (b) Function Evaluations

Figure 2.16: Boxplots are shown comparing the distribution over the final best value of the optimisation, when modelling the port inner wing bending strain of the Tucano aircraft, of the negative log marginal likelihood for fifty runs of both the conjugate gradient descent and the QPSO (a) and the number of function evaluations before convergence is also shown (b).

4. Normal acceleration at the centre of gravity

5. Normal acceleration at the tail

6. Normal acceleration at the port wing

7. Elevation

Again fifty models were trained with each of the conjugate gradient descent and the QPSO, optimising the hyperparameters through minimisation of the negative log marginal likelihood.

Yet again, similar behaviour is observed in the performance of the optimisation algorithms. The conjugate gradient descent requires fewer iterations to converge to a minimum of the cost function but the QPSO finds consistently better optimum values. It would be expected, therefore, that similar outcomes would be observed in the predictive performance of the models.

However, on inspection of the results shown in Figure 2.17 an interesting behaviour is observed. The first point to be noted is that neither model is performing very well on the independent test set. Again it is seen that the overall performance of the QPSO optimiser is better than the conjugate gradient based scheme.

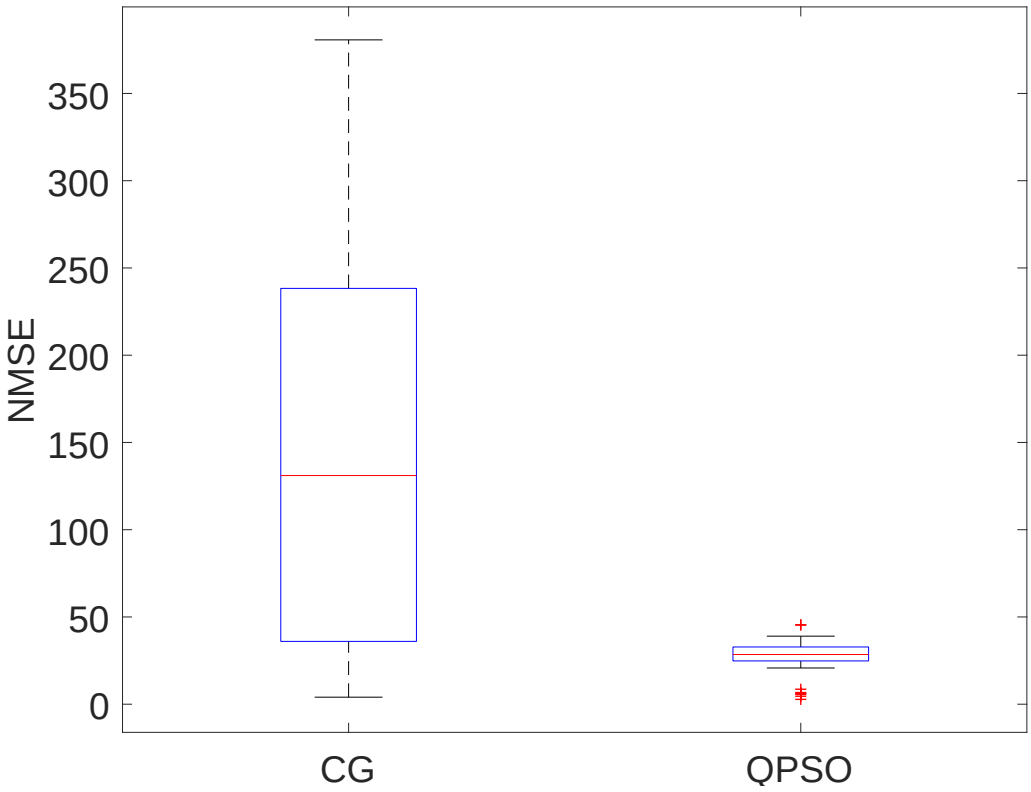
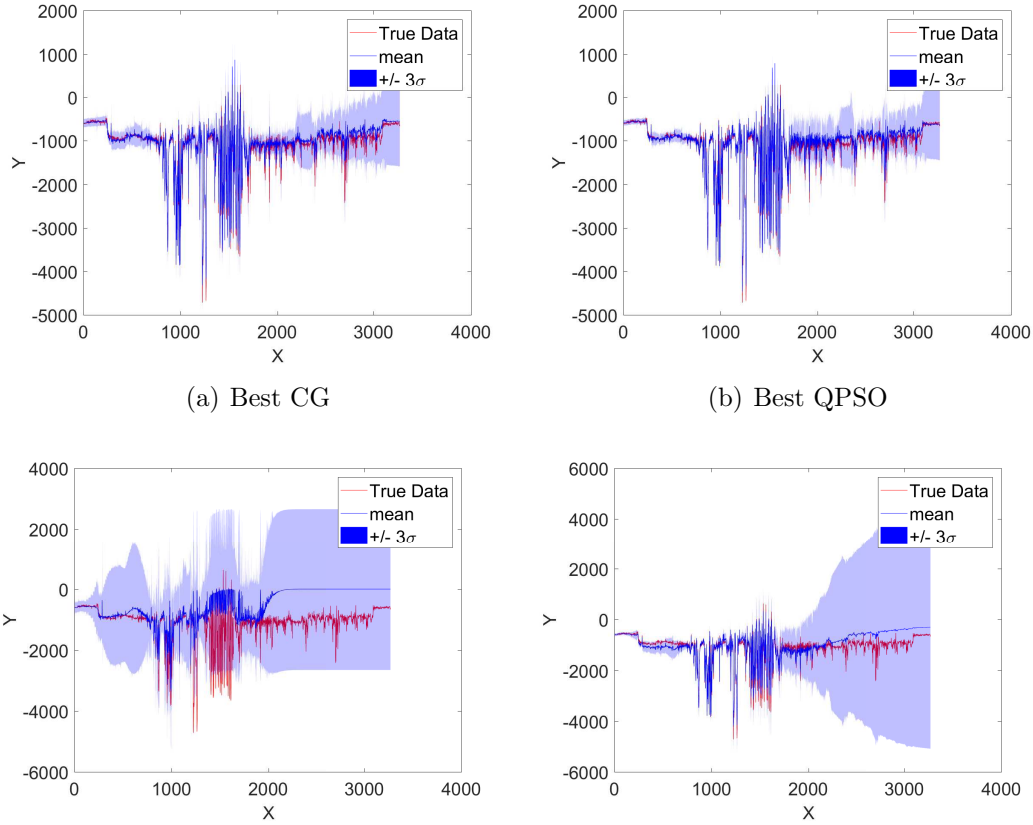


Figure 2.17: Boxplot comparison of normalised mean squared errors for prediction of the port inner wing bending strain on the Tucano aircraft data.

Figure 2.18 shows both the best prediction from each of the optimisations in terms of the NMSE score on an independent test set and also the plot of a 'mean' prediction. This is, for each, the model with a score closest to the mean score. What is observed in all cases is growing variance toward the later part of the time series being predicted. This indicates reduced confidence of the model in making the prediction. This area, from roughly point 2000 onwards, also accounts for most of the error in the signals. Comparing these models, the quality of the models fitted by the QPSO would be said to be better heuristically; when the model is confident the variance is reduced and the 3σ confidence covers the measured function values on this independent test set.

What is clear is that the Gaussian Process has not been able to accurately model all of the behaviour required to predict the value of interest. This highlights the importance of a number of things, the first is that it can't be assumed that since a model has consistently optimised to a minimal value of the marginal likelihood that the model will perform well in prediction. There are three important considerations



(a) Best CG

(b) Best QPSO

(c) Mean CG

(d) Mean QPSO

Figure 2.18: The best and 'mean' model are shown for each of the conjugate gradient and QPSO methods of optimisation including the mean prediction and 3σ intervals. For context, the NMSE scores of these predictions are (a) 4.03, (b) 2.77, (c) 139.16, (d) 27.34.

that shed light on the poor performance of the Gaussian Process in this situation:

1. Is there sufficient training data to cover the input space?
2. Is that training data representative of the true underlying function?
3. Is the model selection correct?

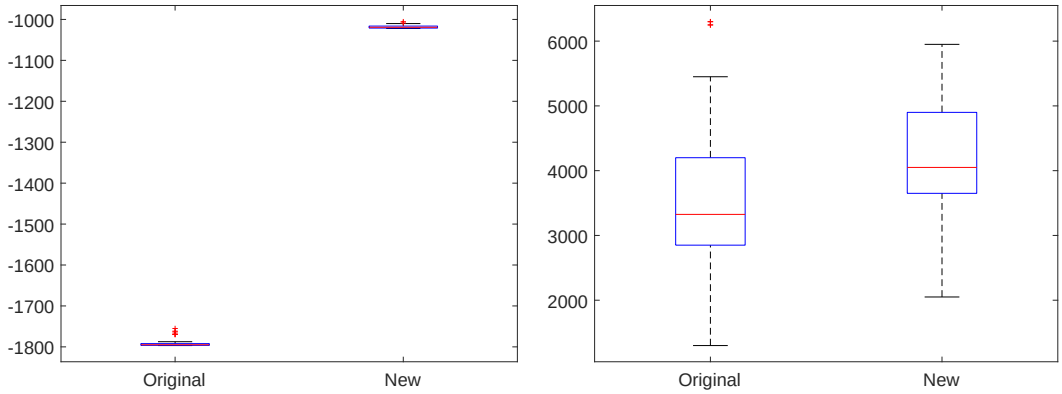
Stepping through each of these it is possible to diagnose the problems with this model and refine it — but this is only possible in the presence of an independent validation set. It can't be stressed enough the importance of checking the model fit in this way before implementing this type of model in an SHM system. The behaviour seen in the "mean" models for each optimiser would suggest that the training set does not

sufficiently cover the input space. This is not limited to identifying the bounds of the space but also ensuring that there are sufficient points that no prediction point is too far from any individual training point. The growth in variance returning close to the value of σ_f^2 the signal variance hyperparameter indicates that the model is unable to make a valid prediction — this come from the low covariance between the test inputs and any training input.

The second consideration, if the training data is representative, can be easily checked by observing the residuals when the variance in the prediction is low. If the residual of the model is high and the variance is low then the model is confidently predicting wrongly. This should be a major concern since it indicates that the functional form learnt from the training set is different from the one that governs the test set. In this case, the model should not be used until further data becomes available and a new test set can be validated. It is of course possible once more data have been collected to incorporate the initial test set into the training set, this may be advised since the model is known to converge to the true function in the limit of infinite data.

Finally, the choice of model can cause poor performance behaviour. For example if the covariance function imposes unrealistic assumptions of smoothness or periodicity. This, however, is far harder to diagnose and requires further investigation into the correlations in the model. This investigation was conducted in this case and it was found that there were certain variables with strong linear trends which had only minor nonlinear variation in them. It was possible, therefore, to redefine the covariance function to be a sum kernel with a linear component placed over input variables four to six — the effect of this is discussed further in Section 2.3. It is useful now to compare the model trained previously using the QPSO with this new kernel form to understand the effect of model selection on the quality of the model.

In Figure 2.19(a) the same consistency in the optimisation is seen for both the old kernel form (Matérn 3/2 only) and for the new kernel form with the added linear kernel. The optimal values found for negative log marginal likelihood are higher for the new kernel — the reason for this is that the quality of fit of the training data is does not improve greatly with the addition of the linear kernel but the model complexity increases significantly. This increase in model complexity is penalised by the determinant term in the negative log marginal likelihood as per the Bayesian Occam's Razor. This indicates that, when concerned with predictive model performance, using solely the marginal likelihood for automated kernel selection may lead to models which fail to generalise. Figure 2.19(b) shows that while the addition



(a) Negative Log Marginal Likelihood (b) Function Evaluations

Figure 2.19: Boxplots showing the difference in distribution over the marginal likelihood (a) and the number of function evaluations (b) when comparing models with two different kernel forms for the Tucano data, with both trained via QPSO.

of the linear kernel increases, in general, the number of function evaluations for the optimiser to converge, this increase is marginal compared to the difference between the gradient descent method and the QPSO. This increase in the number of function evaluations is due to the increase in the number of hyperparameters being optimised from the addition of the extra kernel.

The interesting result is found when considering the predictive performance of the model shown in Figure 2.20. The results seen here for the new kernel form are far closer in line with those seen in the synthetic function experiments. The consistency in the optimal marginal likelihood found is reflected in the predictive performance of the model. The mean NMSE of the predictive models is 1.68 for the new kernel form compared to 27.34 for the previous model, and the best model improves from 2.77 to 1.28. This clearly indicates that by making an informed choice of kernel for the Gaussian Process it is possible to build models that generalise from small training datasets, in this case a single sortie. Here is has been possible to exploit the fact that some of the nonlinear trends also exhibit strong linear components which can be captured through the use of a sum kernel. This has led to a far more robust model for the Tucano data.

This section has attempted to show the importance of the choice of optimisation scheme in learning Gaussian Process hyperparameters. The popular conjugate gradient descent method has been shown to be difficult to use without multiple restarts due to its susceptibility to becoming 'stuck' in local minima. This is significant

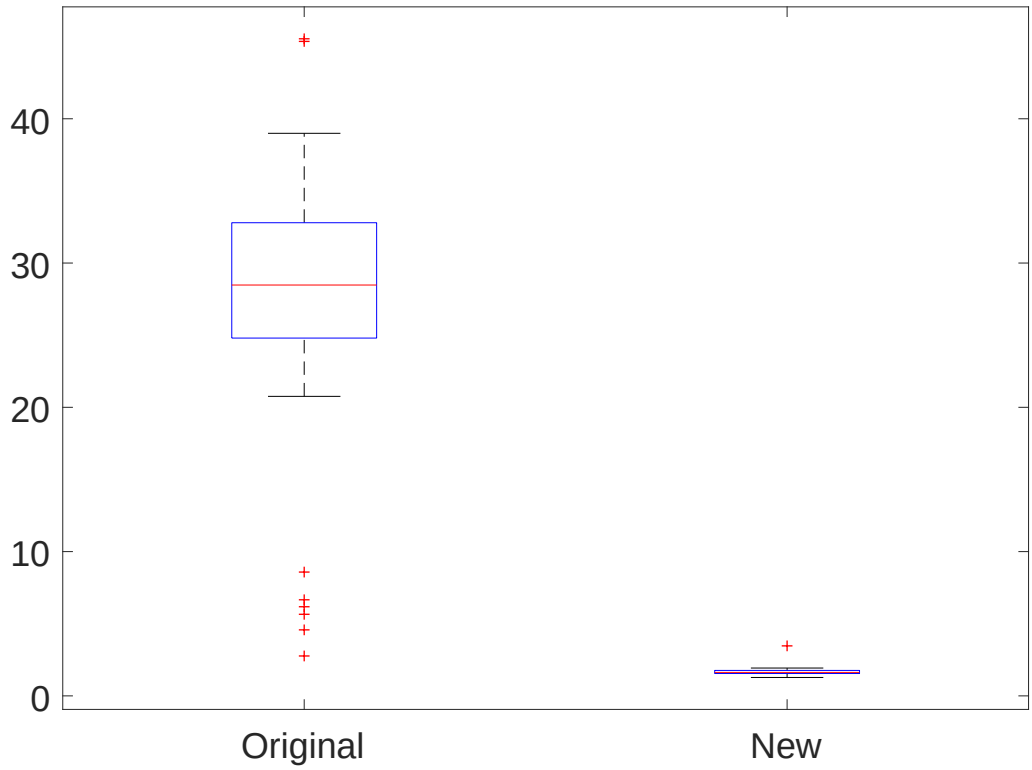


Figure 2.20: Boxplot comparison of normalised mean squared errors for prediction of the port inner wing bending strain on the Tucano aircraft data given the two different choices of covariance funciton.

as the importance of multiple starts in the gradient-based optimisation is rarely (if ever) stated and plays a significant role in the quality of the fit of the model. The QPSO has been used to demonstrate how a 'global' optimisation scheme can have advantages in consistency and also in exploiting better minima. This has been shown to improve performance on two synthetic test functions. It was also demonstrated on data from the Tucano aircraft that even if an optimiser finds consistently minimal marginal likelihood values this can still lead to poor models if the choice of covariance function is poor. The addition of a linear kernel to the Matérn 3/2 kernel was shown to significantly increase predictive performance on an independent test set in terms of the NMSE. However, this did not lead to better values of the marginal likelihood, indicating the difficulty in model selection.

2.6 Discussion

When presenting the Gaussian Process model, it is often not clear the effect of the user choices on the performance of the model. The usual examples for explaining the model are shown in low dimensions using functions with a known form (commonly the data can be drawn as a realisation from the Gaussian Process being trained). In this case it is easy to overlook the points that have been raised in this chapter. The effect of the choice of the optimisation function has been shown which has demonstrated the danger of relying on a single run of a gradient-based optimiser for learning GP hyperparameters, since the function being minimised has a number of local minima. It has also been shown that the role of kernel selection is key in defining a model that is able to generalise effectively. This has highlighted the need for user input in the training process to decide on the most appropriate optimisation scheme and to determine model form.

The approach of gradient descent, namely the conjugate gradient method, has been shown to be efficient at finding local minima. However, this has been shown to lead to high variance in the results of the optimisation of the marginal likelihood. This can lead to alternate solutions to the Gaussian Process model, the Type-I and Type-II solutions [87]. For this reason, it would be recommended that multiple random restarts are used with the conjugate gradient optimisation scheme — an important caveat when discussing the computational advantage of gradient descent methods over population based methods that is often overlooked.

It has been shown that, given this, it is competitive to use a 'global' optimisation scheme. Here, a population based method has been demonstrated to be effective. The use of the QPSO specifically has been shown to allow optimisation efficiently as the dimensionality of the input increases. The consideration of the choice of cost function for the optimisation has also been introduced.

It has been argued that, where possible, it is good practice to make use of the Bayesian Occam's Razor. This behaviour is achieved via the use of the marginal likelihood as the cost function (strictly the negative log marginal likelihood). It has also been suggested that, since the marginal likelihood is implicitly conditioning the hyperparameters on the observed training data by taking a maximum likelihood estimate of the posterior of the hyperparameters $p(\theta \mid \mathcal{D})$ in the optimisation, if the training dataset $\mathcal{D}$ is not representative of the function being modelled this cost

function can lead to solutions that don't generalise.

If a user believes that the training data may not be representative, the first course of action should be to increase the size of the training set. If this option is not available, then the use of the posterior probability of an independent validation set given the Gaussian Process predictions may help. As in traditional parametric models, the leave-one-out approach should be seen as the "gold standard" for cross validation. However, the nonparametric nature of the model means that this is usually unnecessary. One case where a user may be cocerned about this kind of behaviour is when fitting a dynamic model such as the GP-NARX since here the input-output relationship is much harder to visualise. It has also been discussed that the use of a Bayesian estimation for the hyperparameters may also help to protect against some of these issues linked to the maximum likelihood estimate of the hyperparameters. This Bayesian estimation can be achieved via a number of methods, although since the computation of the posterior is intractable, all these will require computationally expensive numerical approximations of the posterior. For this reason, especially so on large engineering datasets, it is usually not feasible to compute the full posteriors unless there is good reason to believe that other estimations of the hyperparameters are insufficient.

In addition to the consideration of learning the hyperparameters, the difficulty of kernel selection for use of Gaussian Process models in SHM has been discussed. Alongside this, the choice of covariance problem the search space for the model selection grows indefinitely with combinations of kernels. It was shown that the choice of kernel when modelling the Tucano aircraft led to significantly reduced predictive performance of the Gaussian Process. However, the process of determining a better model form which more readily generalised the training set required intervention from the modeller. This highlights the fact that the use of machine learning within SHM still requires intervention from an experienced user; a situation which could impede adoption in industry.

This chapter has introduced and discussed a number of challenges to using Gaussian Process models which can be overlooked when they are presented in literature. The modelling choices required and their effect has not previously been presented with respect to the usefulness of these models for SHM and the work shown here should aid in robust practical implementation of these models. It has also highlighted the need for independent validation sets and expert intervention in the modelling process to ensure that the models are performing as expected.

Chapter 3

Handling Dynamic Data with Gaussian Process NARX Models

Highlights:

- *The Gaussian Process NARX model is presented, highlighting some of the challenges in its implementation*
- *A novel comparison of uncertainty propagation techniques for GP-NARX is made*
- *The problem of lag selection in GP-NARX is discussed and contrasted with parametric NARX models*

It has been discussed that the GP is a powerful approach to regression tasks, however, it suffers from two major drawbacks. Firstly, it is — like most if not all data-driven models — an interpolating model, but many engineering tasks require extrapolation in time. Secondly, the model is a static mapping, that is, the GP treats every observation as independent of order. It does not have the facility for managing dynamic effects.

The first of these shortcomings is easier to overcome than the second. The uaual way to solve this problem is to transform the extrapolation problem into an interpolation problem by choice of the regressors (input variables). For example, say that the problem of interest is to predict the strain at some point on a structure as time

progresses, the choice of time as a regressor is a bad one since it will require extrapolation. However, if the strain at one point on the structure is nonlinearly correlated in a static manner to another measurement e.g. the strain at a different point on the structure, this is a far better choice of regressor.

Handling dynamics explicitly within the GP is a more challenging prospect. Various approaches have been considered in literature. These include modifications to the prior distribution [129], latent variable approaches [130, 131], or the Gaussian Process state-space model GP-state-space model (SSM) [132–134]. A good review of some approaches can be found in [135], including discussion on the GP-NARX.

Within engineering, the GP-NARX model has already been applied to a number of problems. Firstly, preliminary results for predicting wave loading have been shown in [96]. It has also been shown to be effective in a grey-box setting for identification of a nonlinear system identification benchmark, [122]. It has also been shown that the probabilistic predictions from the GP-NARX can be propagated to return a distribution over a nonlinear system FRF [136, 137]. There is also significant interest in using the GP-NARX model within a model predictive control setting, for instance see [138].

3.1 The GP-NARX Model

This model is part of the more general family of Nonlinear Auto-Regressive model with eXogenous inputs (NARX) [139], where the output of the model is some function of previous or lagged versions of the output and lagged versions of the inputs. Defining the exogenous inputs as a vector $\mathbf{u}_t$ at each time step t and the output y_t as the output at time t; leads to the input vector to the GP to be given by, $\mathbf{x} = \left[\mathbf{u}_{t-l_u}, \ldots, \mathbf{u}_t, y_{t-l_y}, \ldots, y_{t-1}\right]$ for l_u lags in the input and l_y lags in the output. This leads to a model in the form,

$$y_t = f(\mathbf{x}) + \varepsilon = f\left(\left[\mathbf{u}_{t-l_u}, \ldots, \mathbf{u}_t, y_{t-l_y}, \ldots, y_{t-1}\right]\right) + \varepsilon \qquad \varepsilon \sim \mathcal{N}\left(0\,,\ \sigma_n^2\right) \tag{3.1}$$

Given the discussions previously, it may be advantageous to attempt to place a GP prior over this function rather than a parametric representation of the auto-regressive

function for the same reasons as when performing standard regression tasks. This amounts to learning a GP between the regressors $\mathbf{x}$ and the output y_t which can be assembled into an input matrix X and output vector $\mathbf{y}$ by considering the whole time series with the restriction that the first point in $\mathbf{y}$ will be $y_{\max(l_u,l_y+1)}$.

In many ways the GP-NARX model is identical to the static GP in its implementation. However, there are three considerations which should be made, two concerning prediction using the GP-NARX model and the other concerning training. These are: the difference between one step ahead and full model predictions, the propagation of uncertainty through a GP-NARX model, and the selection of lag terms in the model.

In terms of prediction there are two possible predictions that the model can make,

1. One Step Ahead (OSA) prediction, as it is termed by [139], refers to the prediction of the output given previous measured outputs, i.e. the model only every predicts one step into the future. This is also referred to as the "prediction" method within control systems which can lead to some confusion, for this reason the term OSA is used here.

2. Model Predicted Output (MPO) is the alternative to this in which previous model predictions are fed back into the inputs of the model such that the model will attempt to predict forward in time indefinitely (in theory) without additional observations of the "true" output. Again, within control systems, this can also be referred to as the "simulation" test of a model.

In general, obtaining a good MPO prediction of a model is a more challenging task and, as such, a more rigorous test of the performance. It is useful to consider both the OSA and MPO behaviour of any model when assessing its quality, however, the desired use of the model will guide which of these is a more important metric. It is likely that, in SHM, the MPO will be the more desirable use of this type of model.

The astute reader will notice that both these approaches violate one of the fundamental assumptions of the Gaussian Process; that being that there is no noise on the inputs to the model. The effects and handling of this will be discussed in Section 3.2.

The training of the GP-NARX model also poses some additional challenges compared to the standard GP formulation. This is for two reasons firstly, by moving to the NARX formulation a number of additional hyperparameters have been introduced

to the model. These relate to the selection of which lagged inputs and outputs to include in the model. There are two problems with this, the first philosophical and the second practical.

1. The selection of a finite number of lags to represent a process is relying on truncating the true mechanism of the process based upon the assumption that behaviour that is very far in the past no longer can affect the current behaviour. When choosing to take all lags up to some horizon n this amounts to the assumption that the process being modelled is an n^{th} order Markov process. This is normally a strong assumption that is hard to validate.

2. If it is assumed that the process can be truncated and represented accurately by some finite set of lagged variables, the problem becomes a practical one. The selection of the lags in the input and the output is a combinatorial optimisation problem concerned with the selection of the best subset of lags to capture the process — this is combinatorial integer programming problem which is NP-hard [140].

Since it is necessary to produce models which can be used, it will be assumed that there exists some finite set of lags which will adequately well approximate the process being modelled and that the first issue seen here will not negatively impact the implementation of a GP-NARX model in an excessive way. The continuing prevalence of models with a NARX structure and their longevity as a useful tool would suggest that this is a valid assumption to make. The issue of combinatorial lag selection in NARX models is worth considering, however, and is discussed further in Section 3.4.

3.2 Handling Uncertainty in GP-NARX Predictions

When making use of measured data from experiments or in service testing it is impossible to obtain measurements where the noise on the inputs has been completely eliminated. Through controlling experiments well this can normally be minimised by the use of calibrated, precise instrumentation and making the assumption of no noise on the inputs in the GP (and other models). This assumption is actually encoded in

the construction of the model,

$$y = f(\mathbf{x}) + \sigma_n^2$$

it can be seen that the noise in the model only enters additively after the functional transformation of $\mathbf{x}$. It should be noted that this noise is not strictly the measurement noise but rather the noise between predictions of the process and the true values. Unless otherwise stated this is a fundamental feature of the application of all GPs in the literature.

When training a NARX model the question must be raised as to whether it is satisfactory to consider the observations of the output (in the training set) to also have sufficiently low noise to be able to assume that it will not cause issues when used as an input to the GP in an Auto-Regressive (AR) manner — i.e. the measurement noise in the model is also approximately zero and the function can be modelled almost exactly. If the user decides that the data is acceptable for fitting an AR model, then it is acceptable to treat the OSA predictions of the GP-NARX in the same way as if it were a static model. The MPO predictions, however, still need some consideration since the noise from the output of the model is fed back into the input as the uncertainty should grow as time increases. This problem is discussed in the literature within machine learning, for an overview see [138] or [141], and within engineering a Monte-Carlo approach has been previously explored as in [137] — although, to the authors knowledge, a comparison of these approaches has not been thoroughly presented. Presented here is the result of taking the mean output of the process and not propagating uncertainty through the model and two alternative methods of dealing with this issue of propagating the prediction noise through the model. The first is based on an Monte Carlo (MC) approach where the model is run multiple times with sampling. The second is based on work by Girard and Candela [142–144] which aims to analytically propagate the moments of the predictive distributions forward in time, this approach has so far not been presented for an engineering model. The novelty in the work presented here is to make a comparison between these approaches that cannot be found in machine learning literature and to examine the applicability of each approach in an SHM context.

3.2.1 Fixed Variance

The simplest option when making MPO predictions in the GP-NARX is to assume that the prediction variances are sufficiently small that predictions into the future can be made by taking the mean prediction as equivalent to a noise free observation. This methodology does not attempt to propagate any of the predictive variance through the process.

In this case the predictions are made starting from some known $\mathbb{E}\left[\mathbf{f}_{\star,-\mathbf{l_y}:-\mathbf{1}}\right]$ from which the auto-regression can start[1] at $t = 0$. Predictions are made such that there is a distribution $f_{\star,t}$ for the t^{th} step ahead which is the output of a GP-NARX model with the test input $\mathbf{x}_\star = \left[\mathbf{u}_{t-l_u:t}, \mathbb{E}\left[\mathbf{f}_{\star,\mathbf{t}-\mathbf{l_y}:\mathbf{t}-\mathbf{1}}\right]\right]$.

3.2.2 Monte-Carlo Sampling

The Monte-Carlo sampling approach accounts for the uncertainty in the process by making multiple sets of predictions and, rather than feeding back the mean prediction at each step, the value is set to a sample from the predictive distribution at that step. This procedure is summarised in Algorithm 1. This sampling approach incorporates the variance in the prediction into the future state of the model through sampling and running the model multiple times. This increases the computational demand of the model since it has to be run a (large) number of times to capture fully the variance in the predictions[2]. The posterior of the model is then approximated by all of the sampled paths each weighted $1/K$ where K is the number of MC draws that are made.

In this way the model is able to generate K sample paths ${\mathbf{y}_\star}^{(1:K)}$ each of which is a possible realisation of the process based on the predictive variances. Each of these realisations represents an equally weighted sample of the predictive process. This methodology for understanding the uncertainty in the model has the advantage of begin able to capture non-Gaussian multi-modal posterior distributions of the model which will converge towards the true distribution of the model as the number of MC

[1] The notation used here is similar to that seen in the Matlab software package. With $a : b$ indicating a range of integers from a to b inclusively. Prefacing this subscript with a star ($\star$) indicates that this is a test vector as opposed to the training vector $\mathbf{f}$.

[2] The approach of propagating uncertainty using an unscented transform could be adopted to reduce the computational load. This would, however, remove the guarantee of convergence to a true posterior in the limit of samples and may struggle with highly non-Gaussian posteriors.

Algorithm 1 Monte-Carlo Sampling for Uncertainty Propagation in GP-NARX

Initialise from known inputs $\mathbf{u}$ and outputs $\hat{\mathbf{y}}_{-l_y:-1}$
for $k = 1 : K$ **do** ▷ Number of Monte-Carlo Samples
  for $t = 1 : T$ **do** ▷ Number of Time Steps
    Predict $y_\star$ from GP based on $\mathbf{x}_\star = \left[u_{t-l_u}, \ldots, u_t, \hat{\mathbf{y}}_{t-l_y}, \ldots, \hat{\mathbf{y}}_{t-1}\right]$
    Sample $\hat{\mathbf{y}}_t \sim y_\star$
  end for
  ${\mathbf{y}_\star}^{(k)} = \hat{\mathbf{y}}_{1:T}$ ▷ The k^{th} sampled path
end for

samples increases. However, a large number of samples are needed to get an accurate representation of the process and each sample requires re-computing the model so each step can be sampled differently. This greatly increases the computational load of the model although it is trivial to create parallel implementations of MC type schemes which can help alleviate this problem. This approach is the only technique to handle uncertainty propagation in the GP-NARX model, which will guarantee to fully capture the posterior, the problem with this is that the guarantee only exists in the limit of an infinite number of Monte-Carlo samples. For this reason it is worth investigating alternative approximate methods to propagate the uncertainty.

3.2.3 Moment Matching Uncertainty Propagation

To avoid the computational load of the Monte-Carlo approach to propagating uncertainty through the GP-NARX model, Candela *et al.* [142] present an analytical solution to propagation of uncertainty due to uncertain inputs[3] in GP time series models when the uncertainty on the input is assumed to be Gaussian and when the covariance kernel used is a Squared Exponential[4]. Girard *et al.* [141] describes an alternate method for propagating the uncertainty based on a Taylor series expansion of the Gaussian approximation of the posterior which requires the computation of the partial derivatives of the predictive posteriors. It is the opinion of the author that the work in [142] is preferable since it avoids the second approximation introduced by the Taylor series expansion and is computationally less expensive.

[3]Although the term GP-NARX is not used in these papers the model for k-step ahead time forecasting is an identical model under a different name, what is not shown in the work of Candela *et al.* [142] is the incorporation of exogenous inputs, which is shown here.

[4]The Squared Exponential is sometimes also referred to as the exponentiated quadratic or as the Gaussian correlation function/kernel. It is the Gaussian-like form of the kernel which is exploited in the work by Candela *et al.* [142] in order to obtain closed form solutions to the marginalisation over the predictive input distribution.

In the standard GP formulation predictions are made by marginalisation of the joint Gaussian distribution defined by the covariance between the training inputs and the test inputs (Equation (2.10)) to give the predictive equations in Equation (2.11) and Equation (2.12). In the case of uncertain inputs it is necessary to compute the predictive distribution when marginalising out the test inputs $\mathbf{x}^\star$.

$$p\left(f^\star \mid \hat{\mathbf{u}}, S\right) = \int p\left(f^\star \mid \mathbf{x}^\star, \mathcal{D}\right) p\left(\mathbf{x}^\star \mid \hat{\mathbf{u}}, S\right) \mathrm{d}\mathbf{x}^\star \tag{3.2}$$

Here $\hat{\mathbf{u}}$ is the means of the predictive inputs and S is the covariance of the predictive inputs when $\mathbf{x}^\star$ is a multidimensional input vector of test points. This marginalisation is analytically intractable, however, it can be approximated in a number of ways. The first method would be to use a direct MC approximation of the full integral, this would be a costly approach and in many cases is not feasible. The alternative would be to approximate the predictive posterior to be Gaussian. This can be done in two ways, either make an exact approximation — the first and second order moments of the posterior predictive distribution Equation (3.2) are calculated exactly and used to define a Gaussian approximation of Equation (3.2). Alternatively, use a Taylor series approximation approach which was shown in [141, 145].

The details of the exact approximation process are laid out in full in [142] where, defining,

$$m_j = \left|\Lambda^{-1} S + \mathbb{I}\right|^{-1/2} \exp\left\{-\frac{1}{2}\left(\hat{\mathbf{u}} - \mathbf{x}_j\right)^\mathsf{T} \left(S + \Lambda\right)^{-1} \left(\hat{\mathbf{u}} - \mathbf{x}_j\right)\right\} \tag{3.3}$$

and also,

$$\begin{aligned} M_{ij} = {} & \left|2\Lambda^{-1} S + \mathbb{I}\right|^{-1/2} \\ & \exp\left\{-\frac{1}{2}\left(\hat{\mathbf{u}} - \mathbf{x}_d\right)^\mathsf{T} \left(S + \frac{\Lambda}{2}\right)^{-1} \left(\hat{\mathbf{u}} - \mathbf{x}_d\right) + \left(\mathbf{x}_i - \mathbf{x}_j\right)^\mathsf{T} \left(2\Lambda\right)^{-1} \left(\mathbf{x}_i - \mathbf{x}_j\right)\right\} \end{aligned} \tag{3.4}$$

with $\mathbf{x}_d = \frac{1}{2}\left(\mathbf{x}_i + \mathbf{x}_j\right)$ and the uncertain test input is distributed $\mathbf{x}^\star \sim \mathcal{N}\left(\hat{\mathbf{u}}\,,\ S\right)$; allows the mean prediction of the GP to be given by,

$$\mathbb{E}\left[f^{\star}|\hat{\mathbf{u}},\mathbb{S}\right]=\beta^{\mathsf{T}}\mathbf{m} \tag{3.5}$$

when $\beta = K\left(X,X\right)^{-1}\mathbf{y}$. It is clear that the vector $\mathbf{m}$ can be interpreted as the uncertain input equivalent to the covariance between the test inputs and the training inputs in the standard GP. In fact, when the covariance of the test inputs S is a matrix of zeros (i.e. there are certain inputs) the standard GP solution is recovered.

In an equivalent way, the predicitve variance can be calculated by the law of conditional variances such that,

$$\mathbb{V}\left[f^{\star}|\hat{\mathbf{u}},\mathbb{S}\right]=1-\mathrm{Tr}\left(\left(K\left(X,X\right)-\beta\beta^{\mathsf{T}}\right)M\right)-\mathrm{Tr}\left(\mathbf{m}\mathbf{m}^{\mathsf{T}}\beta\beta^{\mathsf{T}}\right) \tag{3.6}$$

where $\mathrm{Tr}\left(\cdot\right)$ is the trace operator. This is given for the case where $\sigma_f^2 = 1$ although it is trivial to extend this when an alternative value is used. Again it is clear that this expression will reduce to the standard GP predictive variance when there is no uncertainty on the inputs, i.e. $S = 0$. In this way it is possible to handle inputs which have uncertainty in them and to propagate this.

The technique shown here is applicable to a NARX type model when the predictions are uncertain and Gaussian as is the case with a GP-NARX model. In fact the procedure for propagation of uncertainty in this type of model is laid out in [142] (although not referred to as a GP-NARX model the structure is identical). Girard *et al.* [141] demonstrates the application of this model to two case studies. The input uncertainties are calculated recursively starting from some known inputs. To allow this propagation of uncertainty, the covariance between the outputs must also be calculated, for a time step k steps ahead in time,

$$\mathrm{cov}\left(y_{T+k},\mathbf{x}_{T+k}\right)=\sum_{j}\beta_j\mathbf{m}_j\left(c_j-\hat{\mathbf{u}}_{T+k}\right) \tag{3.7}$$

With β_j and $\mathbf{m}_j$ given as before and $c_j = \left(\Lambda^{-1}+S_{T+k}^{-1}\right)^{-1}\left(\Lambda^{-1}\mathbf{x}_j+S_{T+k}^{-1}\hat{\mathbf{u}}_{T+k}\right)$, since $\mathbf{x}_{T+k}\sim\mathcal{N}\left(\hat{\mathbf{u}}_{T+k}\,,\;S_{T+k}\right)$. Following this methodology it is possible to propagate the uncertainty through the model without needing multiple model runs and accounting for the accumulation of uncertainty in time. Discarding the last element of this vector $\mathrm{cov}\left(y_{T+k},\mathbf{x}_{T+k}\right)$ the cross covariance between the previous predictions and

the prediction just made can be calculated which provides the off diagonal terms of the covariance matrix S_{T+k}.

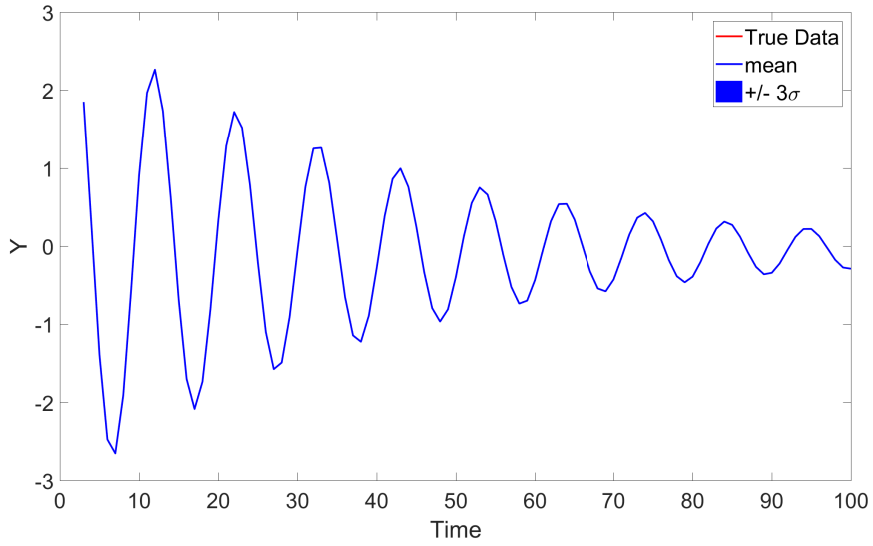
In the following section the three approaches will be compared in a numerical example which is known to be exactly modelled by a NARX formulation. The effect of propagating the uncertainty can then be analysed without being confused by any model form errors and also avoiding the problem of lag selection as is discussed in Section 3.4. It is maybe prudent at this point to make clear that all methods assume a training dataset where the inputs are noise free. This is consistent with the assumption made when using a static GP, although strictly speaking this may not be the case, it is normally not restrictive since the measurement noise of the inputs in SHM problems may be low — if for some reason this is not the case then care should be taken, possibly considering if the GP model is appropriate.

3.3 Comparison of Uncertainty Propagation Methods in GP-NARX

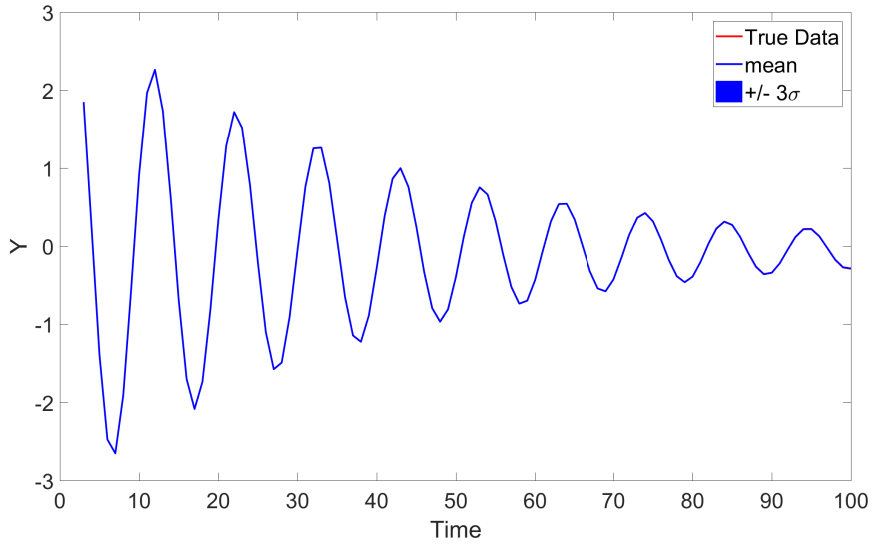
Here the three methods for 'handling' uncertainty in the GP-NARX model are compared; to take only the mean prediction, to propagate via Monte-Carlo sampling, or to propagate via moment matching[5]. These being to propagate only the mean, to use the Monte-Carlo approach, or the 'exact' approximation of [142]. This is demonstrated on three models; the first is a linear AR model simulation with low noise such that the GP-NARX model can fit it with very low variance, the second is a linear AR model where there is high process noise, the third example is a nonlinear AR model which exhibits switching behaviour. Since the test data is generated from (N)AR processes the model order of the GP-NARX can be chosen to be the same as the generating process *a priori* and the issue of lag selection is avoided here.

The first model considered is a discrete time AR(2) model with coefficients $\alpha_1 = 1.6$ and $\alpha_2 = -0.95$ simulated for 100 time steps. The process noise in the model is set to be Gaussian white noise with a variance of 1×10^{-4}. Figure 3.1 shows the ability of the GP-NARX to model accurately dynamic processes which have low process

[5]Throughout this section the term MPO prediction will be used to refer to predictions where the uncertainty in the model is not propagated. When referring to both the Monte-Carlo and moment matching methods the predictions are naturally full model predicted outputs as in the OSA case there is no uncertainty to propagate!



(a)



(b)

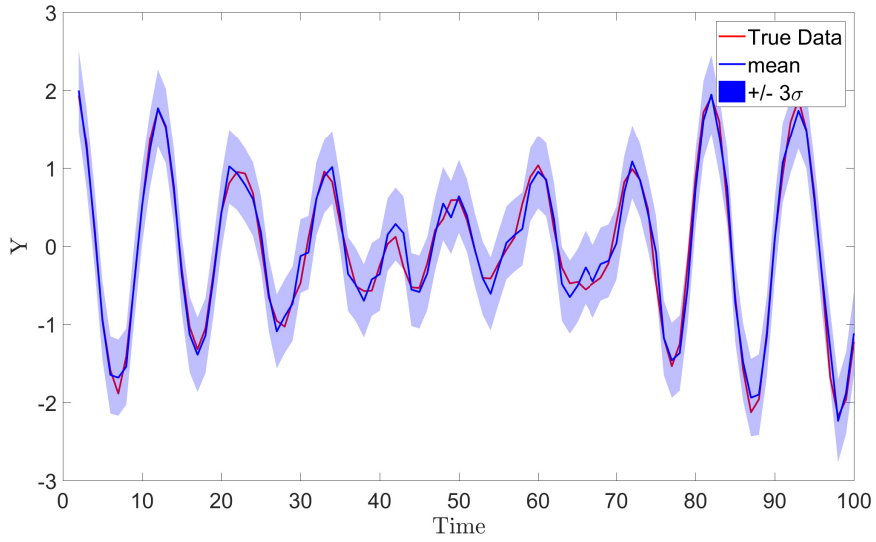
Figure 3.1: Figure showing the one-step-ahead (OSA) prediction (a) and model-predicted-output (MPO) prediction (b) for the AR model with low process noise, the GP-NARX is able to capture the function with low variance.

noise. The NMSE for both the OSA and MPO predictions is less than 1×10^{-3}. This shows an excellent fit of the model, it is also seen that the variance of the predictions is very low which demonstrates high confidence in the prediction. It is clear that under certain circumstances the GP-NARX is a powerful model for dynamical systems. What is of interest is how the model can handle the case where uncertainty is increasing and the model is forced to extrapolate from its previously seen training data. In this case it may seem sufficient to only make use of the OSA and MPO predictions of the model — for this reason there is no need to consider the uncertainty propagation on this simple model.

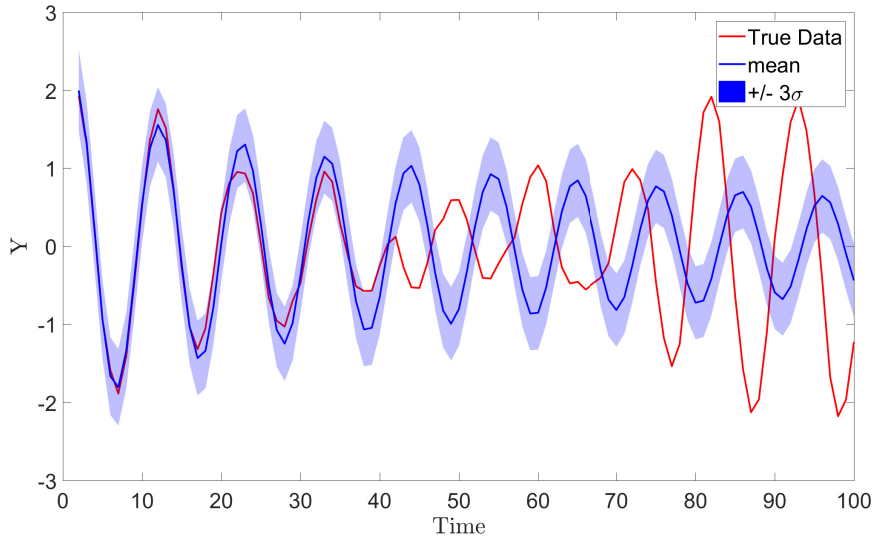
It has been shown that the GP-NARX is able to accurately model the system when the process noise in the model is low. That is that the variance of the posterior in the prediction of the GP-NARX is very low and that the mean follow the true signal closely. This is, unfortunately, hard to achieve when the GP-NARX model is not able to make predictions with the predictive variance very close to zero — this can be the case if a training set does not cover well the input space or if there is a model form error. The model is defined as before but with the process noise variance is set to be 0.1.

The OSA and MPO predictions of this noise AR process are shown in Figure 3.2. The effect of this increase in process noise, which is akin to the GP being unable to model fully the behaviour in an observed process such as the wave loading, has been to increase the predictive variance of the model since the noise hyperparameter increases. The prediction in the OSA case remains good with an NMSE of 2.111, however, this added noise causes the MPO prediction to become poor as the errors in the mean prediction are propagated forward without accounting for the added uncertainty resulting in an MPO error of 119.084.

The model was also run with the MC propagation of the uncertainty using 1000 Monte-Carlo samples each time starting independently from the first prediction. It can be seen more clearly that the increase in predictive variance causes each sample to take a different path which begin by following a similar trajectory but quickly diverge. For comparison, the distribution of the samples has been converted to a Gaussian at every time point to allow comparison to the other prediction methods by calculating the sample mean and variance at each time step. Taking this sample mean also allows comparison of the NMSE, which for this example is 78.693. The Gaussianised prediction shown in Figure 3.3(b) demonstrates more clearly than the sample paths in Figure 3.3(a) that the variance in the paths is appropriate.

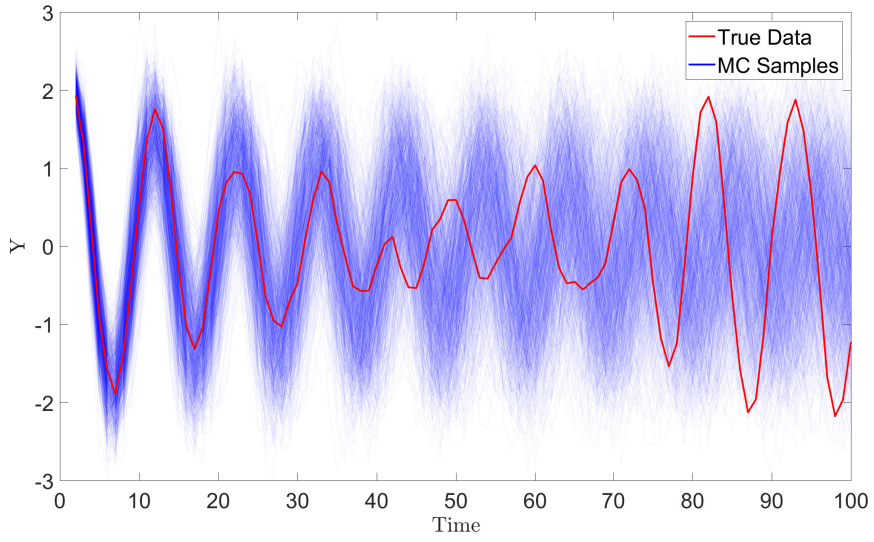


(a)

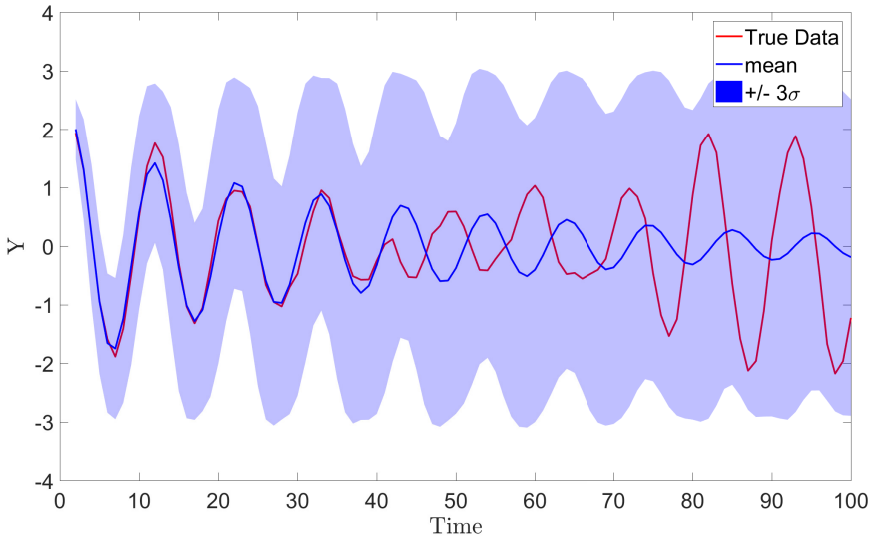


(b)

Figure 3.2: Figure showing the one-step-ahead (OSA) prediction (a) and model-predicted-output (MPO) prediction (b) for the AR model with high process noise the increased variance from the process noise has been well captured but this causes the MPO prediction to be poor.



(a)



(b)

Figure 3.3: Figure showing the samples from the MC prediction (a) and the Gaussianised distribution (b) for the AR model with high process noise.

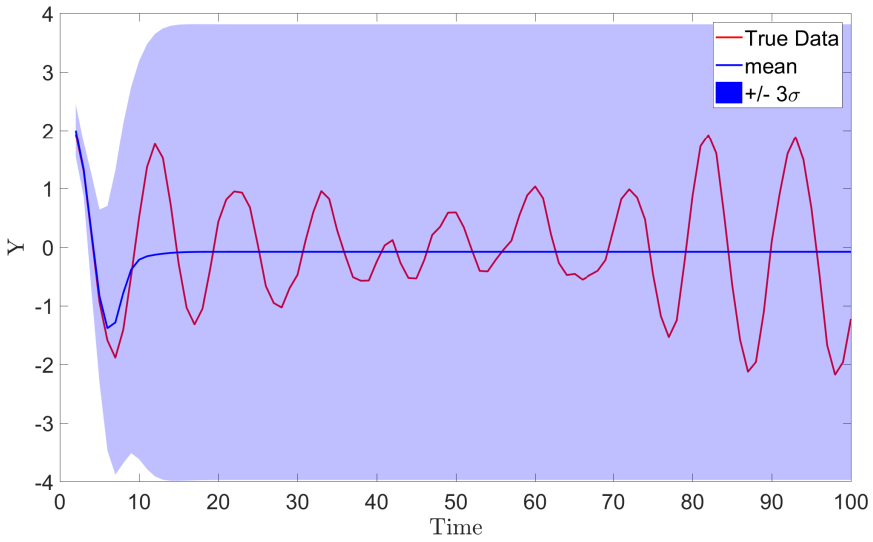


Figure 3.4: Figure showing the prediction of the moment matching uncertainty propagation method for the AR model with high process noise.

Also it can be seen that as time progresses the prediction is returning to the prior distribution of the GP.

The procedure for moment matching uncertainty propagation in the GP-NARX was also followed and the predictive distribution is shown in Figure 3.4. The NMSE for this method is 86.114, which can be explained following inspection of the predictive distribution. The effect of the moment matching approximation of the posterior increases the variance in the predictive distributions more quickly than the MC approach. This causes the variance to grow quickly and return to the prior in only a few time steps when the model has high noise. Since the NMSE is equal to 100 when the predictive mean is equivalent to the mean of the signal it is logical that the NMSE of this prediction is close to 100. What is demonstrated, however, that the moment matching approach is able to propagate the uncertainty from the predictions forward in time until the prediction converges on the GP prior. This is encouraging since it is expected that the model should return to the prior as time progresses and knowledge about the process decreases. Additionally, since, for all these experiments, only 1000 Monte-Carlo samples are used, it is possible that the variance in the MC model is being underestimated due to the lack of samples in the tails of the distributions approximated by the samples at each time point.

The final example used is one of a nonlinear system where the transition from time t

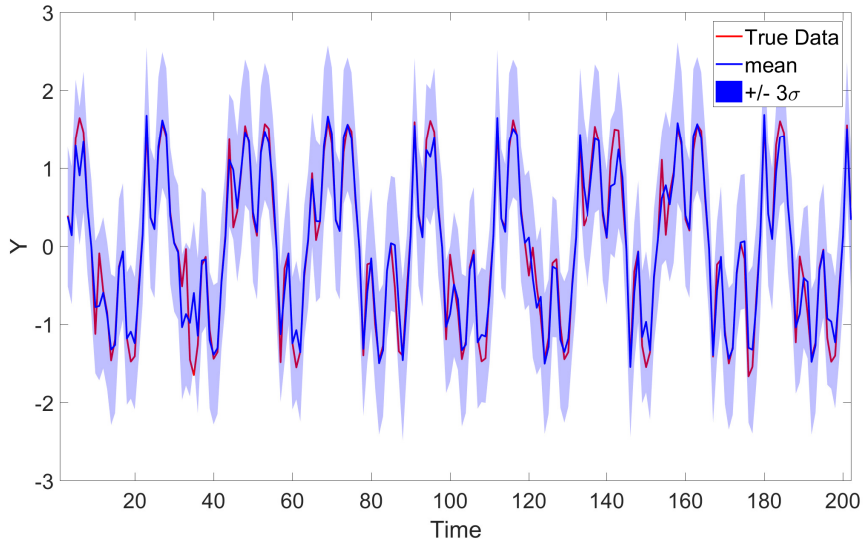
to time $t+1$ is governed by the following equation,

$$y_{t+1} = 0.5y_t + 25\frac{y_t}{(1+y_t^2)} + 8\cos(1.2t) \tag{3.8}$$

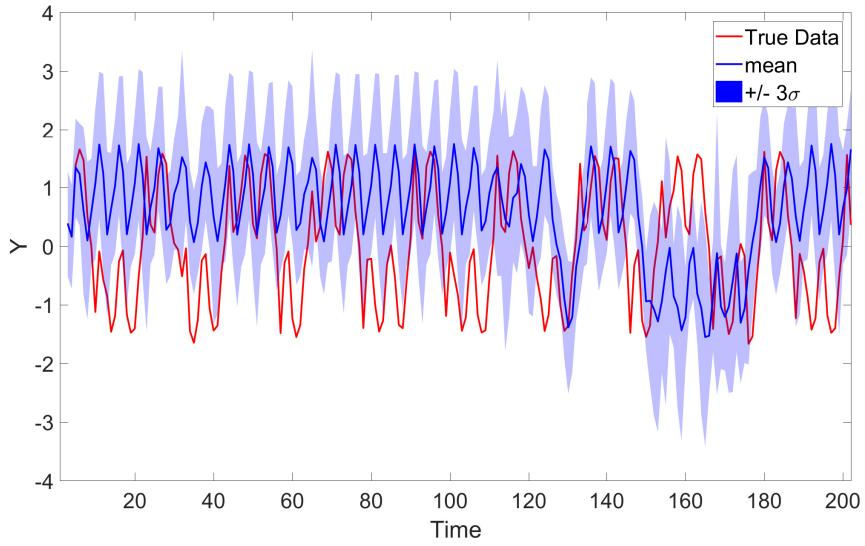
A measurement noise with variance 0.01 is added to the signal following its simulation. This model is an interesting nonlinear system because the nonlinearity causes a switching like behaviour in the model — this may be seen in a physical system which has a contact behaviour in it causing a discontinuity in the stiffness. This type of nonlinearity leads to a multi-modal posterior which presents a challenging problem when attempting to predict. This challenge is seen when attempting to use the GP-NARX to predict the system, even in the relatively simply OSA shown in Figure 3.5 (a) the NMSE is 5.620. It is unsurprising, therefore, that the MPO prediction shown in Figure 3.5 (b) does not manage to accurately model the function, with an NMSE of 188.835 — it is worth pointing out that this indicates the prediction is worse that just taking the mean of the data by some margin!

Considering the MC approach which is shown in Figure 3.6, it can be seen that the sample paths of the model split to capture the multi-modal posterior that arises from the switching behaviour. Converting these samples to a Gaussian distribution as before by calculating the first two moments gives the distribution shown in Figure 3.6 (b). This shows that the process is highly uncertain as the variance quickly increases and the prediction returns close to the GP prior, in which the hyperparameters of the model have been optimised to a space where the GP prior covers the distribution of the function. The NMSE of the sample mean is 99.648, which is consistent with the assertion that the GP-NARX model, unable to accurately model the function, has returned to the prior.

Considering the result from the moment matching approach, shown in Figure 3.7, it is seen that it tends towards the Gaussianised MC approach. This results in an NMSE of 98.756, which is in line with the results seen in the MC approach. The results seen in this section clearly indicate that propagating only the mean of the process in a GP-NARX model as is done in the usual MPO manner is unsatisfactory (in the sense of wanting to quantify the uncertainty in the model) unless the GP-NARX model is highly accurate with very low variance. Therefore, it is necessary to propagate the uncertainty through the model to capture function behaviour in a satisfactory manner. This approach, however, will not (strictly speaking) improve the NMSE of

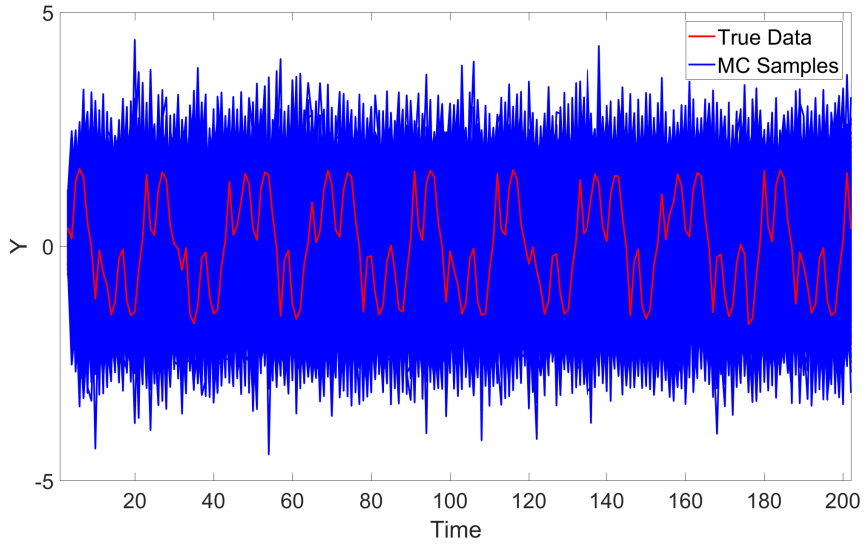


(a)

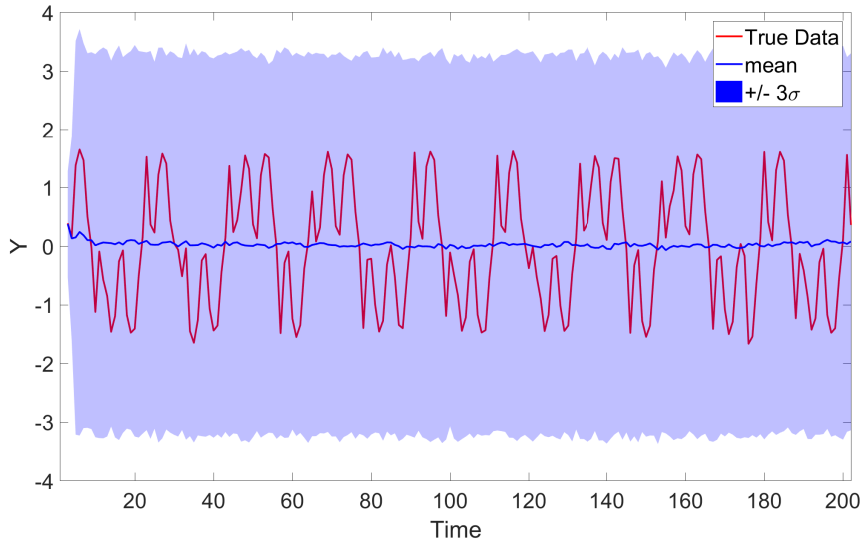


(b)

Figure 3.5: Figure showing the one-step-ahead (OSA) prediction (a) and model-predicted-output (MPO) prediction (b) for the nonlinear model.



(a)



(b)

Figure 3.6: Figure showing the samples from the MC prediction (a) and the Gaussianised distribution (b) for the nonlinear model.

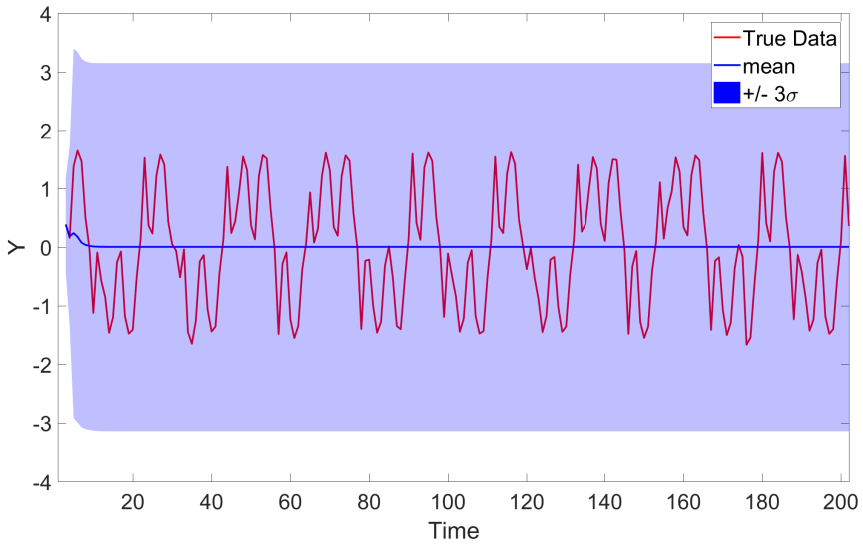


Figure 3.7: Figure showing the prediction of the moment matching uncertainty propagation method for the AR model with high process noise.

the model. Instead, the propagation of uncertainty merely informs the user if the model is producing valid results by comparison to the prior, and allows propagation of this uncertainty into further analysis steps.

Two approaches for this have been shown, each of which will at some time cause the prediction to return to the prior indicating that it is not possible to predict indefinitely. The MC approach has been shown to be more robust to multi-modal posteriors and it is seen that the moment matching approach can cause the model to return to the prior prematurely as a consequence of assuming the posterior can be modelled from only its first two moments — i.e. it is approximated to be Gaussian. It may seem that the MC approach is superior, however, the final consideration is the speed of computation. The cost of running one MC sample is approximately equal to that of running the MPO prediction and the moment matching approach is found to be approximately 10 times slower than the one-shot MPO prediction. Here, 1000 MC samples have been used, which is not a high number when considering the use of MC methods in other applications, but already the MC is ~ 100 times slower than the moment matching approach. This indicates that moment matching may be a feasible approach to uncertainty propagation in GP-NARX when the posterior distributions are approximately Gaussian, the time series is long (or there are other computational constraints), and when it is acceptable that the process will return to the prior more quickly. If a GP-NARX model is able to fit the data with very

low variance and high accuracy, it is believed that the moment matching approach represents the most viable uncertainty propagation technique.

The author believes this is the first time that the MC approach to uncertainty propagation has been directly compared to the moment matching procedure of [142]. The application of rigorous uncertainty propagation is a key issue for SHM applications due to the high risk in overconfident predictions, this will be a key factor in protecting against dangerous false negatives in an SHM system. Quantification of the true uncertainty (in this case through the uncertainty propagation but also generally) also allows for more robust incorporation of these techniques into a risk-based decision framework.

3.4 Lag Selection in GP-NARX

The problem of lag selection for the GP-NARX model is significant and not normally addressed in the Gaussian Process literature. As previously discussed the problem is an NP-Hard combinatorial integer programming problem which is (theoretically) unbounded. For example, if the process is governed by a phenomenon which may require an infinite number of lags to fully explain its behaviour. An exhaustive search of all possible lag combinations is too computationally expensive, even for small problems. Some form of optimisation must be used to determine the lags to be included in the model, not the model terms. A research area where this has seen more interest is in more general Nonlinear Auto-Regressive Moving Average with eXogeneous inputs (NARMAX) modelling, in particular polynomial NARMAX models [139]. The added complication in the polynomial NARMAX model is the selection of the polynomial model terms where the number of terms is given by $(l_y + l_u)^R + 1$ for a NARX model with l_y lags and a polynomial of order R. The number of terms only increases further with the addition of the exogeneous inputs. The use of a GP-NARX model reduces the number of terms significantly through its nonparametric representation of the nonlinear function. This avoids the need to estimate which polynomial terms, for example, are most appropriate for the function being modelled.

It is useful to consider how this term selection problem is performed in the NARMAX literature, mindful of the fact that only the lags themselves need to be chosen. These methods have included the use of an Error Reduction Ratios and Orthogonal

Least Squares [146, 147] and a forward modelling approach for this was presented in [148]. Alternatively Bayesian approaches have been considered for the variable selection problem [149, 150]. The Bayesian approach has been extended more recently using more sophisticated MCMC methods [151], bootstrapping [152], expectation maximisation [153], or variational approaches [154, 155].

Since the use of GP-NARX simplifies the lag selection problem considerably by removing the need to select model terms, it is more feasible to compute exhaustive searches, such as a grid search over the lags in the model. To further reduce the number of models which require learning, the novel work presented here, only optimises the maximum input and output lag. Then, kernels are chosen which have ARD. ARD allows inputs to be weighted differently in the model by placing a separate length-scale over each input. As well as allowing this to model effects on different scales for different lags (long or short term effects), if an input is unimportant the length-scale should become very large and the effect of that input will tend to zero. In this way the lag optimisation problem is reduced to a manageable size which allows exhaustive searches of the maximum lag combinations. The effect of this procedure is demonstrated on measured experimental data in Chapter 4.

3.5 Discussion

This chapter has presented the Gaussian Process as a powerful Bayesian model for regression tasks in SHM. Some of the drawbacks of the GP have also been introduced. The difficulty in fitting dynamic models has been shown, with the presentation of the GP-NARX as a model for dynamic behaviour using GP models.

The issue of handling uncertainty quantification and propagation in the GP-NARX model has been addressed. Two alternate methods for propagating the uncertainty through the model were shown, the first based on a Monte-Carlo and the second on an 'exact approximation' moment matching technique. These two approaches have been demonstrated on three numerical examples to demonstrate their strengths and weaknesses.

It has been shown that the moment matching approach to uncertainty propagation is capable of efficiently propagating uncertainty in the GP-NARX model. However, the approximation of the posterior by only considering its first two moments leads

to a Gaussian assumption and can lead to the variance increasing more quickly than in the Monte-Carlo setting — this behaviour has not been documented in the original papers but is intuitive given the approximation of the posterior to its first two moments at every step. The trade-off is that the Monte-Carlo approach, although able to capture multi-modal posteriors, is significantly computationally more expensive. As with most of the current machine learning techniques, it is both clear and unfortunate that there appears to be no 'silver bullet'. Care must be taken by the end user to account for the assumptions imposed by any model and to choose an appropriate methodology.

In fact, as will become clear in the following chapter, even having settled on using a Gaussian Process model which is nonparametric, there remain a number of steps before that model is usable. The determination of the hyperparameters of the model has been shown to be non-trivial, especially as the dimensionality of the input space increases. The problem of specifying a model form in the choice of kernel is another area where expert knowledge is still required. All of these assumptions, built upon the basic tenets of the Gaussian Process, can appear to restrict the type of data which can be used. These limitations can, however, be readily overcome through rigorous model selection and hyperparameter learning.

It is slightly unintuitive, but, these explicit difficulties in using the Gaussian Process model can actually be viewed as beneficial. The use of the Bayesian approach in the GP requires explicitly stating these modelling assumptions. This forces a user to consider whether, for instance, the uncertainty propagation in the model is being handled appropriately and the Bayesian approach provides a rigorous framework in which to do this. Contrast this to deterministic approaches such as standard neural networks where the choices of hyperparameters (number of layers, number of hidden units, etc.) can be hidden from the user and the outcome of the parametric learning of the weights is usually dependent upon unstated initial conditions and hyperparamters of the optimisation algorithms used.

In conclusion, the use of the Gaussian Process allows the move to nonlinear regression models without needing to specify a known basis — in fact the GP can behave as an infinite basis function set. The limitation of the GP as a static regression tool can be overcome through the use of dynamic GP models, here the use of the GP-NARX is presented. Methods for uncertainty propagation through this GP-NARX model have been presented and their relative merits discussed. The problem of learning specific models remains to be discussed in detail in the following chapter. If, within

engineering, it is to become commonplace for these types of black-box dynamic models to be used for making inferences about physical systems. It is essential that the models are robust to handling the quantification and propagation of uncertainty throughout their use. The challenge in applying a GP-NARX model is seen to be that there are a number of additional hyperparameters introduced in the lag selection and that the nature of the process is such that it will accumulate uncertainty throughout its lifetime. This is at odds with the desire from engineers to have models which will predict an unobserved process indefinitely, if the world is indeed a stochastic entity then the engineering community must come to terms with the fact that — in the absence of incorporating more information — the accumulation of uncertainty will add a shelf life to their models. This has serious implications when collecting further data is difficult and costly as is the case with wave loading for offshore. The application of this type of model is shown for wave loading in the following chapter where some of these difficulties become more apparent as the model is applied to engineering data.

Chapter 4

Wave Load Modelling

Highlights:

- *The modelling of wave loading — a key unknown — on offshore structures is attempted*
- *The current standard model, the Morison equation, receives a Bayesian treatment and is contrasted with a black-box approach*
- *The applicability of Gaussian Process models to wave loading (including GP-NARX) is explored and discussed*

A key source of uncertainty in the offshore industry surrounds the loading history that a structure has experienced. There are two concerns for ensuring structural integrity of offshore structures,

1. Occurance of extreme loading events — extreme waves, vehicle impact with structure, etc.
2. Fatigue damage accumulation from environmental loading — wind, waves, operational

This chapter will focus on the second of these. The time history of loading is very difficult to measure practically and its estimation remains a challenge. The state-of-the-art is to use modal expansion techniques [57, 58, 156, 157] to recover the

strain history from an updated finite element model. A methodology for performing this kind of assessment is shown by Tygesen *et al.* [158] and also discussed in [159]. This approach must be combined with a robust model updating approach which can have a number of challenges. The first of these is determination of the quantities used for the updating procedure which are usually obtained via Operational Modal Analysis (OMA) [38, 160]. This can be difficult in the offshore environment where there is potential for the loading to be non-Gaussian and close to the resonance of the structure. An approach to overcome this is presented in Chapter 6.

Once data become available which can be used to perform the model update a variety of techniques can be used. Although there is not room to fully discuss these here the reader can find an introductory text in [4]. The task of model updating is challenging in itself and often overlooked is the role of model discrepancy in the process where models are calibrated (or updated in engineering terms) but with a functional difference between the model and the true physical process. An approach for handling this within a Bayesian framework was introduced by Kennedy and O'Hagan [161] — although this approach does not remove all complications [162, 163], flexibility in the discrepancy function can still lead to identifiability issues.

If in possession of a calibrated model, for the physical structure, the wave force models also require their own separate calibration, which may be far harder due to the lack of availability of data — this is due to both practical (sensor damage) and cost reasons. This calibration can be achieved via Monte-Carlo simulation of wave states based on a measured wave spectrum but still the forcing on a structure must be determined. This could be done, in some circumstances, via a full Direct Numerical Simulation of the Navier-Stokes equations [164], however, this is computationally very expensive to the point where it becomes non-feasible for full scale applications. In reality a simplified physical model is often used; popularly, the Morison equation [64, 165]. Research continues into building more accurate simplified models of the wave loading process, a good overview of this can be found in [166], many of these models remain based on empirical relationships drawn from scaled down test data collected in a laboratory setting.

To explore the modelling of wave loading, a dataset is used which was collected as part of the Christchurch Bay Compliant Tower project in the UK. The data from this project has been used extensively in literature [167–172] and remains one of few datasets collected outside a laboratory for understanding wave loading. The full setup consists of two columns placed offshore (in shallow water) and instrumented at

five different levels. Additionally, measurements were taken of the particle velocities for the fluid flow close to the small cylinder (but slightly offset), these were used to acquire measurements of the particle acceleration. To complement these there are measurements of the tide and of wave height at the small cylinder. The force exerted on both the small column was measured by means of force sleeves, the force on the main column was measured only at the third level, again by means of a force sleeve. The small column is considered here only at level three, this column has a diameter of 0.48m, data are collected at a sample rate of 13.25Hz. Further details regarding the setup can be found in [172].

For the work shown here, three sets of data from across a single run are taken. Each of these being 1000 points long. The datasets are selected to have independent training, validation, and testing sets for each prediction method. Results are compared based on performance on the unseen test set. These data are chosen such that, as far as is possible, the wave state is unidirectional and there is no vortex shedding. This allows the use of the 1D Morison equation [166] and the forces are only calculated in line with the wave loading direction (x-direction).

For the experiments carried out here, data are collected from the 'small column' for which the force at level three is considered as the output for a regression model. As inputs to the regression model, the velocity and accelerations of the flow are measured.

4.1 White-Box Modelling

Here the choice of a physics based or *white-box* model for wave loading is considered. Not considered is the approach of Direct Numerical Simulation of the Navier-Stokes equations or other Computational Fluid Dynamics models as this would require its own significant body of work. Suffice to say these methods share the fact that they require a large amount of computing resource and require a skilled modeller to ensure that appropriate settings and models are used. The approach of Computational Fluid Dynamics is by no means trivial and represents a powerful modelling tool but can still be difficult to calibrate, verify, and validate. The use of this approach would also only be possible if the flow properties were known exactly (the full flow field), which they are not for this dataset. The full modelling of wave loading in this way still seems out of reach — as Sarpkaya [166] puts it "Nature has its vagaries and no

one knows it better and understands it better than the offshore industry".

Instead this work focuses on the use of computationally efficient simplified physical models for the force exerted by wave loading. The currently adopted industry standard is to make use of the Morison equation [165].

$$F = \underbrace{\rho C_d A_p}_{\hat{C_d}} |U|U + \underbrace{\rho C_m V_0}_{\hat{C_m}} \dot{U} \tag{4.1}$$

$$= \hat{C_d}|U|U + \hat{C_m}\dot{U} \tag{4.2}$$

This simplifies the model of wave loading down to a two term linear in the parameters model where the first term is related to the added inertia experienced by the structure. This term is dependent upon the value of $|U|U$ where U is the particle velocity in the flow. The second term relates to the drag experienced by the structure which is dependent on $\dot{U}$, the particle acceleration. Each of these terms depend on a number of constants, ρ the fluid density, A_p the projected frontal area, V_0 the volume of fluid displaced by the structure, and C_d and C_m which are two empirical coefficients. This form allows analysis of structures in different locations with different dimensions by non-dimensionalising the equation. The empirical coefficients are determined from literature but are generally geometry specific. When fitting a model on known data it is possible to combine all of these constants into a single value for each term, $\hat{C_m}$ for the first term dependent upon velocity and $\hat{C_d}$ for the second term dependent upon the acceleration.

A more thorough introduction to theoretical models for wave loading can be found in Sarpkaya [166] or Wilson [64]. All of these models, however, have a similar form to the Morison equation and most modifications are based on empirical work. By increasing the complexity of these models it is required that more information is known about the structure, e.g. its displacement in the fluid which can be difficult to estimate.

For determination of the parameters in the model there are a number of choices and, given the availability of particle velocity and acceleration data, the procedure is relatively simple given the linear form of the model, i.e. it is possible to perform an ordinary least squares solution [20]. The work in this chapter approaches the problem of wave load identification from a regression standpoint (constructing models

between the particle velocities and accelerations and the force), therefore, from now on it may be useful to write the problem in the matrix form of a linear model.

$$\mathbf{f} = X\mathbf{w}^{\mathsf{T}} + \varepsilon \quad \varepsilon \sim \mathcal{N}\left(0\,,\ \tau^{-1}\right) \tag{4.3}$$

This linear model does not imply that the process is linear, instead it shows that the model is linear in the parameters given the basis function expansion into the design matrix. Here the design matrix X can be formed by stacking the column vectors of the input variables together such that $X = \begin{bmatrix} |U| & U & \dot{U} \end{bmatrix}$ these are then multiplied by the weights of the model which in this case are given by $\mathbf{w} = \begin{bmatrix} \hat{C_m} & \hat{C_d} \end{bmatrix}$. The model also encodes an assumption of Gaussian white noise with a known precision τ, this is equivalent to a Tikhonov (ℓ_2) regularisation in a deterministic framework [20, 173].

The ordinary least squares solution relies on computing the Moore-Penrose pseudo-inverse of the design matrix X. This allows a computation of a 'best' set of parameters $\hat{\mathbf{w}}$.

$$\hat{\mathbf{w}} = \left(X^{\mathsf{T}}X\right)^{-1} X^{\mathsf{T}}\mathbf{f} \tag{4.4}$$

This model doesn't account for the value of τ in favour of minimising the amount of the function that is explained by the noise. Bayesian estimation in linear models is also relatively simple and can be computed with relative efficiency since the distributions are conjugate and, therefore, many of the required integrals can be solved in closed form. The aim is to compute the posterior distributions over the weights $\mathbf{w}$ and the precision parameter τ. The full calculation can't be completed in closed form, however, it is possible to calculate the posterior via an MCMC technique where the joint posterior can be approximated by sequentially drawing samples where each sample is a valid sample from the true joint posterior. These can then bes used as a point mass approximation to the true distribution. Or, since in many cases one is interested in sampling from the posterior anyway, the samples can be used directly. Gibbs sampling, an efficient MCMC technique can be used in this case [32]. It proceeds by sampling in turn from each of these conditional distributions where sampling from the conditionals will generate valid samples from the joint.

The Gibbs sampler works when it is possible to sample from the conditional distributions of the posterior but not the joint (it would work if the joint could be

computed analytically it would just be a waste of time since the joint distribution is the object of interest that fully defines the model). In the case of a Bayesian linear regression such as the one shown in Equation (4.3) this is possible. The Gibbs sampler alternates between sampling the weights of the model, which are the parameters being determined, $\mathbf{w}$ and the precision value τ. Since the posteriors over both $\mathbf{w}$ and τ are being computed in a Bayesian manner it is necessary to place priors over these values.

The priors used for the parameters of the model of wave loading on the Christchurch Bay tower are that the weights are distributed according to the Gaussian distribution,

$$p(\mathbf{w}) = \mathcal{N}\left(\begin{bmatrix}150\\150\end{bmatrix}, \begin{bmatrix}1\times10^{8} & 0\\ 0 & 1\times10^{8}\end{bmatrix}\right)$$

and the precision is distributed according to tha Gamma distribution given by,

$$p(\tau) = \mathcal{G}a(1,2)$$

.

The choice of these prior distributions ensures appropriate support for the parameters and has the benefit of being conjugate to the likelihood which is Gaussian. The high variance over the weights of the model $\mathbf{w}$ encodes that the belief regarding these parameters is not strong *a priori*. It is also possible to set these priors based on the results of a simpler method for determining the parameters such as the ordinary least squares solution. The conditionals that are calculated are the distributions $p(\mathbf{w} \mid \tau, X, \mathbf{f})$ and $p(\tau \mid \mathbf{w}, X, \mathbf{f})$. Given these priors and the likelihood model which is,

$$p(\mathbf{f} \mid X, \mathbf{w}, \tau) = \mathcal{N}\left(X\mathbf{w}^{\mathsf{T}},\ \tau^{-1}\right)$$

the conditionals can be written down in closed form. Where the conditional distributions are given by,

$$\begin{aligned}\hat{\sigma}^2 &= \frac{1}{\tau_1 + \tau^{(k-1)} X_{:,1}^{\mathsf{T}} X_{:,1}}\\ \mathbf{w}_1^{(k)} | \mathbf{w}_2^{(k-1)}, \tau^{(k-1)} &\sim \mathcal{N}\left((\mu_1\tau_1) + \tau^{(k-1)}\left(\left(\mathbf{f} - X_{:,2}\mathbf{w}_2^{(k-1)}\right)^{\mathsf{T}} X_{:,1}\right)\hat{\sigma}^2,\ \hat{\sigma}^2\right)\end{aligned} \tag{4.5}$$

$$\begin{aligned} \hat{\sigma}^2 &= \frac{1}{\tau_2 + \tau^{(k-1)} X_{:,2}^{\mathsf{T}} X_{:,2}} \\ \mathbf{w}_2^{(k)} | \mathbf{w}_1^{(k)}, \tau^{(k-1)} &\sim \mathcal{N}\left((\mu_2 \tau_2) + \tau^{(k-1)} \left(\left(\mathbf{f} - X_{:,1} \mathbf{w}_1^{(k)} \right)^{\mathsf{T}} X_{:,2} \right) \hat{\sigma}^2 \,,\; \hat{\sigma}^2 \right) \end{aligned} \tag{4.6}$$

$$\tau^{(k)} | \mathbf{w}_1^{(k)}, \mathbf{w}_2^{(k)} \sim \mathcal{G}a\left(a + \frac{N}{2}, b + 0.5 \left(\mathbf{f} - X \left(\mathbf{w}^{(k)} \right)^{\mathsf{T}} \right)^{\mathsf{T}} \left(\mathbf{f} - X \left(\mathbf{w}^{(k)} \right)^{\mathsf{T}} \right) \right) \tag{4.7}$$

Algorithm 2 Gibbs Sampler for Inference of Morison equation parameters

Set $\mathbf{w}^{(0)}$ and $\tau^{(0)}$ ▷ Start points for the Gibbs sampler
$p(\mathbf{w}_1) = \mathcal{N}\left(\mu_1 \,,\; \tau_1^{-1}\right)$ ▷ Prior for first weight
$p(\mathbf{w}_2) = \mathcal{N}\left(\mu_2 \,,\; \tau_2^{-1}\right)$ ▷ Prior for second weight
$p(\tau) = \mathcal{G}a(a, b)$ ▷ Prior for noise precision
Set K as number of steps
for $s = 1, \ldots, K$ **do**
    Sample $\mathbf{w}_1^{(k)} | \mathbf{w}_2^{(k-1)}, \tau^{(k-1)}$ from Equation (4.5)
    Sample $\mathbf{w}_2^{(k)} | \mathbf{w}_1^{(k)}, \tau^{(k-1)}$ from Equation (4.5)
    Sample $\tau^{(k)} | \mathbf{w}_1^{(k)}, \mathbf{w}_2^{(k)}$ from Equation (4.7)
end for
Discard burn-in samples

Since there is no cross-covariance between the weights in the model $\mathbf{w}$ these can be sampled independently. The procedure of Gibbs sampling for general priors is summarised in Algorithm 2. Although it is equally possible to sample in the case of correlated parameters, this would require the joint conditional of the weights given the data and the noise.

Here, the convention is adopted that a subscript to a vector indicates selecting selecting an element from the vector, e.g. $\mathbf{a}_i$ is selecting the i^{th} element of the vector $\mathbf{a}$. For matrices a similar notation is adopted but also following the indexing conventions of Matlab where a colon indicates selecting an entire column or row and the indexing is column-major. The notation of superscripted parentheses is used for indexing which step of the Gibbs sampler is being chosen. For example, $\mathbf{w}_1^{(k-1)}$ corresponds to the first value in the weight vector at step $k-1$ of the Gibbs sampler. Please accept the author's apologies for the somewhat cluttered looking equations!

Putting aside the not inconsiderable amount of book keeping required to implement

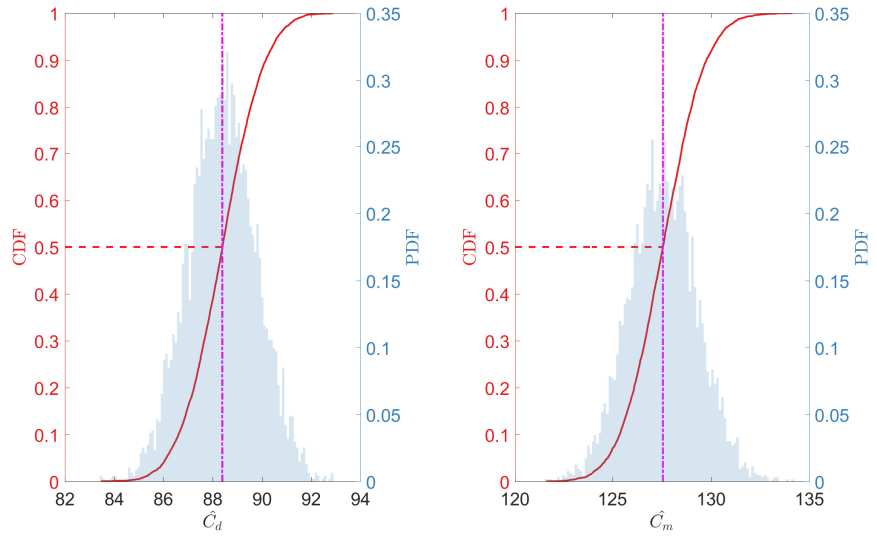


Figure 4.1: Posterior distributions over C_d and C_m from Gibbs sampling the Morison equation. The distributions over the parameters are shown as both the PDF in blue and the CDF in red. The mean is shown with a solid purple line and the estimate relating to a CDF value of 0.5 is shown in dashed red.

the Gibbs sampler, it is a very powerful and surprisingly simple method for obtaining fully Bayesian solutions in this type of linear model. Running the sampler for 5200 steps to obtain 5000 samples with a burn-in of 200 samples it is then possible to obtain posteriors over the parameters and the noise precision. The choice of the number of samples is generally limited by computational power but it is necessary to obtain a number which accurately approximates the distributions of interest. Gelman *et al.* [33] discuss a number of techniques for assessing the quality of a Markov chain.

It can be seen that the Gibbs sampler returns distributions that are symmetrical and unimodal over the parameters, shown in Figure 4.1. A similar shape distribution over τ is also observed. These samples could be propagated through the model in a Monte-Carlo manner to account for the parameter uncertainty in the predictions of the force made by the model. Since the variance over the parameters is relatively small and the distributions are unimodal it is also possible to take a *maximum a posteriori* (MAP) estimate of the paramters and propagate this through the model.

The additional variance introduced from the parameters is in fact very small, this can be seen when all the sample paths are plotted in Figure 4.2, all 5000 sample paths for each set of sampled parameters are overlaid in blue with the measured function in red. It would appear that the variance in the parameters does not account for the

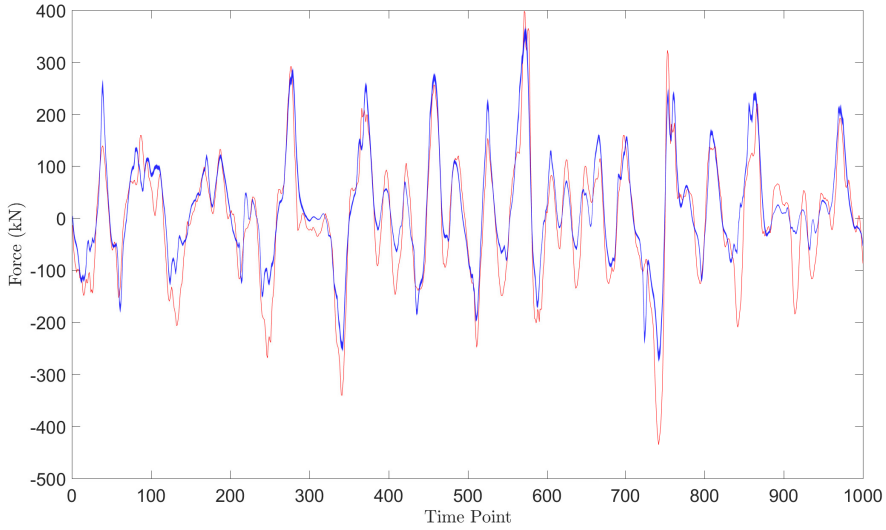


Figure 4.2: Monte-Carlo realisations of the wave loading by propagating the Morison coefficients sampled from the Gibbs sampler, with the samples shown in Figure 4.1.

	OLS	**Gibbs**
$\hat{C}_d$	90.71	88.39 (1.37)
$\hat{C}_m$	135.20	127.56 (1.73)
τ	-	2.90×10^{-4} (1.29×10^{-5})

Table 4.1: Values found for the parameters through ordinary least squares, optimisation, and via the Gibbs sampler (mean with standard deviation shown in brackets).

errors observed. This would indicated one of two things; either the process has a high level of noise or there is a model form error and the Morison equation is unable to fully describe the wave loading process.

It is possible to compare the predictions on the test data from the models identified by each of the methods; OLS and Gibbs sampling. These are shown in Figure 4.3, the top panel shows the measured force in red with the prediction of the model with the OLS parameters in blue. It can be seen that the model for the OLS has an NMSE of 19.528.

The model learnt via the Gibbs sampler can have the distribution over the predictions plotted since the model is learning the noise distribution in addition to the model parameters. The bottom panel of Figure 4.3 shows this, the measured forces are shown in red with the mean of Gibbs sampled model in blue and the predictive

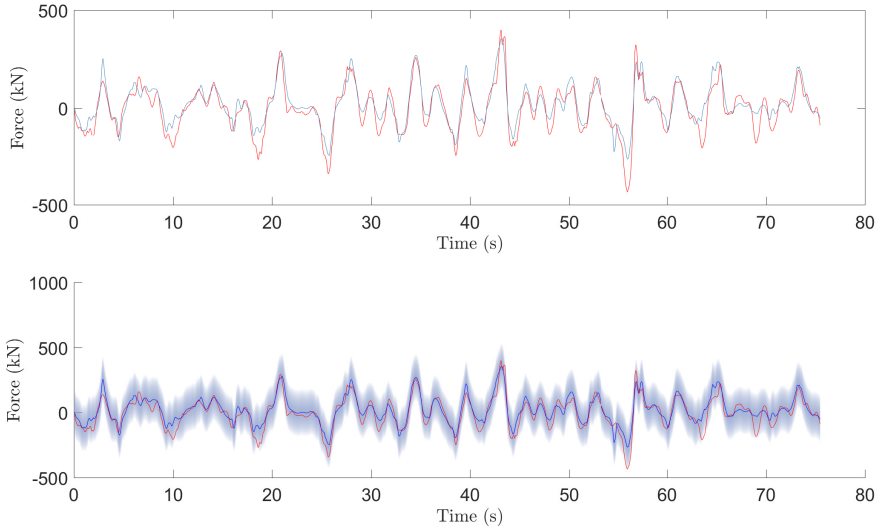


Figure 4.3: Predictions on an independent test set with the measured values shown in red. The top panel show the results of the OLS and optimised paramters, the bottom panel shows the predictions of the MAP estimates from the Gibbs sampler with the distribution over the noise shaded.

distribution is shown up to the 3σ level by the shaded area. Considering the mean prediction of this model the NMSE is 19.547, this is close to the performance of the other models but there is additional understanding added through quantification of the uncertainty in the model.

This remains a valid approach to determining the loading on the structure, the added information of using a Gibbs sampler can help when implementing a Risk & Reliability based decision framework. The purpose of the following work is to establish whether it is valuable to move to a fully data-driven approach.

4.2 Black-Box Modelling

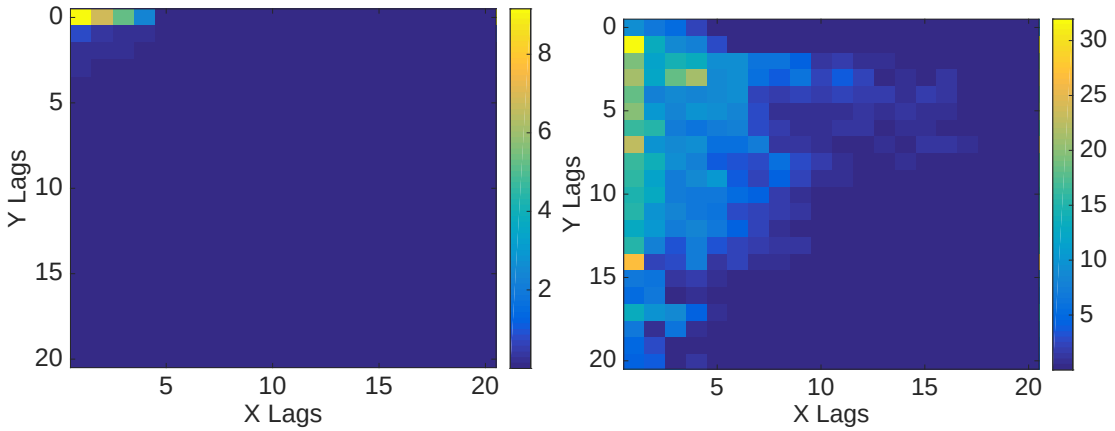
The, philosophically, alternative approach to the white-box model would be to rely fully on a data-driven approach. In the offshore sector this is still an emerging technology area. There has been investigation into using machine learning approaches to aid a number of problems. This includes the prediction of scour around pipelines [174]. Malekmohamadi *et al.* [175] discuss the use of machine learning techniques for prediction of wave height. Fernández *et al.* [176] approach the wave modelling

probem from a different perspective, instead building a classifier based model for wave height. In [177] and [178] the problem of wave loading modelling is tackled with a polynomial NARMAX model, this work is extended to use a GP-NARX model of the same order in [96]. Here a GP-NARX model is used to capture the dynamic behaviour of the wave loading. This fully data-driven approach can be though of as a *black-box* model which provides not physical insight into the behaviour of the system. The Gaussian Process NARX model, as previously introduced provides a framework for building Bayesian regression models of dynamic systems. The difficulties in using the model have also been discussed. These include the selection of lags, choice of kernel, optimisation of hyperparameters, and the propagation of uncertainty when predicting. The work shown here goes beyond that seen in [96] by considering the problem of lag selection and presenting the use of a cross validation training approach to learning the GP-NARX model. Here it will be shown how the choice of cost function in the optimisation can affect the results of training the model and how the lag selection complicates the learning process.

There is little prior knowledge regarding the form of covariance function that most appropriately captures the behaviour in the wave loading data being considered and so the Matérn 3/2 kernel is used, which is a powerful and general nonlinear kernel [92]. The difficultly in lag selection is addressed here via grid search up to a pre-selected maximum lag in both the input and output coupled with the use of a kernel with Automatic Relevance Determination (ARD) where a separate lengthscale is placed over each input dimension, this should allow uninformative inputs to have a minimal effect on the model.

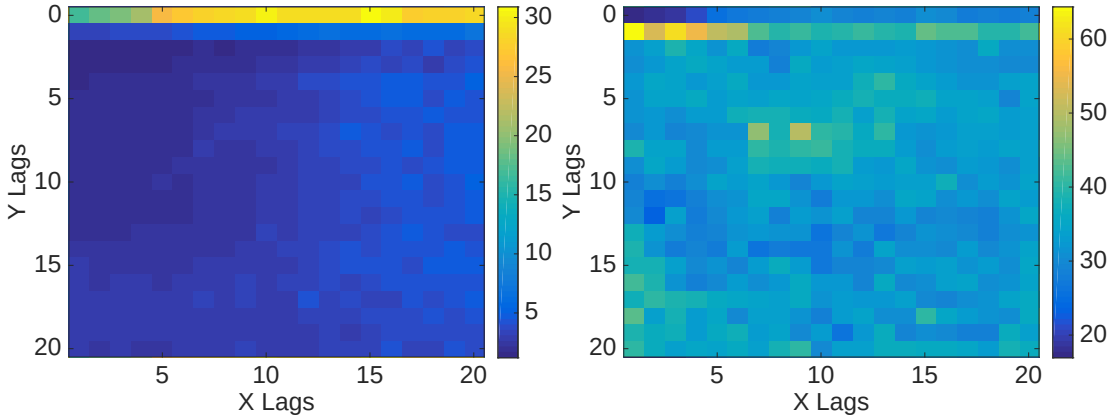
The same data as the was used in the white-box experiment are used here to test the purely data-driven approach to wave load modelling. The physics driving the wave loading process is known to be dynamic, therefore, a GP-NARX model is tested as well as a static GP. This leads to showing how a grid search for combinations of the input lags $l_u = 1, \ldots, 20$ and output lags $l_y = 0, \ldots, 20$ could be achieved when twenty is chosen as the maximum number of lags in the input and the output — this number is mainly limited by computational expense. For each of the lag combinations, the Gaussian Process kernel hyperparameters have to be optimised.This proceeds in the standard way via minimisation of the negative log marginal likelihood, the QPSO is used for the optimisation.

To assess the model it would be usual to consider the error between the prediction and the measured signal. The normalised mean-squared error provides an interpretable



(a) OSA (b) MPO

Figure 4.4: Heat maps of the normalised mean-squared error of the training data for the OSA and MPO predictions.



(a) OSA (b) MPO

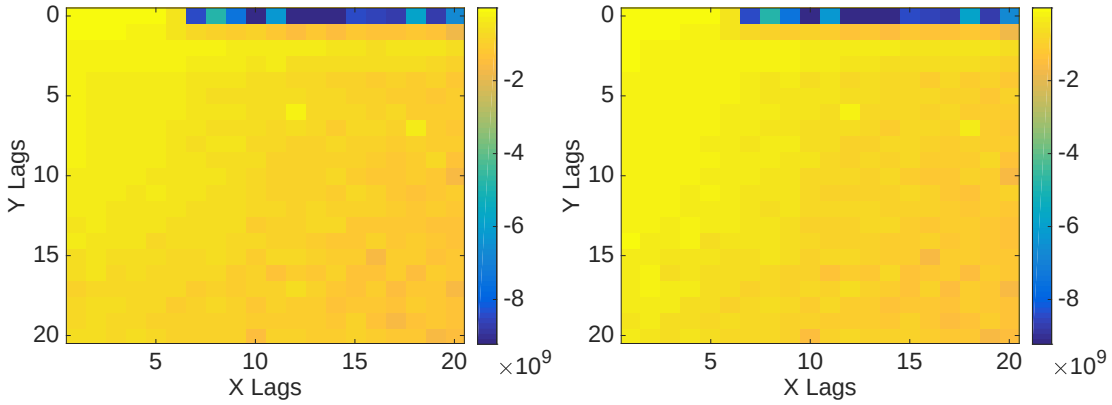
Figure 4.5: Heat maps of the normalised mean-squared error of the test data for the OSA and MPO predictions.

way to do this, where a score of zero corresponds to a perfect prediction and a score of 100 is equivalent to predicting the mean of the test data at every point. Figure 4.4 shows the NMSE when predicting the training data using the optimised GP for each lag combination. The results shown in these figures are as expected, the highest error is seen with the fewest lags in both the input (X Lags) and the outputs (Y Lags). The error then plateaus close to zero after a sufficient number of lags are added to capture the dynamics of the process. As has been shown before the OSA predictions have lower error and this stabilises with fewer lags. The MPO check of the training data is shown to be harder, with higher errors overall and more lags are required before the error stabilises close to zero.

It is, of course, important to ensure the model performance by considering the prediction quality on an independent test set. The same plots for the NMSE in the case of both OSA and MPO predictions on an independent test set are shown in Figure 4.5. The story that these plots tell is very different to that seen in the training data and in an unexpected way. Considering first the performance in the OSA predictions (and ignoring for now the static GPs with no lags in the output), the best performing modes are those with two or three lags in the outputs and up to five lags in the input. As the number of lags is increasing, the model performance is degrading. The likely cause stems from the fact that the GP is an interpolating model; as the number of lags increases so does the dimensionality of the space which the model interpolates, this causes the models to have lower covariance. For example when twenty lags are considered for there to be high covariance (which allows a good prediction to be made) then all of the previous twenty points in the sequence have to be close to the training data which is far less likely unless the signals are identical. This is coupled with the loss of sensitivity that is seen as the dimensionality of the models increases due to the kernels being based on a distance metric [179]. The static models, with only lags in the input, are seen to perform worse than the dynamic models for the OSA prediction for all lags. Interestingly, the performance of those models also decreases with increasing numbers of lags.

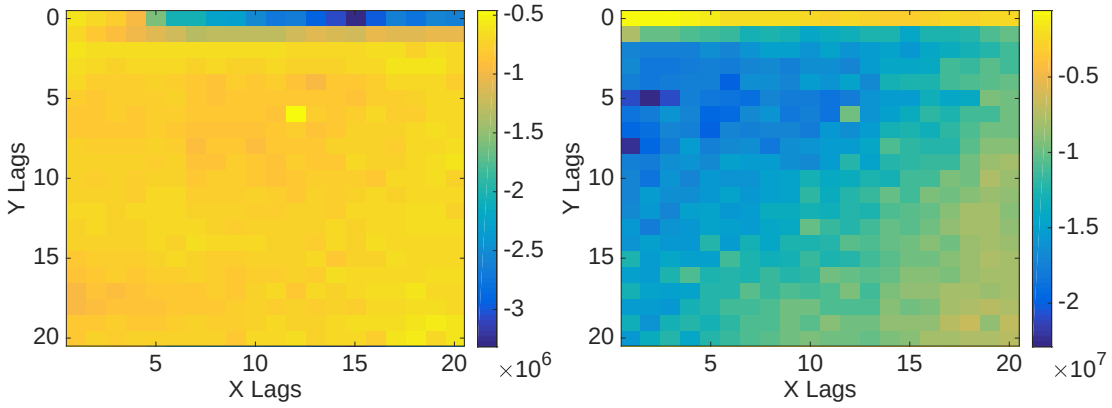
Now considering the MPO performance of the models in terms of the NMSE, it is seen that the best model is the fully static model. Following this many of the dynamic models have similar scores for the NMSE. The MPO score of the static model is 16.92 which is an improvement over the results seen from using the Morison equation. Despite the success of this model in reducing the error compared to the Morision equation, it is interesting/important to consider why the dynamic models are performing worse when the wave force process is known to be dynamic. This is actually quite simple to explain, it is known that the model is unable to make perfect predictions (zero variance, zero error) and the GP prediction will return to the mean as covariance between the test point and training points decreases. This causes the model predictions to be slightly wrong and this error is fed back into the next prediction which lowers its covariance and so on and so forth. This is all without propagating the uncertainty which will only exacerbate the problem — the model will return to predicting on the mean of the process faster and the NMSE will tend to 100.

There is more information to be found in the results seen here. Looking instead at the



(a) OSA (b) MPO

Figure 4.6: Heat maps of the log posterior likelihood of the training data for the OSA and MPO predictions.



(a) OSA (b) MPO

Figure 4.7: Heat maps of the log posterior likelihood of the test data for the OSA and MPO predictions.

(log) posterior likelihood of the predictions $p(\mathbf{y}_\star \mid X^\star, \mathbf{y}, X)$ tells a very different story and highlights an important point. Looking at both the OSA and MPO posterior likelihood for the training data, shown in Figure 4.6, the maximum value is found for the static model and the opposite effect to the NMSE is seen with increasing lags. This says that the models become less likely as the lag number increases.

Looking now at the test data in Figure 4.7. For the OSA predictions a similar picture emerges except for one model (12 lags in the input and 7 in the output) which has the highest likelihood. In the MPO model all of the predictions with only input lags perform better than the dynamic models. This mirrors the scores seen in the NMSE for the MPO results. The important thing to extract from these results is

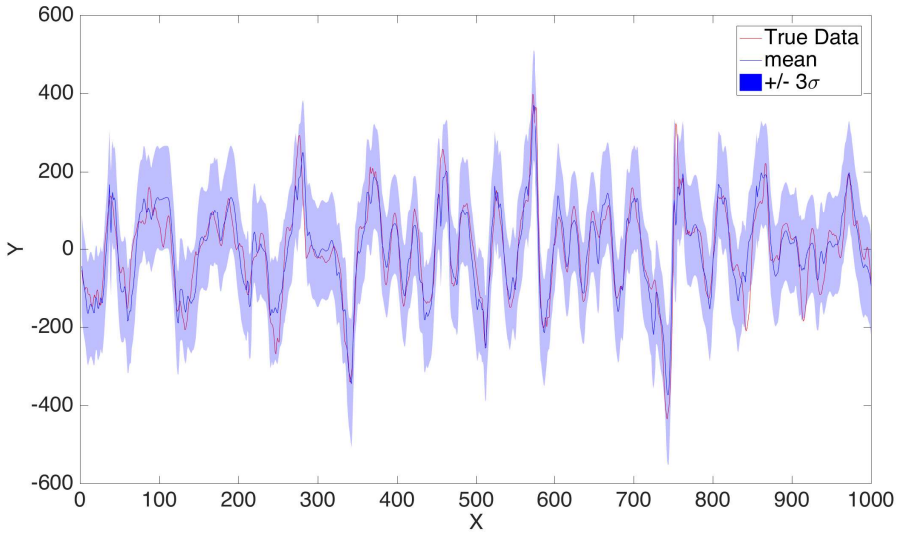


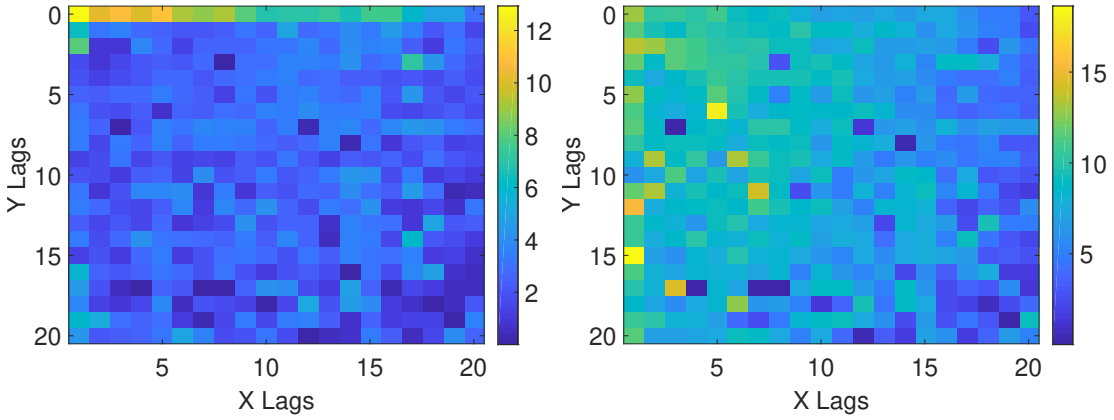
Figure 4.8: Prediction made by the best performing Gaussian Process on the test data, there is no difference between the OSA and MPO predictions since the model is static.

that consideration of only the NMSE performance of a probabilistic model in training can obscure the actual quality of the model.

Shown in Figure 4.8 is the prediction of the wave loading from the static GP. It can be seen that the uncertainty in many areas of the signal, where the predictions are worse, increases. The area of concern is around point 850 where the model is confident of its prediction but this is far from the measured signal (outside a 3σ interval). Although it is expected that a small number of points would lie outside this interval, the strong autocorrelation in the errors suggests a problem in the model. It is likely that this could be the result of the process being non-stationary in a manner which is not captured by the inputs to the model. For example, this change in force may be the result of some out of plane loading on the structure for which there is no information in the inputs.

4.2.1 Cross-validation Training of GP-NARX Models

As previously discussed the choice of cost function in an optimisation scheme can affect the results. Within a NARX setting it can be useful to move away from the usual negative log marginal likelihood cost function often used for optimisation of Gaussian process hyperparameters. Since, in the case of wave loading, it is desirable



(a) OSA (b) MPO

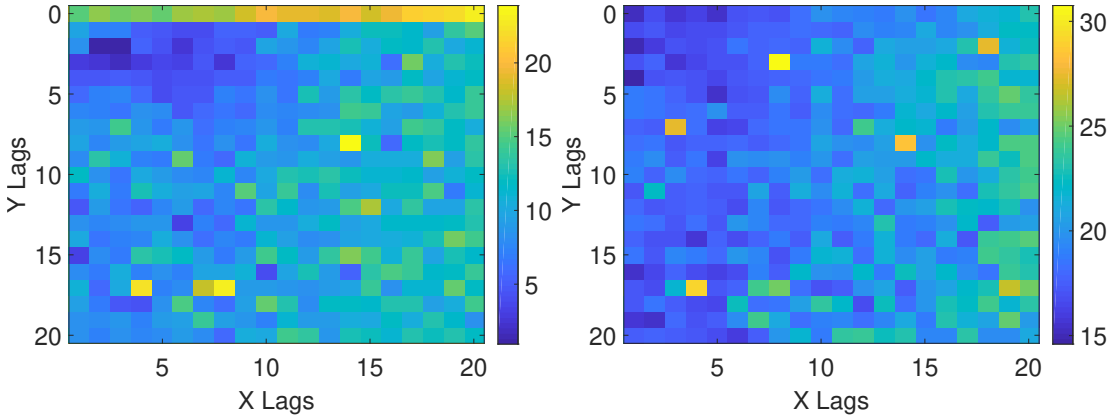
Figure 4.9: Heat maps of the normalised mean-squared error of the training data for the OSA and MPO predictions for the GPs trained via cross-validation.

to build models for full MPO outputs, the use of a cross-validation against this type of prediction is investigated. As discussed the NMSE fails to capture, as a cost function the full behaviour of the model. Therefore, the (negative log) posterior likelihood of the predictions on an independent validation set is used as the cost function for optimisation. As before, the QPSO optimiser is used in an effort to find the global minimum of this cost function.

The experiment is conducted as previously, the only change made being the choice of cost function. It should be noted that this approach adds the computational burden of prediction an MPO output on a validation set at each evaluation of the fitness of a particle in the swarm. In practice, the length of training is not tangibly different and since the optimisation of each Gaussian process could be computed in parallel on a distributed computing resource.

As before, it is possible to consider the normalised mean-squared error of the predictions on both the training set and the test set; this is shown in Figures 4.9 and 4.14. It is clear that the change in the cost function for the optimisation has had a profound effect on the resulting models. There is no longer the same plateau shape seen with the error tending to zero as the lags increase. Instead lower order models begin to perform better than those with higher numbers of lags. Again, however, the models which perform best in training are not the models which perform best on the test data.

Investigating the best performing model in terms of the NMSE on the MPO prediction



(a) OSA (b) MPO

Figure 4.10: Heat maps of the normalised mean-squared error of the test data for the OSA and MPO predictions for GPs trained via cross-validation.

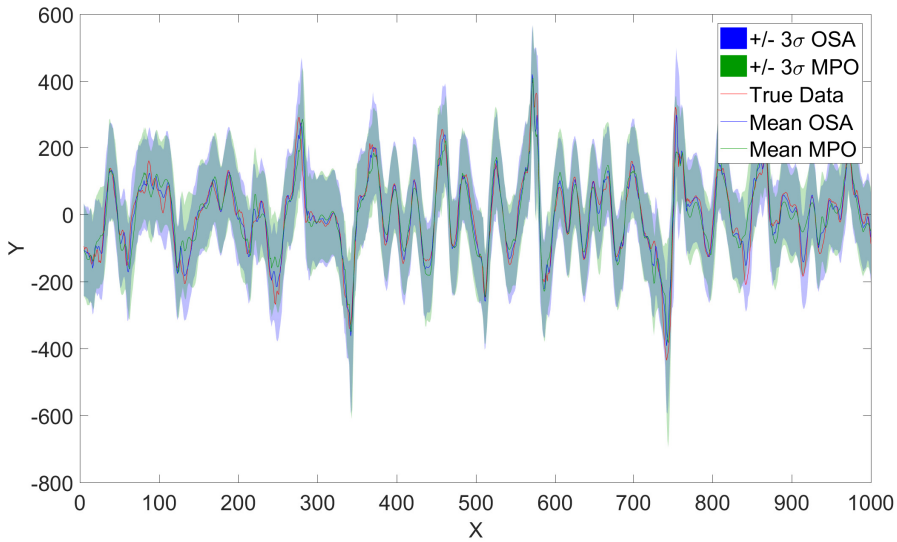
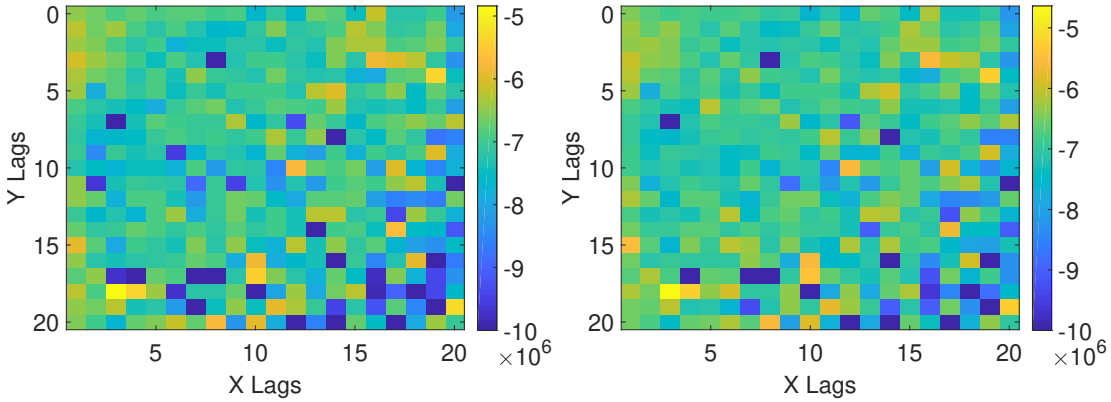


Figure 4.11: Plot of the mean and 3σ confidence intervals for both the OSA and MPO prediction on the test data for the lags which give the lowest NMSE score (14.60), this is taking $l_u = 1$, $l_y = 4$.

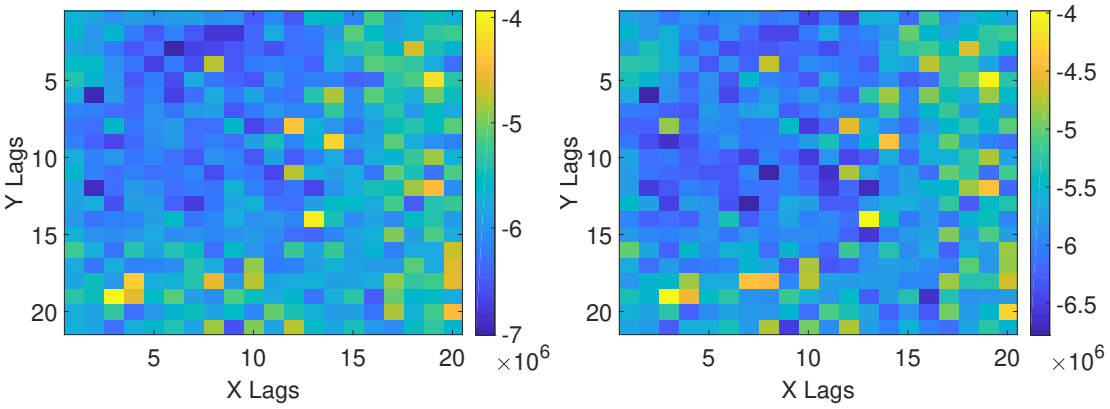


(a) OSA (b) MPO

Figure 4.12: Heat maps of the log posterior likelihood of the training data for the OSA and MPO predictions for the GPs trained via cross-validation.

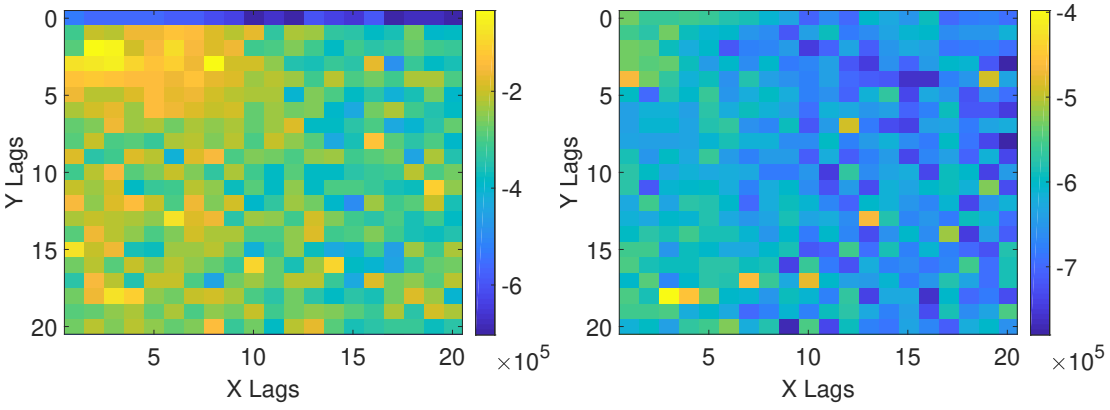
of the test data, the predictions by this model are shown in Figure 4.11. This model has $l_u = 1$ and $l_y = 4$, therefore, the best performing model is now dynamic when switching to the cross-validation cost function. Additionally, the NMSE score for this model is now 14.60 which is an improvement over the previous best model of 16.92 from the static GP learnt via the optimisation of the negative log marginal likelihood. In the same way as with the results shown for the learning with the negative log marginal likelihood, the NMSE does not tell the full story of the performance of the model. For instance, it is worrying that the models which have the lowest NMSE are different sets of lags for each of the training, validation, and test data. For this reason it is also important to consider the performance of the model in terms of the posterior likelihoods. It should be noted as well, that since the model is now trained to maximise the posterior likelihood of a validation set for MPO prediction, the OSA performance of the model decreases with respect to the previous results. This highlights how the choice of cost function in the optimisation can be tailored to the purpose of the model being learnt. It is seen, however, that the results on the training data and the test data are more similar in the case of the cross-validation training indicating that the model performance is better able to generalise.

Figures 4.12 to 4.14 show the posterior likelihoods of the models following cross-validated optimisation for combinations of lags on the training, validation, and test data respectively. It is seen that the posterior likelihood of the predictions is maximised in all MPO predictions for $l_u = 3$ and $l_y = 18$, interestingly this is not the model with the lowest NMSE. However, it is encouraging that this optimum



(a) OSA (b) MPO

Figure 4.13: Heat maps of the log posterior likelihood of the test data for the OSA and MPO predictions for GPs trained via cross-validation.



(a) OSA (b) MPO

Figure 4.14: Heat maps of the log posterior likelihood of the test data for the OSA and MPO predictions for GPs trained via cross-validation.

would remain the same for each of the datasets and it more strongly suggests the generalisation of the model behaviour. This would add confidence in the continued application of the model. This choice of lags also gives rise to the maximum posterior likelihood in the OSA predictions for both the training and validation data but not the test data. It is suspected that this is a result of training the model to perform MPO predictions rather than OSA, these highest likelihood models also correspond to the best performing models in an NMSE sense for OSA as well.

Figure 4.15 shows the predictions made on the test data for the lag combination that maximises the posterior likelihood across all three datasets for MPO predictions. Heuristically, this model looks similar in performance to the one seen in Figure 4.11,

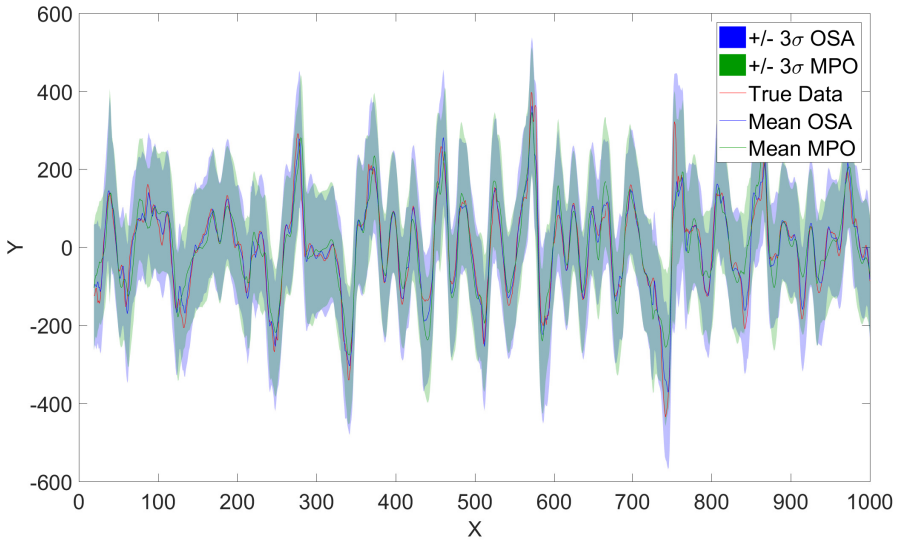


Figure 4.15: Plot of the mean and 3σ confidence intervals for both the OSA and MPO prediction on the test data for the lags which give the highest posterior likelihood score (for comparison this gives an NMSE of 17.60), this is taking $l_u = 3$, $l_y = 18$.

the NMSE for the MPO prediction in this case is 17.60 as opposed to 14.60, however, the increased posterior likelihood would indicate that this model better explains the data in a probabilistic sense. This model still outperforms the quality of the Morison equation see previously in both the prediction using OLS estimation of the parameters and the Gibbs sampler. Considering the likelihoods of the models the Morison equation learnt with the Gibbs sampler and the best likelihood from the black-box model trained with cross validation, the log posterior likelihood of the black-box model is -3.98×10^5 which is a significant improvement over the white-box model learnt via Gibbs sampling where this value is -1.15×10^9. It is not possible to compare the OLS solution in this manner since the model is deterministic.

Now, having chosen a dynamic model for black-box modelling of the wave loading, it is possible to demonstrate the effect of uncertainty propagation through the GP-NARX model when predicting wave loading. This is shown in Figure 4.16 where the possible realisations of the process are shown in blue compared to the measured signal shown in red. Here, it is seen that the variance in the model is increased as expected by the propagation of uncertainty. However, the model is stable enough that within the prediction range shown the model does not return to the prior. Taking the mean of these Monte-Carlo realisations and calculating the NMSE gives a score of 17.56 which is a marginal improvement over the MPO prediction. It is also

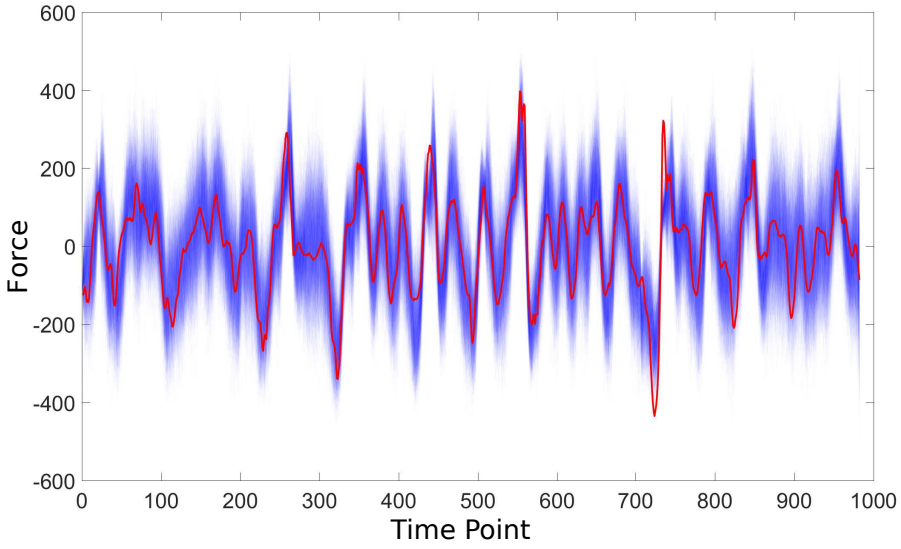


Figure 4.16: Monte-Carlo uncertainty propagation through the model with the best posterior likelihood, the measured signal is shown in red, Monte-Carlo realisations shown in blue.

seen that the uncertainty better captures the time history of the measured signal although there are still areas of concern, for example around point 725 the model fails to capture the measured signal well. This warrants further investigation but could be due to this loading level not being well covered in the training set. Despite this, the results of the cross-validation training of the model shows that the use of the posterior likelihood of an independent validation set can help improve the learning of GP-NARX models when looking for good MPO performance.

4.3 Discussion

This chapter has presented methods for quantifying the wave loading on an offshore structure using data from the Christchurch Bay Tower [167]. For comparison the use of the Morison equation has been shown with the parameters learnt as a Bayesian linear regression problem as well as the usual deterministic parameter identification. This allowed quantification of the uncertainty in the model, not only in terms of the learnt noise variance but also with relation to the parameter uncertainty. This was compared to data-driven approach where the application of Gaussian Process models was tested.

It was shown that approaching the Morison equation as a Bayesian linear regression problem has allowed rigour in handling of the uncertainty to be increased. This leads to predictions where the wave loads are able to be considered in a probabilistic manner which more easily can be included into a Risk & Reliability based framework — for example this model provides the ability to propagate possible force time histories could allow probabilistic rain-flow counting in a fatigue analysis and continued uncertainty propagation. The results also show that the noise variance of the process contributes a significant amount to total variance of the signal. It is clear that the noise process is not i.i.d. (from the autocorrelation in the residuals) Gaussian and this would motivate further investigation into extending the Morison model of wave loading. It was shown that approaching the Morison equation as a Bayesian linear regression problem has allowed more rigour in handling of the uncertainty to be increased. This leads to predictions where the wave loads are able to be considered in a probabilistic manner which more easily can be included into a Risk & Reliability based framework — for example the ability to propagate possible force time histories could allow probabilistic rain-flow counting in a fatigue analysis and continued uncertainty propagation. The results also show that the noise variance of the process contributes a significant amount to total variance of the signal. It is clear that the noise process is not i.i.d. (from the autocorrelation in the residuals) Gaussian and this would motivate further investigation into extending the Morison model of wave loading.

When adopting the purely data driven methodology, it was seen that moving to a Gaussian Process model improved the quality of the fit, this was seen heuristically and in an improvement in the NMSE. The investigation into the advantage of using a dynamic model in the form of th GP-NARX raised some interesting observations for this dataset. It was seen that when testing the model for its OSA prediction ability the GP-NARX model was able to predict with low error when including lagged outputs — this model is able to predict well on the independent test set. However, when looking for a model that generalises well to an independent test set for MPO type predictions the addition of lags in the output reduces the quality of the model.

This raises some interesting considerations if one is planning on using the GP-NARX model. First, as expected, the OSA performance of the model will far exceed the MPO performance. The introduction of the output lags into the model leads to difficulties in implementing the model and ensuring that it's capable of representing the physical process. It was also shown that consideration of only the NMSE of the

predictions can lead to poor model choices when using probabilistic models. This is of course limited to having been only shown on this data set but some general advice can be drawn from the results seen here. The GP-NARX model is a powerful framework for modelling dynamic systems but this is most useful in the OSA context — for example if using it for model predictive control. The GP-NARX will struggle when there is significant process noise since the propagation of uncertainty will — rightly so — return the predictions to the mean shortly after beginning prediction. This is coupled with the difficulty in finding an appropriate starting condition for the model, something that has been assumed known in this test.

It was also demonstrated that the model performance in the MPO sense can be improved by moving to the posterior likelihood of a validation set as the cost function for optimisation. It was shown here, as well, that the use of only NMSE for assessing model performance can obscure the probabilistic quality of the model. By considering the posterior likelihoods, more consistent models are obtained with the same lags being chosen for all three datasets; training, validation, and testing. This helps ensure confidence in the generalisation of the model. One problem, however, is that even when the uncertainty is propagated through the GP-NARX model it fails to capture well the measured signal. This may suggest that the process itself is in fact non-stationary or there are other variables affecting the process, for example, not modelled here is the wave direction or any current effects.

While some of the difficulties encountered, in using GP-NARX for the modelling of wave loading, could be overcome by tuning the model selection in the GP-NARX, it has been shown that kernel selection in high dimensional spaces is difficult and non-intuitive. The combined task of lag and kernel selection is not readily solved via optimisation without an infeasible amount of computing resource. This limits the applicability of the GP-NARX model in SHM to cases where an OSA prediction is suitable or cases where the process can be very well represented with low noise. Examples where the OSA prediction would be useful could be for fault detection by tracking the model error over time to see when the functional behaviour of a system changes. This type of methodology has already been used with linear AR models for example see [180]. However, when the aim is to make accurate predictions for extended periods of time a different approach to modelling must be used where information from other parts of the sensor system can be fed back to reduce uncertainty and to guide the process. A natural framework in which to do this — where AR and NARX are a subset — is state-space modelling, that will be

investigated in the following two chapters.

Chapter 5

Particle-Gibbs for Nonlinear System Identification

Highlights:

- *The novel use of particle Gibbs is shown for system identification problems in structural dynamics*
- *It is shown how ancestor sampling and particle rejuvenation can aid Particle Gibbs methods in structural dynamics*
- *The identification of a Duffing oscillator highlights the effectiveness of the method — especially in the presence of high noise*

The GP-NARX is one way in which it is possible to model dynamic systems, however, as has been seen there are a number of drawbacks. These include the difficulty in training, handling of uncertainty, and the specification of the model order. An alternative approach is to use a state-space philosophy. This chapter introduces the application of a rigorous Bayesian approach to identification of parameters in a nonlinear dynamic system in the state-space framework. To do this, the approach of Particle Gibbs is used and a number of additions to the algorithm are made to improve its performance. The effectiveness of this approach is demonstrated by the identification of a Duffing oscillator in the presence of high measurement noise. This is to the author's knowledge the first time the particle Gibbs framework has been

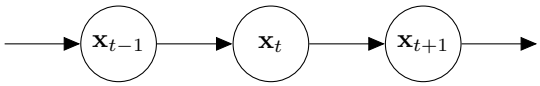


Figure 5.1: The graphical model of a Markov process where each state is only conditioned on the previous state in time forming a chain structure.

shown to be effective in a structural dynamics setting including the relatively recent developments in ancestor sampling and particle rejuvenation.

The SSM is more naturally suited to the task of modelling dynamic systems than a Gaussian Process, in fact these are the systems it was developed to model. The fundamental assumption is that a process can be modelled as some set of states $\mathbf{x}$ evolving through time for time $t = 0, \ldots, T$. These states are also assumed to obey the Markov property such that the probability distribution over the states at a point in time t conditioned on all previous time points $t = 0, \ldots, t-1$ is equivalent to the probability distribution conditioned only on the previous time point $t-1$, i.e.

$$p\left(\mathbf{x}_{t+1} \,|\, \mathbf{x}_{1:t}\right) = p\left(\mathbf{x}_{t+1} \,|\, \mathbf{x}_t\right) \tag{5.1}$$

Where the notation $\mathbf{x}_{1:t}$ indicates the sequence of states $\mathbf{x}_1, \ldots, \mathbf{x}_t$. This process forms a chain of random variables, each one only dependent upon the last. The graphical model for a Markov process is shown in Figure 5.1.

This is a very physically intuitive assumption, the systems it models can be thought of like knocking over a line of dominos. Suppose at some time t the interest is in if a particular domino in the line will be standing or have fallen. Although this is obviously dependent on whether the initial domino has been pushed over, the only information required to predict the state of the domino of interest is what the state of the preceding domino is at the time instant beforehand $t-1$. There are of course a large class of system which do not obey this assumption, however, many physical dynamic systems can be written in such a way that they do including all single and multi-degree of freedom linear oscillators and a large number of nonlinear systems.

The complete state of a system is rarely known, instead it is normally observed through some other mechanism or sensor and there are (at least in engineering problems) commonly some external influence on the behaviour of the system — traditionally this would be referred to as the control input but is just a general input for example forcing that is measured and not part of the system itself. The full

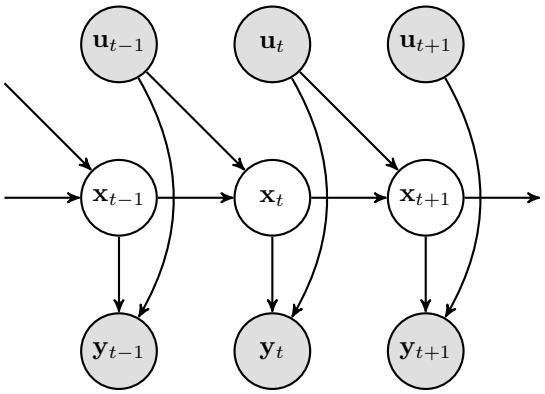


Figure 5.2: The graphical model of a general state-space model. The latent states $\mathbf{x}_{1:T}$ form a Markov process with the observation variables $\mathbf{y}_{1:T}$ conditioned only on the latent state at that time such that $p(\mathbf{y}_t \mid \mathbf{x}_{1:t}) = p(\mathbf{y}_t \mid \mathbf{x}_t)$.

state-space model accounts for this by introducing a set of observed variables $\mathbf{y}$ and a set of control inputs $\mathbf{u}$ at every time step.

The general SSM is shown in Figure 5.2, observed quantities are indicated by shaded nodes and hidden quantities by the unfilled nodes. This is a section of the model centred around time t but will extend backwards in time to $t = 0$ and forward to $t = T$. The model states that there is some function that transitions the latent states $\mathbf{x}$ from any time t to another time $t + 1$ which is a function of the current states $\mathbf{x}_t$ and the current inputs $\mathbf{u}_t$. Then the states at a time t are functionally related to the observed variables as some function of $\mathbf{x}_t$ and $\mathbf{u}_t$.

5.1 The Bayesian State-Space Model

The Bayesian approach to solving this model is to assume that the functional relationships in the SSM are in fact stochastic and can be modelled as probability densities. The first distribution to consider is the transition density $f_\theta(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_{t-1})$ which describes how the states evolve through time. The second fundamental part of the model specification is the observation density $g_\theta(\mathbf{y}_t \mid \mathbf{x}_t, \mathbf{u}_t)$ which models the relationship between the latent states $\mathbf{x}$ and the observed variables $\mathbf{y}$. The subscript notation to these distributions indicates their dependence on some set of parameters — this is often omitted but important here since the determination of these parameters will be a part of the end goal. It has already been shown how a specific function with a known noise model can be expressed as a probability density, specifically the

formulation of Bayesian linear regression is a simple example,

$$\mathbf{y} = X\mathbf{w}^{\mathsf{T}} + \varepsilon \quad \varepsilon \sim \mathcal{N}\left(0\,,\ \sigma_n^2\right) \quad \Leftrightarrow \quad p\left(\mathbf{y} \,|\, X, \mathbf{w}, \sigma_n^2\right) = \mathcal{N}\left(X\mathbf{w}^{\mathsf{T}}\,,\ \sigma_n^2\right) \tag{5.2}$$

The transition density and observation density fully describe an SSM if the initial conditions for the model are specified. This gives rise to a general form of the SSM,

$$\mathbf{x}_t \sim f_\theta\left(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}, \mathbf{u}_{t-1}\right) \tag{5.3a}$$

$$\mathbf{y}_t \sim g_\theta\left(\mathbf{y}_t \,|\, \mathbf{x}_t, \mathbf{u}_t\right) \tag{5.3b}$$

These equations form the basis of a very flexible model for time-series data, although SSMs can be applied to non-time-series data provided it is a valid Markov process. The state-space model is a popular tool and is widely adopted, a search of the Scopus database will return over 38,000 results with almost 35% of these being from engineering disciplines — good introductory texts on the subject can be found in [75, 181, 182].

When both the densities in Equation (5.3) are Gaussian with means described by a linear function the model is solvable in closed form. This is true for both the filtering density $p_\theta\left(\mathbf{x}_t \,|\, \mathbf{y}_{1:t}\right)$ and the smoothing density $p_\theta\left(\mathbf{x}_t \,|\, \mathbf{y}_{1:T}\right)$. These two densities are interesting to the modeller since they describe the distribution over the hidden states of the model at a point in time, given the sequence of observations up to the current time (in the case of the filtering density) or the total sequence of observations (in the case of the smoothing density). In general it is normally assumed that engineering systems can be adequately approximated to be linear. However, the majority of systems (if not all) will in fact be governed by some nonlinear phenomena. The SSM is flexible enough to describe these nonlinear systems, however, computing these two densities is no longer possible in closed form.

5.2 Methods for Nonlinear State Space Models

The continued use of the Linear Gaussian State-Space Model (LG-SSM) for modelling engineering systems is a testament to its flexibility, in fact this will be the basis of the inference performed in Chapter 6. However, as time progresses it is important to address the handling of nonlinear systems. In structural dynamics the use of lighter, more efficient structures, and the adoption of new nonlinear materials (such as composites) is forcing the explicit handling of nonlinearity.

The framework of the general state-space model does have the capacity for handling an arbitrary nonlinear system in either its transition or in its observation. This is provided the model can be written down in terms of its transition and observation densities which must be valid probability densities. Where it falls down is in actually performing inference in this type of model — the solutions to the required integrals are only available in closed form for the LG-SSM. That being the case, it is necessary to find approximate solutions to the system, this can be achieved in two main ways:

1. To modify the Kalman filter in such a way as it can be used for nonlinear problems via linearisation of the system
2. To approximate the behaviour of the nonlinear system directly without linearising

The differences between the two approaches are briefly highlighted here before discussing the later approach in more detail — specifically the application of Sequential Monte Carlo.

5.2.1 Modifications to the Kalman Filter

The two most popular modifications to the Kalman filter are the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF).

Extended Kalman Filtering

The EKF [182–184] is a model in which the noise on the process and the observation are assumed to be Gaussian and additive to some nonlinear transition and observation

functions. At every time step the filtering distributions are linearised using a first order Taylor series approximation of the nonlinear functions. This approach has become popular when estimating parameters in a model via the state augmentation approach, where the parameters of the model e.g. mass, stiffness, damping are assumed to be additional states evolving through time and related to the evolution of the physical system. This introduces a coupling between the states which introduces the nonlinearity to the model. The model is restrictive in that it is only usable in the case of Gaussian noise and when the nonlinear components of the model can be differentiated at least once. This requires the computation of the Jacobian matrices which may not be possible if the transition function is not differentiable [182].

The EKF is known to be ineffective when the nonlinearity in the model is not weak, that is that it cannot be well approximated (locally) by a linear model with minor correction. For the majority of systems it will result in poor performance even if it is possible to implement. However, if appropriate, it remains the simplest method in which a nonlinear model can be handled in terms of the filtering equations themselves which closely resemble those of the standard Kalman filter. The use of the EKF for parameter estimation via state augmentation can be misleading as there is no guarantee that the distributions over the states which represent the states are actually representative of the true parameter distributions. One reason for this is that these distributions are Gaussianised at every step. This may not cause too many difficulties especially if this assumption of weak nonlinearity and Gaussian distribution over the parameters holds true. However, there are instances where this is not the case and more rigorous handling of the nonlinearity in the system is necessary.

Unscented Kalman Filtering

The Unscented Kalman Filter [182, 185, 186] makes use of the unscented transform which is a method for Gaussianising a nonlinear transformation of a probability distribution by means of a set of *sigma points* — these deterministic points from the previous distribution are propagated through the nonlinear transition and used to estimate the mean and covariance of a Gaussian that best describes their distribution following the transformation. A set of sigma points, the number of which are determined by the dimensionality of the problem are each passed through the nonlinear function. The sigma points are chosen deterministically as the mean and points chosen by pre-set parameters α and κ which determine how spread

the points are around the mean of the distributions. The mean and covariance of the distribution following the nonlinear step can then be estimated from these sigma points. The UKF more closely resembles a Monte Carlo approach where the distribution is represented by number of point masses although rather than sampling those points it approximates the distribution on the basis of a fixed set of points. It is unclear in literature whether the UKF can be said to be absolutely better than the EKF, however, it is clear that in systems with strong nonlinearity the first order approximations in a standard EKF are not sufficient and the UKF will perform better [187–190]. The additional advantage of the UKF over an EKF approach is that there is not a requirement to be able to compute the Jacobian of the system. However, by moving to this sigma point method the procedure approaches a full MC approximation of the true nonlinear distribution which avoids the implicit linearisation of both the EKF and UKF. The most popular MC method is Sequential Monte Carlo which is introduced here.

5.3 Sequential Monte Carlo

In a Monte Carlo approach, distributions are approximated by a number of point masses; Sequential Monte Carlo (SMC) methods [103] are a subset of this. In a SMC setting an explicit evolving relationship between the distributions it exploited, i.e. a sequence of probability distributions through time are modelled. This could be achieved via sequential importance sampling where, at each point in time, each distribution is approximated by a number of point masses, each of which has a different importance weight where a higher weight corresponds to a higher likelihood at that point in the distribution. Originally SMC algorithms, namely the bootstrap particle filter [191], were developed for tackling nonlinear Bayesian filtering problems which appear in nonlinear SSMs.

A problem with a pure sequential importance sampling approach is sample degeneracy where particles quickly gain very low weights meaning they contribute little to understanding the mass of the Probability Density Function (PDF). If a particle moves to an area of low probability in the next distribution it is likely it will continue to move to areas of low probability. Therefore, it is not contributing much to understanding the true distribution of the probability mass. In an ideal situation each particle would contribute equally to the “map” of the probability distribution

this only happens when they have equal importance weights. The particle filter handles this problem by resampling particles in order to ensure that at the end of every step each particle has a weight equal to $1/N$.

A general nonlinear SSM can be considered where,

$$x_t \sim f_\theta \left(x_t \,|\, x_{t-1}, u_{t-1} \right) \tag{5.4a}$$

$$y_t \sim g_\theta \left(y_t \,|\, x_t, u_t \right) \tag{5.4b}$$

here x_t is a vector of some hidden (latent) states at time t, the evolution of which is governed by $f_\theta \left(x_t \,|\, x_{t-1}, u_{t-1} \right)$. u_t is a vector of 'control' inputs to the model at time t, in structural dynamics this would generally be the force input to the oscillator. There is a slight notation change for this section where the states are now lower case despite the them being vectors, a bold variable indicates that it is a set of particles for that state, e.g. $\mathbf{x}_t$ would indicate the set of particles representing the latent states at time t. These states are related to a vector of observed variables y_t through the probabilistic model defined by $g_\theta \left(y_t \,|\, x_t, u_t \right)$. In this formulation $f_\theta \left(x_t \,|\, x_{t-1}, u_{t-1} \right)$ is the transition density of the model and $g_\theta \left(y_t \,|\, x_t, u_t \right)$ the observation density of the model. Both of these distributions have their dependence on the unknown model parameters θ explicitly denoted for clarity. The first distribution of interest is the filtering distribution of an SSM, which is given by Bayes theorem,

$$p_\theta \left(\mathbf{x}_{1:t} \,|\, y_{1:t} \right) = \frac{g_\theta \left(y_t \,|\, x_t, u_t \right) p_\theta \left(x_t \,|\, y_{1:t-1} \right)}{p_\theta \left(y_t \,|\, y_{1:t} \right)} \tag{5.5a}$$

$$p_\theta \left(x_t \,|\, y_{1:t} m \right) = \int f_\theta \left(x_t \,|\, x_{t-1}, u_{t-1} \right) p_\theta \left(x_{t-1} \,|\, y_{1:t-1} \right) \mathrm{d}x_{t-1} \tag{5.5b}$$

By restricting the forms of $f_\theta \left(x_t \,|\, x_{t-1}, u_{t-1} \right)$ and $g_\theta \left(y_t \,|\, x_t, u_t \right)$ it is possible to obtain closed form solutions to these equations which are the well known Kalman filter formulation [192]. However, the restrictions in the Kalman filter model — linear dynamics and observation equations, and Gaussian noise — are too restrictive for many systems encountered; this includes all structural systems with nonlinearity.

The solution which allows the use of more flexible nonlinear models in SMC is to use sequential importance sampling to approximate the filtering distribution. The

filtering density Equation (5.5b) is then approximated by,

$$p_\theta\left(x_t \,|\, y_{1:t}\right) \approx \frac{g_\theta\left(y_t \,|\, x_t, u_t\right)}{p_\theta\left(y_t \,|\, y_{1:t-1}\right)} \sum_{i=1}^{N} w_{t-1}^i f_\theta\left(x_t \,|\, x_{t-1}, u_{t-1}\right) \tag{5.6}$$

where, the superscript notation is used for indexing, for example w_t^i denotes the importance weight of the i^{th} particle at time t. The notation adopted here is to use indexing similar to the software package Matlab where a subscript $1:T$ indicates the section of the indices of a vector from 1 to T inclusively. For matrices the same notation is adopted but with commas separating dimensions and following column major order. This is again approximated using importance sampling such that the (unnormalised) importance weights of the filtering density are given by,

$$\tilde{w}_t^i = \frac{g_\theta\left(y_t \,|\, x_t^i, u_t\right) \sum_{i=1}^{N} w_{t-1}^i f_\theta\left(x_t^i \,|\, x_{t-1}^i, u_{t-1}\right)}{\sum_{j=1}^{N} \nu_{t-1}^j q_\theta\left(x_t^i \,|\, x_{t-1}^j, y_t, u_t\right)} \tag{5.7}$$

it remains for the proposal density $q_\theta\left(x_t^i \,|\, x_{t-1}^j, y_t\right)$ and proposal weights ν_{t-1}^j to be chosen. In the simplest application — a bootstrap particle filter — it is set such that,

$$\sum_{j=1}^{N} \nu_{t-1}^j q_\theta\left(x_t^i \,|\, x_{t-1}^j, y_t, u_t\right) = \sum_{i=1}^{N} w_{t-1}^i f_\theta\left(x_t^i \,|\, x_{t-1}^i, u_{t-1}\right) \tag{5.8}$$

i.e. the proposal weight is set to be the previous particle weight and the proposal density is chosen to be the transition density. In this case, the unnormalised importance weights are given by $\tilde{w}_t^i = g_\theta\left(y_t \,|\, x_t^i, u_t\right)$, as the proposal cancels out the other term in Equation (5.7). The procedure for running a bootstrap particle filter is shown in Algorithm 3.

In Algorithm 3 the resampling step is shown as drawing the particle ancestors from a multinomial defined by the normalised weights. It is possible to form more efficient sampling strategies that are equally valid. In practice, it is usual to use a systematic resampling scheme where the empirical Cumulative Density Function (CDF) is partitioned into N sections and a single random number determines the location within that section from which each particle is drawn, for a more thorough review see [193].

Algorithm 3 Bootstrap Particle Filter

Initialisation:
  $i \leftarrow \{1, \ldots, N\}$ ▷ For N particles
  $x_1^i \sim p_\theta(x_1)$
  $\tilde{w}_1^i = g_\theta(y_1 \mid x_1^i, u_1)$
  $w_1^i = \frac{\tilde{w}_1^i}{\sum_{j=1}^N \tilde{w}_1^j}$ ▷ Normalisation
For $t = 2, \ldots, T$:
  Resampling: $a_t^i \sim \mathcal{MN}(\mathbf{w}_t)$ ▷ Sample from Multinomial
  Propagation: $x_t^i \sim f_\theta\left(x_t^i \mid x_{t-1}^{a_t^i}, u_{t-1}\right)$
  Weighting:
    $\tilde{w}_t^i = g_\theta(y_t \mid x_t^i, u_t)$
    $w_t^i = \frac{\tilde{w}_t^i}{\sum_{j=1}^N \tilde{w}_t^j}$

It has been shown that, in certain circumstances, better proposal distributions can be used through the introduction of auxiliary variables [194, 195] giving the Auxiliary Particle Filter (APF). If the process and observation noise are Gaussian with linear observation equations, optimal proposals can be made in the form of Kalman filter updates, this is referred to as the *fully-adapted particle filter*. In this case the proposal weights ν_t^j for $j = 1, \ldots, N$ are set to $p_\theta\left(y_t \mid x_{t-1}^j\right)$ and the proposal distribution $q_\theta\left(x_t^j \mid x_{t-1}^j, y_t\right)$ is set to $p_\theta\left(x_t^j \mid x_{t-1}^j, y_t\right)$

This is possible if the model can be written as,

$$p_\theta\left(x_t^j \mid x_{t-1}^j\right) = \mathcal{N}\left(f_\theta\left(x_{t-1}^j, u_{t-1}\right),\ Q\right) \tag{5.9a}$$

$$p_\theta\left(y_t \mid x_{t-1}^j\right) = \mathcal{N}\left(C\, x_t^j + D\, u_t,\ R\right) \tag{5.9b}$$

Then $p_\theta\left(x_t^j \mid x_{t-1}^j, y_t\right)$ has the same form as Equation (A.6) and $p_\theta\left(y_t \mid x_{t-1}^j\right)$ has the same form as Equation (A.5). This procedure greatly increases the efficiency (reduces the variance) of the particle filter.

In order to write down a general algorithm for SMC filtering, it is useful to collapse some of the steps in the filter down notationally to form a general SMC algorithm where a system of particles x_t^i (for $i = 1, \ldots, N$ particles) are propagated through time with their ancestors a_t^i (the particle index from which this particle transitioned) by a proposal kernel $M_{\theta_t}(a_t, x_t)$. These particles are then assessed through a weighting

function $W_{\theta,t}(\mathbf{x}_{1:t})$, which calculates the normalised weights of a set of particles x_t^i. In this way, if an APF is used the change to the algorithm is encompassed in a new proposal kernel $M_{\theta_t}(a_t, x_t)$ and change to the weighting function $W_{\theta,t}(\mathbf{x}_{1:t})$.

Algorithm 4 General Sequential Monte Carlo

1: **Initialisation**:
2: $i \leftarrow \{1, \ldots, N\}$ ▷ For N particles
3: $x_1^i \sim p_\theta(x_1)$
4: $w_1^i = W_{\theta,t}(\mathbf{x}_1)$
5: **For** $t = 2, \ldots, T$:
6: $\{a_t^i, x_t^i\} \sim M_{\theta_t}(\mathbf{a}_t, \mathbf{x}_t)$
7: $\mathbf{x}_{1:t}^i = \left\{x_{1:t-1}^{a_t^i}, x_t^i\right\}$
8: $w_t^i = W_{\theta,t}(\mathbf{x_{1:t}})$

In Algorithm 4 the additional step of recording the paths of each particle has been included in line 7. Here, each ancestral path $x_{1:t}^i$ is updated by concatenating the current particle position x_t^i with the path of the ancestor particle $x_{1:t-1}^{a_t^i}$. Here, a_t^i is shorthand for the ancestor of particle i at time t since all the ancestors are recorded in the vector a_t at time t.. This records the trajectory of that particle through time when tracing back through its ancestors. Although only a bookkeeping step, this will be crucial when it comes to forming an effective Particle Gibbs (PG) algorithm. The ancestral paths of every particle $i = 1, \ldots, N$ for time $t = 1, \ldots, t$ are represented by the bold notation $\mathbf{x_{1:t}}$, likewise the weights of every particle at time t is represented by $\mathbf{w}$, the vector of weights.

5.4 Particle Gibbs

To make use of SMC within an MCMC scheme such as particle Gibbs, it is necessary to make a slight modification to Algorithm 4 which ensures it is a valid Markov kernel [196]. This will be referred to here as the Conditional Particle Filter (CPF). In the CPF one of the particle trajectories is held constant as a reference trajectory $x'_{1:t}$. By convention, this is usually the N^{th} particle in the particle system. At every time step this particle is propagated forward in time as usual, however, the value corresponding to x'_t is not updated in the resampling step. This has the effect of 'guiding' each run of the SMC through the state-space [197] — this 'guiding' ensures the ergodicity of the Markov Chain.

The PG algorithm, a subset of Particle MCMC methods, can be thought of conceptually as a Gibbs sampler for an SSM, where samples are drawn iteratively from the CPF, for the state trajectories conditioned on the parameters, and then from the conditional distributions of the parameters given the states. For a more thorough introduction, along with proofs of PG as a valid Markov kernel, the reader is directed to Andrieu *et al.* [196].

5.4.1 Ancestor Sampling

A simple yet powerful modification to the PG algorithm was proposed by Lindsten *et al.* [197] which they termed Particle Gibbs with Ancestor Sampling (PG-AS). In this construction, rather than fixing the ancestors of the reference trajectory $a_{1:t}^N$ to be N at every time step, the ancestor for x_T^N is resampled at each time step t. The ancestor of the particle is the index of the particle at the previous time step which has been propagated to the current particle.

This change helps to tackle the path degeneracy problem encountered in PG, where all particles will share a single common ancestor if looking far enough back in time. In PG, path degeneracy leads to the state values close to $t = 1$ not being resampled very often — i.e. there is poor mixing in the Markov chain. This will lead to slow convergence of the model. To achieve this the ancestors of the model are sampled such that,

$$\mathbb{P}\left(a_t^N = i\right) \propto \tilde{w}_{t-1|T}^i = w_{t-1}^i \, p_\theta\left(x_t' \,|\, x_{t-1}^i, y_{1:t-1}\right) \tag{5.10}$$

where $p_\theta\left(x_t' \,|\, x_{t-1}^i, y_{1:t-1}\right)$ is the likelihood of the reference particle given the dynamics of all the particles $i = 1, \ldots, N$ at the previous time step — note this includes the previous point in the reference trajectory. Conceptually, the ancestor is sampled based on which is the most likely parent for the reference trajectory at time t. Applying Bayes rule this is proportional to the prior for that particle w_{t-1}^i multiplied by the likelihood that the reference x_t' was drawn from the transition density for each possible ancestor $p_\theta\left(x_t' \,|\, x_{t-1}^i, y_{1:t-1}\right)$.

This methodology has been shown to be effective in a number of system identification tasks, e.g. [134, 198], however, it has been shown that if the model is nearly degenerate or degenerate then the benefit of ancestor sampling is greatly diminished. Ancestor

sampling relies on sampling from the *backward kernel* of the SSM.

$$p\left(\mathbf{x}_t \mid \mathbf{x}_{t+1}, y_{1:t}\right) \propto f\left(\mathbf{x}_{t+1} \mid \mathbf{x}_t\right) p\left(\mathbf{x}_t \mid y_{1:t}\right) \tag{5.11}$$

This quantity can be difficult to assess in certain models including those which have a degenerate structure and in other cases it can be that this quantity does not exist in closed form [199].

5.4.2 Particle Rejuvenation

In a degenerate model all of the probability mass can be centred on only the ancestor relating to the reference trajectory. If this is the case only this ancestor will be sampled, i.e. $a_t^N = N$ for every time step t, and the algorithm returns to the standard PG formulation without ancestor sampling. In this situation the issues with path degeneracy causing poor mixing return and a very high number of particles are needed to produce a valid result. Since the model for a structural dynamic system is nearly degenerate, i.e. there is no process noise on the first state relating to the displacement of the oscillator, mixing in the Markov Chain can still be poor even when employing the ancestor sampling technique — this leads to very slow convergence to the stationary distributions in the MCMC approximation.

Lindsten *et al.* [199] propose a solution to this based on a modification to the target distribution of the Gibbs sampler which they term *particle rejuvenation*. By also resampling a part of the reference trajectory with the ancestors at each time step the degeneracy in the model can be avoided as the reference is *loosened up*. To introduce PG-AS with particle rejuvenation it is necessary to develop some additional notation, $\tilde{x}'_{t:T}$ is the future reference trajectory and Ξ is some subset of the future reference trajectory $\Xi \in \tilde{x}'_{t:T}$.

To cope with degeneracy in the model, the Gibbs sampler is partially collapsed over a subset of future state variables Ξ such that $\Xi = \{\mathbf{x}_t, \ldots, \mathbf{x}_{\kappa_t}\}$ with $\kappa_t = \min\{T, t + \ell - 1\}$. Since the goal is to resample both the ancestor a_t^N and part of the future reference trajectory Ξ it is necessary to sample from the joint PDF of (a_t, Ξ_t) where,

$$p\left(a_t, \Xi_t\right) \propto w_{t-1}^{a_t} f_\theta\left(\mathbf{x}'_{\kappa_t+1} \,|\, \mathbf{x}_{\kappa_t}\right) \left\{ \prod_{s=t+1}^{\kappa_t} f_\theta\left(\mathbf{x}_s \,|\, \mathbf{x}_{s-1}\right) g\left(\mathbf{y}_s \,|\, \mathbf{x}_s\right) \right\} f\left(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}^{a_t}\right) g\left(\mathbf{y}_t \,|\, \mathbf{x}_t\right) \tag{5.12}$$

In general it will not be possible to sample from this PDF in closed form, therefore, a Markov kernel is chosen which generates valid samples from this PDF. The choice of this kernel will be problem dependent, but a sensible choice within an SMC framework is to employ an importance sampling approach. A conditional importance sampling scheme is established where the unnormalised importance weights are given by Equation (5.12). If ℓ is chosen to be one then Equation (5.12) simplifies to,

$$p\left(a_t, \Xi_t \,|\, \ell = 1\right) \propto w_{t-1}^{a_t} f_\theta\left(x'_{t+1} \,|\, \mathbf{x}_t\right) f\left(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}^{a_t}\right) g\left(\mathbf{y}_t \,|\, \mathbf{x}_t\right) \tag{5.13}$$

The complete procedure for PG-AS with particle rejuvenation is show in Algorithm 5, with a Markov kernel K_t to sample the new ancestor and Ξ_t.

Algorithm 5 Particle Gibbs with Ancestor Sampling and Particle Rejuvenation

Initialisation:
  Simulate $\Xi_1^\star \sim K_1\left(\Xi'_1, \cdot\right)$
  Update $x'_{1:\kappa_t} \leftarrow \Xi_1^\star$
  Set $x_1^N \leftarrow x'_1$
  $x_1^i \sim p_\theta\left(\mathbf{x}_1\right)$ for $i = 1, \ldots, N-1$
  $w_1^i = W_{\theta,t}\left(\mathbf{x}_1\right)$ for $i = 1, \ldots, N$
For $t = 2, \ldots, T$:
  $\left\{a_t^i, x_t^i\right\} \sim M_{\theta_t}\left(a_t, \mathbf{x}_t\right)$ for $i = 1, \ldots, N-1$
  Simulate $\left(a_t^N, \Xi_t^\star\right) \sim K_1\left(\left(N, \Xi'_t\right), \cdot\right)$
  Update $x'_{t:t+\kappa_t} \leftarrow \Xi_t^\star$
  $x_{1:t}^i = \left\{ x_{1:t-1}^{a_t^i}, x_t^i \right\}$ for $i = 1, \ldots, N$
  $w_t^i = W_{\theta,t}\left(\mathbf{x}_{\mathbf{1:t}}\right)$ for $i = 1, \ldots, N$
Sample $k \sim \mathcal{MN}\left(\mathbf{w}_T\right)$
return $\mathbf{x}'_{1:T} = \mathbf{x}_{1:T}^k$

For further details on PG-AS with particle rejuvenation the reader is directed to Lindsten *et al.* [199]. It is clear from the algorithm that the methodology is very similar to that of the general SMC scheme presented in Algorithm 4. The value returned from the PG step (here the term is used to cover all methods that fall under this methodology) is a sample of the ancestral path for a particle, this is a sample of

a path from the conditional smoothing distribution $p(x_{1:T} \mid y_{1:t}, \theta)$. To achieve this it is necessary to use a conditional particle filter to ensure validity of the Markov chain and the process of ancestor sampling and particle rejuvenation is used to help better mixing — i.e. more independent samples of the state trajectories are drawn.

PG-AS with particle rejuvenation allows efficient Gibbs sampling from the smoothing distribution of a nonlinear SSM. To perform inference it is necessary to utilise this technique as part of a blocked Gibbs sampler [32, 33] where samples are drawn for the state trajectory conditioned on the parameters and then for the parameters conditioned on the state trajectory [196]. The full procedure is shown in Algorithm 6, for notational convenience X' is used to represent the full sampled reference trajectory $x'_{1:T}$.

Algorithm 6 Blocked Gibbs Sampler for Inference in SSMs

1: Set X'_0 and θ_0
2: Set S as number of steps
3: **for** $s = 1, \ldots, S$ **do**
4: Sample $X'_s|\theta$ as in Algorithm 5
5: Sample $\theta_s \sim p(\theta \mid X'_s)$
6: **end for**
7: Discard first s_b samples as burn-in

5.5 Identification of a Duffing Oscillator

The Duffing oscillator has been a benchmark system within the structural dynamics community for a number of years and has received a number of different treatments. Part of the reason for its continued popularity is that despite the simplicity of the differential equation it gives rise to many interesting phenomena, including chaos. It has been possible to create systems which show Duffing like behaviour both mechanically [200, 201] and electronically, most notably in the *silverbox* benchmark dataset [202, 203]. The *silverbox* data has been studied extensively within the nonlinear system identification community [204–208].

A comprehensive introduction to be behaviour of the Duffing oscillator can be found in [209] where many of its main characteristics are detailed. Within engineering it has also been considered a benchmark for system identification [210]. It has even received a Bayesian treatment in [211], although in that work the problem of estimating

the latent states of the model is not considered only the parameter estimations, additionally the identification was carried out with a modest noise level, 5% RMS. The system has also been identified via an optimisation approach [212] where the parameters are estimated in a deterministic setting.

This work applies, for the first time, a particle Gibbs identification methodology to a Duffing oscillator, in order to demonstrate its applicability in structural dynamics. To do this it is necessary to make use of both ancestor sampling and particle rejuvenation.

There are four parameters to identify in the Duffing oscillator,

$$m\ddot{y} + c\dot{y} + ky + k_3 y^3 = F \tag{5.14}$$

the mass m, stiffness k, damping c, and cubic stiffness k_3. The system is simulated using a fifth order Runge-Kutta formulation [213], see Appendix B. One challenge in implementing a state-space approach to the identification of systems such as the Duffing oscillator is converting the continuous time ordinary differential equation (ODE) into a discrete time SSM for which the methods are developed. However, this can be solved in the same manner as the time-step integration used for simulation of nonlinear systems, as the procedure is to produce a model for $\mathbf{x}_{t+1}$ given $\mathbf{x}_t$ where the state vector $\mathbf{x}_t = [y\ \dot{y}]^\mathsf{T}$ as is common in application of numerical techniques to second order ODEs. Therefore, if the same fifth order Runge-Kutta scheme is used, the state transition density can be written down, assuming a Gaussian noise with unknown covariance across the states, as,

$$f_\theta\left(\mathbf{x}_{t+1} \,|\, \mathbf{x}_t\right) = \mathcal{N}\left(r(\mathbf{x}_t)\,,\ Q\right) \tag{5.15}$$

with $r\left(\mathbf{x}_t\right)$ being the equation for the nonlinear propagation of the states which is achieved with the 5^{th} order Runge-Kutta scheme detailed in Appendix B and Q being the discrete time process noise covariance — when the process noise is assumed to be Gaussian as it is here, it is also assumed to be small since it is known that the model form is exactly correct.

5.5.1 Modelling of the Duffing Oscillator using SMC

Before demonstrating the identification of the Duffing oscillator using PG methods, the system is modelled assuming the parameters of the oscillator are known. This serves to demonstrate the effectiveness of SMC approaches to modelling nonlinear dynamical systems. For both the examples shown here it is assumed that the displacement of the oscillator can be measured directly but corrupted with some Gaussian noise which has a covariance R, since the dimension of the observation y is one, R is in fact just the variance of the noise on the observation.

The Duffing oscillator is simulated with a sample frequency of 65536 Hz for 500 time steps. The parameters of the oscillator are set as: $m = 0.1$ kg, $k = 9.8696 \times 10^7$ $\mathrm{N\,m^{-1}}$, $c = 100\pi$ $\mathrm{N\,s\,m^{-1}}$, and $k_3 = 1000000$ $\mathrm{N\,m^{-1}}$. The simulated data has Gaussian white noise added to it at 50% of the root-mean-squared (RMS) level of the signal, this corresponds to a signal to noise ratio of 6 dB. The signal being considered in shown in Figure 5.3 with the noise free signal shown in blue and the noisy signal shown in red.

The observation likelihood $g_\theta\left(y_t \,|\, x_t, u_t\right)$ is given by,

$$g_\theta\left(y_t \,|\, x_t, u_t\right) = \mathcal{N}\left(\begin{bmatrix}1 & 0\end{bmatrix}\mathbf{x}_t\,,\ R\right) \tag{5.16}$$

with R set to the known noise level. The simulated data is shown in Figure 5.3.

The identification of the oscillator using PG-AS will now be demonstrated. Considering Algorithm 6 it is necessary to specify the procedure for sampling $\theta_m\,[c/m, k/m, k_3/m, 1/m]$. This can be done in the same manner as setting up a Gibbs sampler for a Bayesian Linear Regression problem when the sampled state $X'_s|\theta$ is known and contains a sample of the displacement and the velocity at each time step from the smoothing distribution. This is possible because the equation is linear in the parameters, which can be seen by writing the design matrix D,

$$D = \begin{bmatrix}\mathbf{y} & \dot{\mathbf{y}} & \mathbf{y}^3 & \mathbf{F}\end{bmatrix} \tag{5.17}$$

This design matrix is formed as a basis function expansion of the sampled state trajectory in the Gibbs sampler X'_s such that,

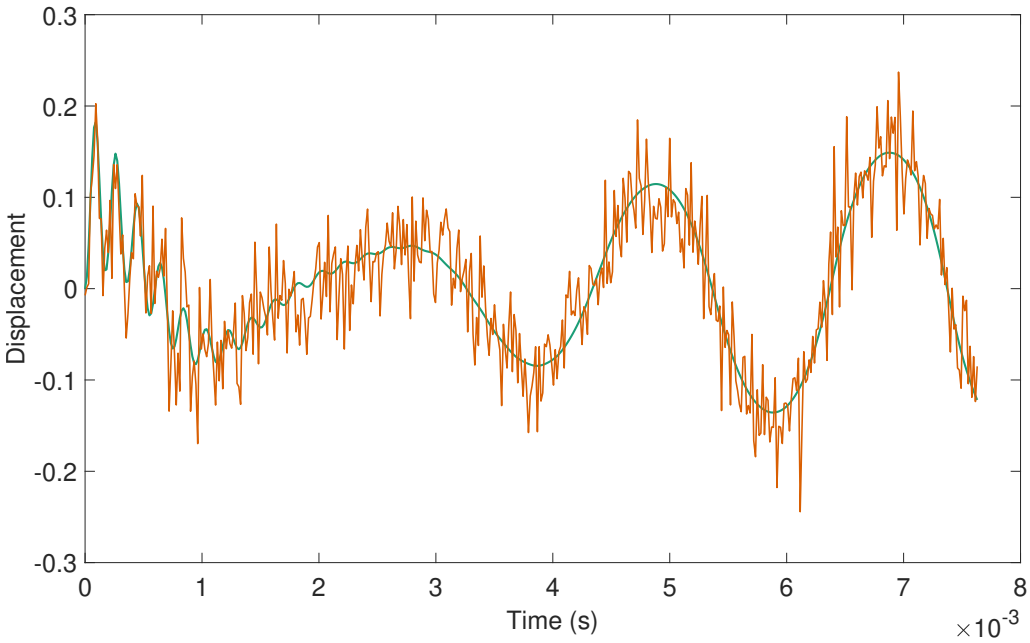


Figure 5.3: Displacement observed from the simulated Duffing oscillator with the noise free simulation shown in green overlaid with the noisy data used in the experiments in orange.

$$D = \Phi\left(X'_s\right) = \begin{bmatrix} X'_{s,1} & X'_{s,2} & \left[X'_{s,1}\right]^3 & F_{1:T} \end{bmatrix} \tag{5.18}$$

Where $X'_{s,c}$ is the c^{th} column of the matrix X'_s which corresponds to the c^{th} state of $\mathbf{x}_{1:T}$. Then the model can be written as,

$$\ddot{\mathbf{y}} = \beta D^{\mathsf{T}} + \varepsilon \qquad \varepsilon \sim \mathcal{N}\left(0\,,\ 1/\tau\right) \tag{5.19}$$

the noise is parametrised in terms of its noise precision τ which is the reciprocal of the variance. The noise here is not simply the noise related to the process noise of the model but also the first order approximation in the Gibbs sampler. Within the Gibbs sampler the parameters of this linear model will not be the known true values but rather samples from the parameters distribution $\theta_m^{(s)}$. In order to ensure the model remains linear so that the Gibbs sampler can be applied a first order approximation of the continuous process must be used here. The model being identified, therefore, is,

$$y_{t+1} = y_t + \Delta\left(\theta_m^{(s)} D_t\right) + \varepsilon \tag{5.20}$$

with Δ being the time step. This model remains linear in the parameters by setting $\hat{D} = \Delta D$ and setting the target $t = y_{t+1} - y_t$. It is now possible to place conjugate priors over all of the parameters in the model,

$$p\left(\theta_{m,p}\right) = \mathcal{N}\left(\mu_p\,,\ 1/\tau_p\right) \tag{5.21}$$

indexed by $p = 1, \ldots, 4$. The conjugate prior for the noise precision τ is a Gamma distribution which is parameterised in terms of it shape a and rate b [33], this is explicitly shown here to avoid confusion.

$$p\left(\tau\right) = \mathcal{G}a\left(a, b\right) = \frac{b^a}{\Gamma\left(a\right)}\tau^{a-1}\exp\left\{-b\tau\right\} \tag{5.22}$$

with $\mathcal{G}a\left(\cdot\right)$ being the Gamma function. The process noise covariance Q is known to be close to zero everywhere, it must be set to some fixed known small value a *priori* to ensure stability in the model. The hyperparameters of the prior are set to be $a = 1$ and $b = 500$.

To sample from the posterior $p\left(\tau \,|\, \theta_m^{(s)}, X_s'\right)$, it is necessary to compute the residual between the sampled states and those predicted by the linear in the parameters model.

$$\hat{A} = \begin{bmatrix}\theta_{m,1}^{(s)} & \theta_{m,2}^{(s)} & \theta_{m,3}^{(s)} & \theta_{m,4}^{(s)}\end{bmatrix} \tag{5.23}$$

$$\hat{X}_s' = X_{s,2:T}' - D\hat{A}^{\mathsf{T}} \tag{5.24}$$

The posterior $p\left(\tau \,|\, \theta_m^{(s)}, X_s', y_{1:t}\right)$ can then be written down as,

$$p\left(\tau \,|\, \theta_m^{(s)}, X_s'\right) = \mathcal{G}a\left(a + \frac{T-1}{2}, b + \left[\hat{X}_s'\right]^{\mathsf{T}}\left[\hat{X}_s'\right]\right) \tag{5.25}$$

It is now possible to write down the conditional posteriors over each of the parameters in θ_m using the notation $(\cdot)_{:,-p}$ to indicate choosing all columns except the p^{th} from

a matrix

$$r = X'_{s,2:T} - D_{:,-p}\hat{A}^{\mathsf{T}} \tag{5.26a}$$

$$\hat{\sigma} = \frac{1}{\tau_p + \tau D_{:,p}^{\mathsf{T}} D_{:,p}} \tag{5.26b}$$

$$\hat{\mu} = \hat{\sigma}\left(\tau_p \mu_p + \tau\left(r^{\mathsf{T}} D_{:,p}\right)\right) \tag{5.26c}$$

$$p\left(\theta_p \,|\, \tau, \theta_{-p}, X'_s\right) = \mathcal{N}\left(\hat{\mu}\,,\ \hat{\sigma}\right) \tag{5.26d}$$

In this way, a blocked Gibbs sampler has been constructed. This allows samples of the state trajectories X'_s to be drawn from the smoothing distribution, by means of a conditional particle filter. Crucially, samples for the parameters θ_m can also be drawn from their respective posteriors using Equation (5.26).

All of the components are now available to identify the Duffing oscillator using a PG-AS scheme with particle rejuvenation. The method is tested on a simulated dataset from a Duffing oscillator with 50% RMS noise added, 6dB signal to noise ratio. Gaussian prior distributions are placed over the parameters of the model with the means perturbed by a Gaussian random number with zero mean and a variance equal to 50% of the true parameter value and the prior variance is set to twice the absolute value of the parameters. The initial reference trajectory for the CPF is set to be zero at every time step. The number of particles used is 50 and the prior distribution over the states is set to be zero mean with a variance of 0.01 for each state. These choices will be discussed in more detail later.

he PG-AS scheme is run with a burn in of 250 steps and before 14750 samples are drawn of both the state trajectories X'_s and the parameters θ_s which include the process noise covariance for $s = 1, \ldots, 14750$. For this test the process noise covariance matrix Q is set to be

$$Q = \begin{bmatrix} \epsilon & 0 \\ 0 & 1 \times 10^{-8} \end{bmatrix}$$

where ϵ is machine precision. The samples of the state trajectories X'_s are samples from the smoothing distributions over the latent states of the model — the displacement

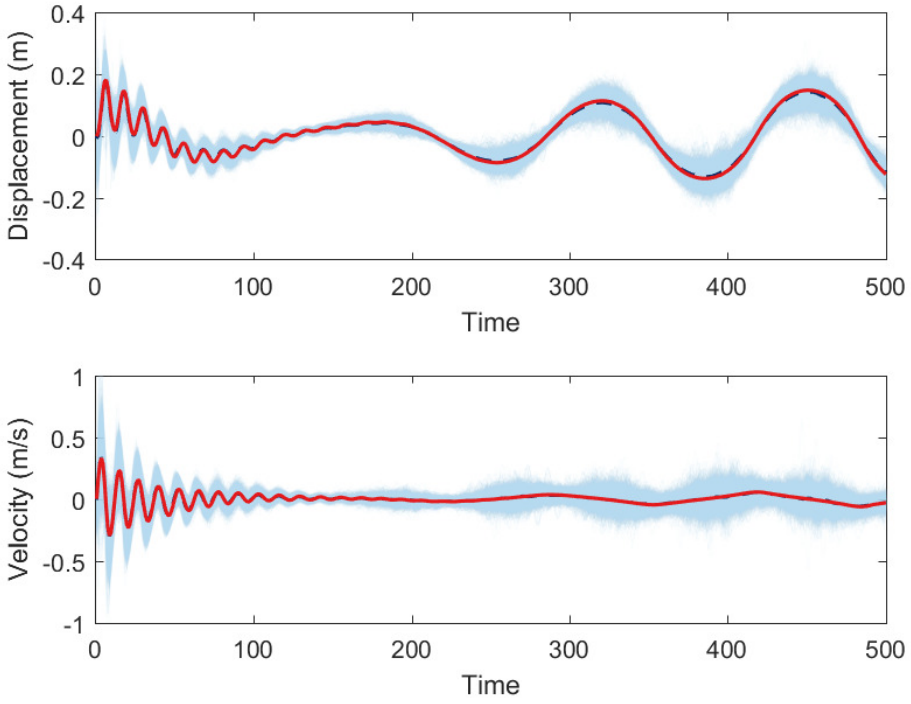


Figure 5.4: Figure showing the smoothing distributions over the displacement and the velocity of the Duffing oscillator estimated by the PG-AS algorithm. The path of the individual sample paths is shown in light blue, the mean of the particles is shown in dashed black and the true noise free path is shown in red.

and velocity of the oscillator. All of the samples are plotted in Figure 5.4 in light blue. The mean of these samples is taken at every time step and this is plotted in blue against the true noise free state of the oscillator. The fit of the model can be seen to be very good with NMSEs of 0.2397 and 0.4874 in the displacement and velocities respectively. It is clear also that the true state trajectories lie well within the probability mass of the densities approximated by the samples. If considering only the measured signal this gives a NMSE of 25.2836 with the true displacement, the problem comes in this high noise case it is difficult to estimate the velocity required for Gibbs sampling the parameters of the model. A naïve estimation of the velocity state via first differences would give an NMSE of 4.0598×10^{11} in this case! This identification of the latent states of the model (here the velocity) is the most powerful aspect of the state-space approach as this increase leads to more robust parameter estimations. This highlights the strength of this novel application of particle Gibbs within structural dynamics.

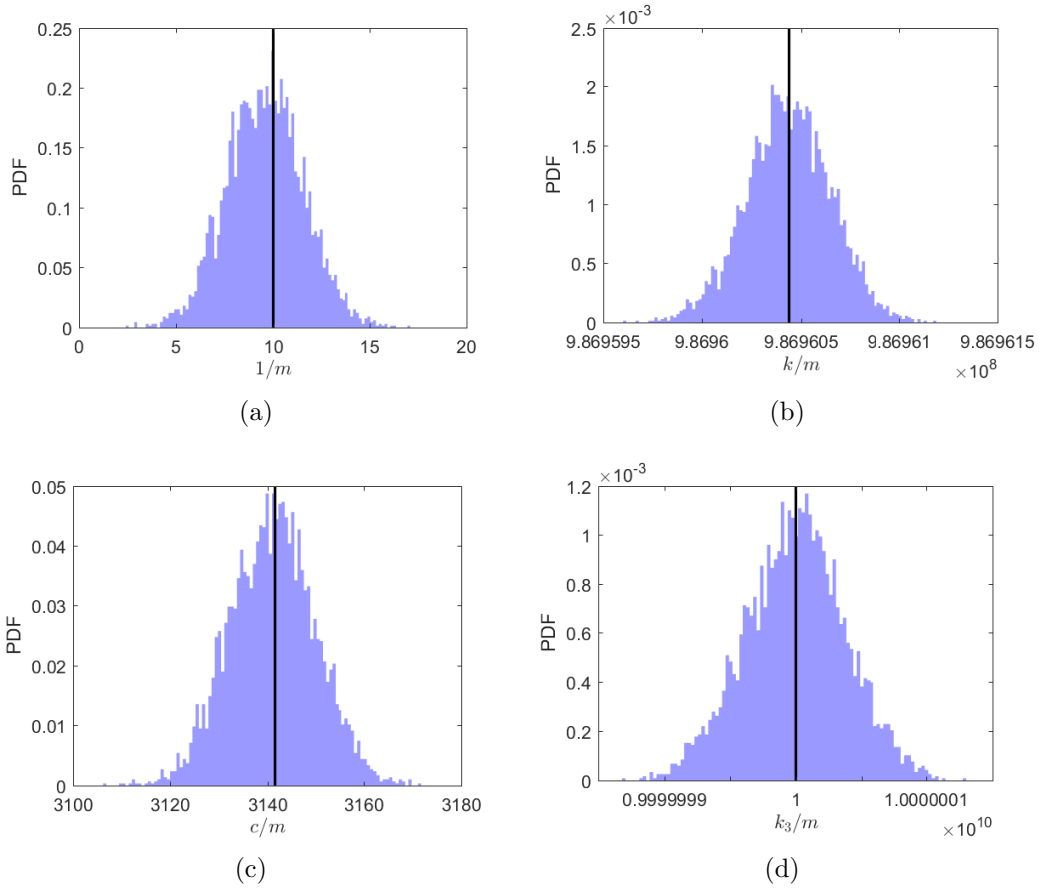


Figure 5.5: Figure showing histograms of the samples of the couple parameters $1/m, k/m, c/m, k_3/m$ from the Gibbs sampler after burn-in using the PG-AS algorithm. The known ground truth is shown in black.

Alongside the estimates of the smoothing distributions, PG-AS also returns distributions over the parameters. The distributions over the parameters of the model θ_m are shown in Figure 5.5. A histogram the samples from the Gibbs sampler after burn-in with the known ground truth marked using the solid black line. The distributions over the the parameters $1/m$, k/m, and k_3/m have the ground truth value well within the probability mass and close to the means. The absolute percentage errors in the *maximum a posteriori* estimates of the coupled parameters is less that 0.02% for all parameters except the $1/m$ term which is underestimated by 4.20%. It is likely that this is due to the first order approximation in the Gibbs sampler.

Decoupling the parameters it is seen that the results are also encouraging and follow directly from those seen for the transformed parameters. The discrepancy in the $1/m$ parameter, however, does now cause a shift in the other parameters of the model since they must be multiplied by m to recover the individual parameters. It could

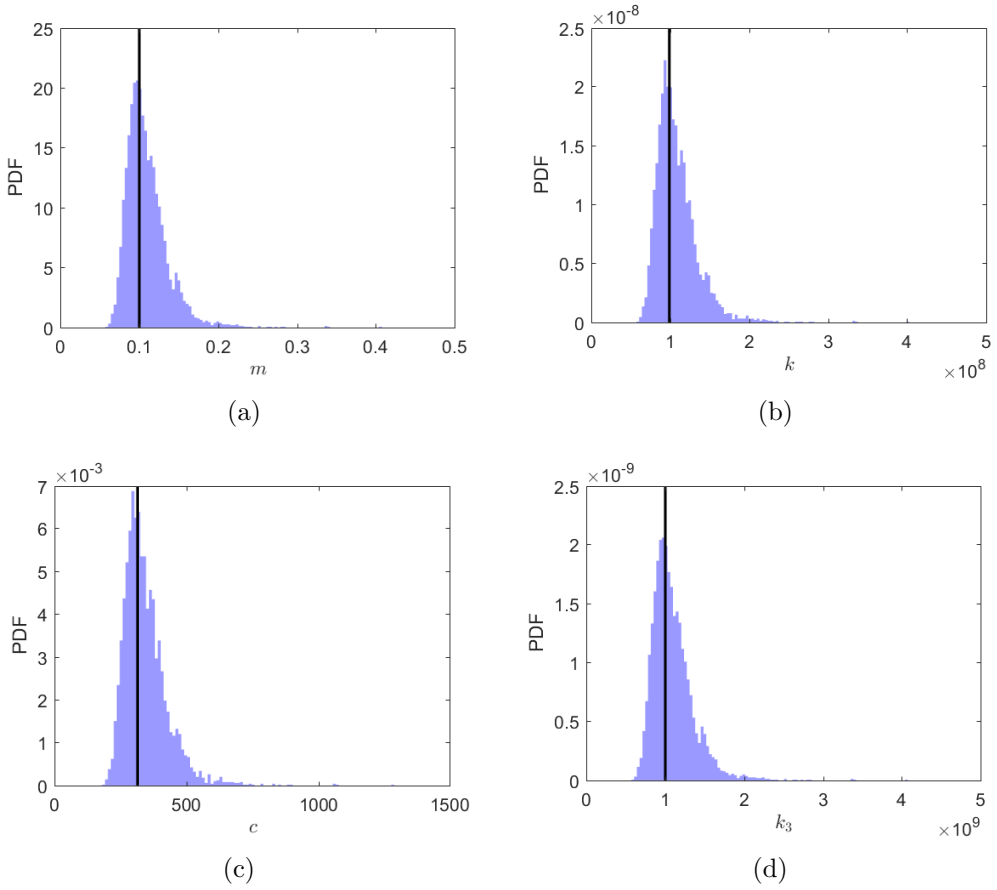


Figure 5.6: Figure showing histograms of the samples of the individual system parameters m, k, c, k_3 from the Gibbs sampler after burn-in using the PG-AS algorithm. The known ground truth is shown in black.

be that moving to a higher order approximation for sampling the parameters would help this, alternatively it may be useful to formulate the problem in terms of modal parameters.

5.5.2 The Role of Particle Rejuvenation

The use of particle rejuvenation in the model helps to allow good mixing in the Markov chain when the model is nearly degenerate, i.e. the process noise covariance is nearly singular. This is the case in this test since the model is known to have very low or zero process noise. To allow the computation of the filter it was not possible to set the process noise on the first state to be equal to zero; instead it was set to the machine precision of the computer on which the experiment was conducted.

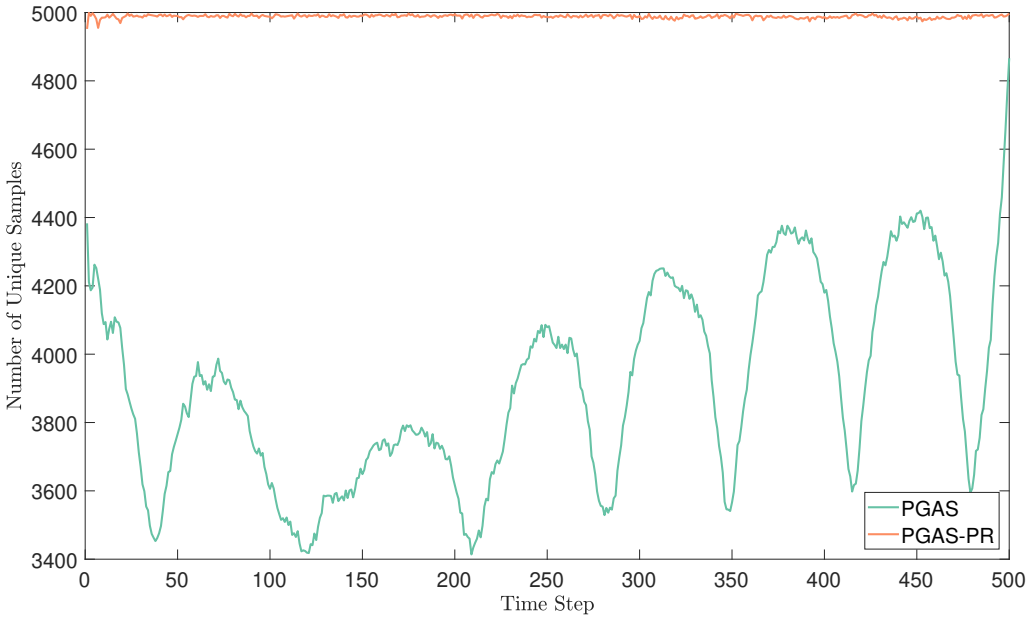


Figure 5.7: Comparison of path degeneracy with and without particle rejuvenation for N=40 particles with the process noise set to be nearly degenerate.

To investigate the role of the particle rejuvenation scheme, the Markov chain is run for 5000 iterations and for comparison the burn-in period is not discarded in this case. For this test to increase the computation speed the sample rate was set to 32768. All other variables were set to be the same as previously, including the driving force signal, except where noted.

Figure 5.7 shows a comparison of the path degeneracy problem for the standard PG-AS formulation and when particle rejuvenation is used. N, the number of particles, is set to forty for both cases. The effect of particle rejuvenation is clearly shown to allow better mixing in the Markov chain. The robustness of the PG-AS with particle rejuvenation to reducing the number of particles is shown in Figure 5.8 where the model is run with varying numbers of particles N. It is also worth noting that the standard PG-AS formulation was unstable when running with the machine precision as the process noise. To alleviate this numerical problem the process noise had to be increased to 1×10^{-12} which ensured stability in the calculation. This is another reason why a scheme using particle rejuvenation would be preferable when a model exhibits this nearly degenerate structure.

Figure 5.8 shows the resampling of the states in the Gibbs sampler when using the PG-AS algorithm with Particle Rejuvenation in the degenerate model described

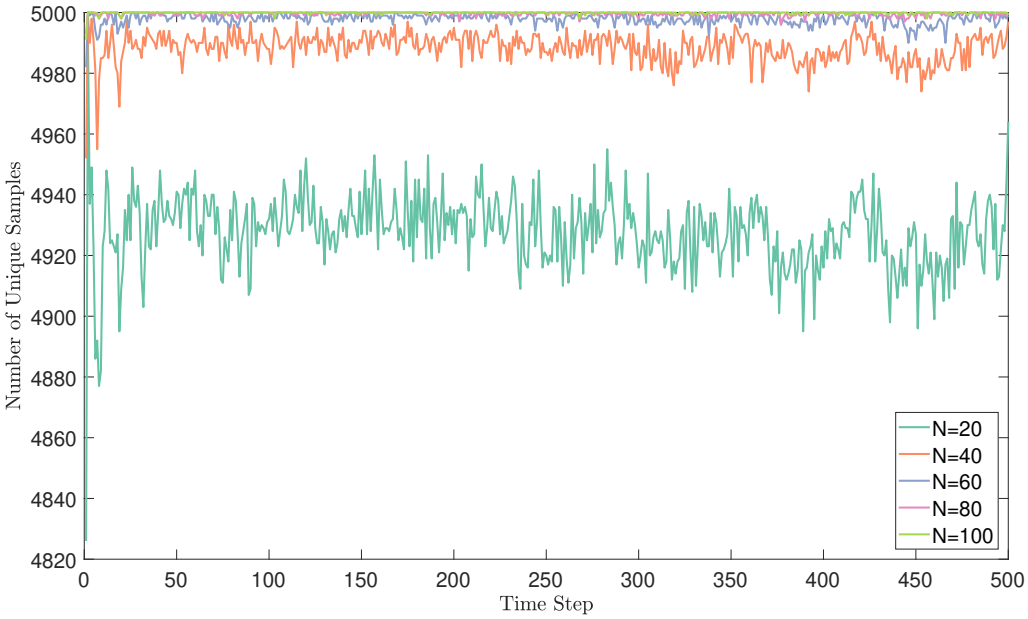


Figure 5.8: The effect of varying the number of particles on the path degeneracy problem, in terms of the number of unique samples at each point in time of the state vector. The number of particles is increased from 20 to 100 in steps of 20 to show the effect of increasing the number of particles on the path degeneracy problem.

above. Even with a very small number of particles (20) the resampling in the PG-AS model with particle rejuvenation is very good. When increasing N to 40 it is seen that each step in the state vector is resampled more than 4950 times, which is more than 99% of the time. This will ensure good mixing in the Markov chain and make for a more efficient inference algorithm. This also justifies the use of particle rejuvenation since the increase in computational expense for the rejuvenation step is outweighed by the ability of the algorithm to run with a greatly reduced number of particles. It was also observed that this particle rejuvenation step increased the stability of the algorithm when running with a low number of particles. The use of particle rejuvenation directly impacts the speed of the algorithm which scales linearly with the number of particles (although the addition of the the Particle Rejuvenation step approximately doubles the computation required [199]).

5.6 Discussion

It has been shown in this chapter the ability of Sequential Monte Carlo methods to accurately model nonlinear dynamical systems in by means of sequential importance

sampling. This state-space approach has been combined with a Markov-Chain Monte Carlo scheme, Particle Gibbs with Ancestor Sampling, which has allowed joint inference over the latent states of a nonlinear system — the Duffing oscillator — and the parameters of that model. It has been shown that the structure of this type of system is nearly degenerate, a trait shared by many physical nonlinear systems. To allow more robust and efficient inference in this setting the use of a particle rejuvenation scheme has been shown to be effective. This Bayesian approach to nonlinear system identification is capable of recovering the posterior distributions over the states and the parameters of the model even in the presence of strong nonlinearities.

This work has shown a novel methodology for identification of a Duffing oscillator as an example of a nonlinear system encountered within structural dynamics. The strength of this type of approach is to allow recovery of probability distributions over not just the parameters but also the latent, noise free, states of the model — here, the displacement and velocity. This allows the utilisation of this type of model in situations where there is signification measurement noise. This ability to handle high measurement noise situations makes the use of this type of methodology appealing for application on 'real world' data where it is often difficult to remove noise sources in the same manner as in a laboratory setting. The flexibility of the model also allows the use of non-Gaussian noise models if the noise is of a known form, this again may be a valuable tool when applying this type of model on a full scale structure.

There is, however, still development to be done. It would be beneficial to investigate further if the bias in the mass parameter is linked to the first order approximation in the model. If so it will require the adoption of an alternative strategy for sampling the parameters, this could possibly be achieved with a Metropolis-within-Gibbs [33] type approach. One of the key drawbacks of the current model is the requirement to know the model form *a priori*. For the particle filter to be computed currently the model is a parametric model of the nonlinearity for which physical insight is required. It is desirable, therefore, to investigate the combination of this model will a semi-parametric or non-parametric form in which the nonlinearity itself could be learnt. This would require investigation of a model similar to that seen in [214], the challenge being to incorporate the nonlinearity into a feedback path as is the case in the Duffing oscillator.

The other key problem faced in the offshore industry is the difficulty in acquiring measurements of the inputs to a system. This was discussed as motivating the work

in Chapter 4. It turns out that this problem can also be tackled within the state-space framework by treating the forces as latent states of a partially observed system — additionally, the parameters of the model can be learnt whilst inferring the forcing on the system. This is considered in the next chapter where a Gaussian process prior is used for modelling the unknown inputs within a state-space framework. Although, not presented there, the combination of the work of this chapter with that methodology could open up interesting avenues for output-only identification of nonlinear structural dynamical systems provided the model form is known.

Chapter 6

State Space Models for Coupled Load-Parameter Identification

Highlights:

- *The Gaussian Process Latent Force Model is introduced for the problem of load estimation in structural dynamics*
- *It is shown that this can be efficiently rewritten as a linear state-space model*
- *This is used to perform joint input-state-parameter estimation in an operational modal analysis setting*

It has been seen that the modelling of wave loading in isolation from the rest of the structure is challenging. The use of a GP-NARX model to do this in Chapter 4 highlighted a number of difficulties. The primary concern is that in a NARX type model there is no mechanism to reduce uncertainty once it has appeared unless more, direct observations of the process are made. If a user were planning on implementing the GP-NARX this would present a problem. The reason for modelling the wave loads is that it is hard and expensive to obtain direct readings, and merely increasing the gaps between these might not be good enough. It is desirable, therefore to make use of indirect measurements related to the system, e.g. the acceleration of a platform at the top-side. This situation is ideally suited for the SSM and in fact the astute

reader will notice that the NARX model is a particular form of state-space model.

The benefit of the GP-NARX is in the flexibility of the GP to represent arbitrary nonlinear relationships between the variables. The complete extension of this into the SSM framework is to move to the GP SSM [132–134]. It turns out that this is not necessary in some situations and it is possible to harness the flexibility of the GP within the state-space framework without generating a nonlinear state-space model. If considering a linear dynamical system, then only the evolution of the forcing in time is nonlinear, this can be modelled using a temporal GP (a Gaussian process with time as its inputs) and placed within an Latent Force Model (LFM) where the force is considered as a latent function (i.e. it is not observed directly).

Considering the equation of motion for a linear system in structural dynamics, its behaviour is fully characterised by that equation. There are three key components to in the model:

1. The parameters of the system — the mass, stiffness, and damping matrices; M, K, and C
2. The inputs to the system, the forcing at each degree of freedom — F
3. The outputs of the system — displacement, velocity, and acceleration; $\mathbf{y}$, $\dot{\mathbf{y}}$, and $\ddot{\mathbf{y}}$

Knowing any two of these allows the calculation of the missing component. For instance if the inputs and outputs are known then the parameters of the system can be determined, this is usually done through modal analysis; or if the parameters and inputs are known, the outputs can be simulated. The process is complicated by the presence of noise, although this can normally be overcome with sufficient measurements. The other issue is related to model form error, in extreme cases this would be the breaking down the assumption of linearity in the model.

It is unfortunate, therefore, that in most cases there is (in addition to the measurement noise) unknowns in both the parameters and the input forcing. Indeed, in the case of an offshore platform an SHM system will usually monitor accelerations from which the outputs of the system can be determined with relatively high certainty. The parameters of the system will be known but only at some nominal value and it is normally useful to attempt to update these to more closely reflect the physical

system. Additionally, as previously discussed it is very difficult to directly measure the inputs to the system — the forcing on the platform.

The current solution to this is to use techniques in OMA, or output-only modal analysis. Reviews of some techniques which may be used can be found in [38, 160], however, a common problem encountered is the identification of modal behaviour when the loading has significant harmonic components which are close to the modes of the structure and when the structure is lightly damped. In this case it becomes very difficult to separate the narrow peaks in the frequency domain due to the harmonic loads and those from the resonances of the structure.

The majority of these techniques rely on an assumption that the structure is subject to a Gaussian white noise inputs. If not, assumptions can be made:

1. That the dominant frequency of the forcing is well separated from any of the resonant frequencies of the structure
2. That the forcing will appear as a peak with far lower damping than the modal responses or with negative damping

The problem of input estimation is not new in engineering. Lourens *et al.* [215] present a method based on Kalman filtering of a reduced order model similar to the approach shown here but do not simultaneously perform parameter inference. Similarly Ma *et al.* [216] show results for estimation using a Kalman filter on a lumped mass model of a beam with known parameters. Azam *et al.* [217] again apply a Bayesian filter based system to model an output only system without estimating the parameters of the model; this work is extended in [218]. Ching *et al.* [219] demonstrate the use of Bayesian filtering (comparing a particle filter and extended Kalman filter are compared) for parameter and state estimation when the inputs to the system are known. This problem is also tackled in Gillijns and De Moor [220]. However, the task of joint input-state-parameter estimation remains challenging and is still an emerging area of investigation [221].

An alternative approach to those above is to attempt to model this input in a Bayesian manner by placing a prior over it and inferring its posterior distribution. A natural way to approach this problem is to assume that the forcing function can be modelled as a Gaussian process in time, i.e. the covariance function calculates the covariance in time — $k(t, t')$. This approach was first explored by Alvarez *et al.* [222] who develop

a model which very closely resembles the input identification problem for a linear system, although they did not tackle this specific task — instead focussing on human motion. They name this the Latent Force Model LFM. Subsequent discussions were presented in [223] addressing some of the computational issues with reference to sparse implementations [127, 224].

It has become commonplace to use the Extended Kalman Filter (EKF) formulation where the parameters of the model are included as additional states, which is shown in [225] and discussed in [219]. In this way recursive estimates of the system parameters can be made within the Bayesian filtering setting [226]. Lourens *et. al* [215] show a Kalman filter based formulation for estimation of the states of the structure and the forcing signal based on fixed parameter sets from a reduced order finite element (FE) model. The issues with this use of an Extended Kalman Filter in this way have been discussed in Section 5.2.1.

In order to tackle the challenge of OMA, this work presents a novel use of the LFM [222] in a state space formulation [227]. This model is ideally suited to this task and by construction resembles the problem of joint parameter-state-input estimation present in OMA but as yet (to the author's knowledge) has not been applied to this setting. The use of the LFM in its state-space formulation allows efficient inference with a nonparametric prior over the forcing function. Alongside this, Bayesian estimates of the system properties can be obtained using MCMC inference over the state space model, this avoids the linearisation present in the EKF formulation for parameter estimates. Being a Bayesian model of the full system, this methodology will converge to the true distributions over the parameters and also the forcing function which will give indication of the uncertainty in both the inputs and also the parameters. This can be carried forward into further uncertainty quantification analysis or used for heuristic assessment of model quality.

A key aspect of this model is the use of the state-space representation of the LFM. It is necessary, therefore, to present the procedure for converting a temporal Gaussian process into its Kalman filter-smoother representation [228]. This allows the LFM to be written as an linear Gaussian state-space model which can be solved with the Kalman filtering and Rauch-Tung-Striebel (RTS) smoothing equations.

6.1 Continuous-Discrete LGSSMs

The linear Gaussian state-space model, as presented in Appendix A, is shown for a discrete time model. However, most physical systems are defined in terms of their continuous time properties, i.e. the differential equations governing dynamic systems. It would be possible to make some approximation of the continuous time behaviour via a time-step integration as was done when numerically solving the Duffing oscillator in the previous chapter. For linear systems, however, it is possible to solve the continuous system for the time interval of a given time-step Δ if the input is assumed constant [181]. The procedure for this conversion and also the handling of the process noise when moving from the continuous definition, as the spectral density of a white noise process, to the discrete-time definition in terms of the process noise covariance is shown now.

The notation, for this section, is $\mathbf{x}(t)$ is used to represent the state vector, and $\mathbf{y}(t)$ the observations in the model for the continuous case or $\mathbf{x}_t$ and $\mathbf{y}_t$ for the discrete time case. $\mathbf{u}(t)$ and its discrete time counterpart $\mathbf{u}_t$ is a known input to the system at time t. It is normally most convenient to form a continuous time state-space model since differential equations readily convert into this form. The continuous time state-space transition is defined as,

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = F\mathbf{x}(t) + G\mathbf{u}(t) + Lw(t). \tag{6.1}$$

Here, F is the continuous state transition matrix, G the continuous input matrix, and L the continuous time process noise matrix when $w(t)$ is a white noise process with spectral density q. The discrete model form required for solving the system with the Kalman filter and RTS smoother equations is,

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t + Q, \tag{6.2a}$$

$$\mathbf{y}_t = C\mathbf{x}_t + R, \tag{6.2b}$$

with A being the discrete time transition and B the discrete time input. Q is the process noise, C the observation matrix and R the observation noise. Since the solutions to models of this form are in discrete time it is necessary to convert a

continuous time transition model into its corresponding discrete time representation, i.e. find A and B from F and G. This procedure is well documented, for example see [181], and is shown here for completeness.

$$A = \exp_m (F\,\Delta t)\,, \tag{6.3a}$$

$$B = (A - \mathbb{I})\,F^{-1}G. \tag{6.3b}$$

$\exp_m$ is the matrix exponential operator, efficient approximations of which are readily available in most numerical software packages. Δt is the sampling period when discretising the continuous time model. The continuous time noise process $Lw(t)$ must also be discretised to perform inference over the model.

$$Q = \int_0^{\Delta t} \Phi\,(\Delta t - \tau)\,CqC^{\mathsf{T}}\Phi\,(\Delta t - \tau)\,\mathrm{d}\tau, \tag{6.4}$$

$\Phi(\tau) = \exp_m (F\tau)$. Once discretised, it is now possible to recover the filtering distributions $p\,(\mathbf{x}_t \mid \mathbf{y}_{1:t})$ and the smoothing distributions $p\,(\mathbf{x}_t \mid \mathbf{y}_{1:T})$, for a time point t in a data record that is T time points long, using the Kalman filtering equations [192] and the Rauch-Tung-Striebel (RTS) smoother [229] or see Appendix A.

6.2 The Latent Force Approach

The Latent Force Model (LFM) [222, 223] is an approach for combining GPs with a mechanistic model where the GP is used to represent the forcing experienced by the system. In this way, the output of the GP can include modelling of the dynamics of a system. This model is a form of *grey-box* where known physical processes are augmented with machine learning technology to improve model performance. Of most interest, from the point of view of structural dynamics, is a second order LFM which has the familiar form of a R degree of freedom linear dynamical system.

$$M\ddot{\mathbf{x}} + C\dot{\mathbf{x}} + K\mathbf{x} = F. \tag{6.5}$$

In the LFM, F — the forcing on every degree of freedom (in fact this also allows for forcing to enter in on the velocity terms if desired but this is rarely the case in structural dynamics) — is defined as a set of independent Gaussian Processes in time. The PDF of F is given by,

$$\begin{aligned} p\left(F \mid \mathbf{t}\right) &= \prod_{r=1}^{R} \mathbf{f}_r = \prod_{r=1}^{R} \mathcal{N}\left(\mathbf{0},\ K\left(\mathbf{t}_r, \mathbf{t}_r\right)\right), \\ \mathbf{f}_r &\sim \mathcal{GP}\left(\mathbf{0}, k(t,t')\right), \end{aligned} \tag{6.6}$$

where the covariance of each of the latent forces in time is defined by the covariance function $k(t,t')$ giving rise to the covariance matrix $K_{\mathbf{f}_r,\mathbf{f}_r}$ which is the covariance between the latent forces at all points in time. For each degree of freedom r, this is $K\left(\mathbf{t}_r, \mathbf{t}_r'\right)$ with $\mathbf{t}_r$ being the vector of all time points when the output is observed. While this model has produced some very promising results in modelling [222, 223], the main challenge in implementing this form of model is the computational complexity which is $\mathcal{O}\left(N^3R^3\right)$ where R is the number of dimensions (degrees of freedom) and N the number of observed training points, which here is the length of the signal in time. This issue can be addressed using sparse methodologies that have become popular within the GP literature, good reviews can be found in [126, 127], more specifically [230] shows how a multiple output GP, such as the one in the LFM, can be computed in a sparse manner reducing the computational complexity from $\mathcal{O}\left(N^3R^3\right)$ to $\mathcal{O}\left(N^3R\right)$ using the partially independent training conditional (PITC) sparse approximation.

That being said, work by Hartikainen and Särkkä [228] has shown that, for GPs where the input is temporal and the covariance function is stationary, the inference procedure can be converted into a linear time-invariant (LTI) stochastic differential equation (SDE) which has a linear state-space formulation. This state space model can be exactly solved, for certain cases, using a Kalman filter [192] and RTS smoother [229]. This methodology computes in $\mathcal{O}\left(NR^3\right)$ which is a significant improvement given that, usually, $N \gg R$.

Following Hartikainen and Särkkä [228] and converting the covariance function into an LTI SDE requires the spectral density, by taking the Fourier transform. First, however, it is necessary to present a linear time-invariant stochastic differential equation of order m which has the form shown in Equation (6.7), where $f(t)$ is a random process of interest, $\mathbf{a}$ are some set of coefficients and $w(t)$ is a white noise

process which has a spectral density $S_w(\omega) = q_c$.

$$\frac{\mathrm{d}^m f(t)}{\mathrm{d}t^m} + a_{m-1}\frac{\mathrm{d}^{m-1} f(t)}{\mathrm{d}t^{m-1}} + \ldots + a_1\frac{\mathrm{d}f(t)}{\mathrm{d}t} + a_0 f(t) = w(t). \tag{6.7}$$

This can be rearranged into a state-space model is possible by constructing a vector $\mathbf{f(t)} = \left[f(t)\frac{\mathrm{d}f(t)}{\mathrm{d}t} \ldots \frac{\mathrm{d}^{m-1}f(t)}{\mathrm{d}t^{m-1}}\right]$ which gives rise to a first order vector Markov process.

$$\frac{\mathrm{d}\mathbf{f(t)}}{\mathrm{d}t} = F\mathbf{f(t)} + Lw(t), \tag{6.8}$$

where,

$$F = \begin{pmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & 0 & 1 \\ -a_0 & \ldots & -a_{m-2} & -a_{m-1} \end{pmatrix}, \qquad L = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}. \tag{6.9}$$

The coefficients $a_0, \ldots, a_{m-1}$ (Equation (6.9)) are the coefficients of the denominator in the rational transfer function. The rational transfer function of $\mathbf{f(t)}$ is given by,

$$H\left(i\omega\right) = \frac{b_0}{a_{m-1}\left(i\omega\right)^{m-1} + \ldots + a_1\left(i\omega\right) + a_0} \tag{6.10}$$

where the spectral density of the white noise process is given by the magnitude of the numerator squared, $q_c = |b_0|^2$. From this the spectral density of the process can be written down as,

$$S(\omega) = H(i\omega)q_c H(i\omega)^{-1} \tag{6.11}$$

For an SSM in the form of Equation (6.9), with an observation matrix $C = \begin{bmatrix}1 & 0 & \ldots & 0\end{bmatrix}^\mathsf{T}$, the spectral density of the Markov process can be written as,

$$S(\omega) = C\left(F + i\omega\mathbb{I}\right)^{-1} LqL^\mathsf{T}\left[\left(F + i\omega\mathbb{I}\right)^{-1}\right]^\mathsf{T} C^\mathsf{T} \tag{6.12}$$

It turns out you can do this backwards, working from a covariance function for a

temporal process and calculating the spectral density yields a stable Markov process provided the covariance is stationary.

For example, consider a GP is defined with a Matérn covariance [74, 231] between any two points in time. Since this covariance is defined in terms of the absolute difference between the two input points, $r = |t - t'|$, it is stationary and, therefore, it is possible to perform a conversion to an LG-SSM.

To recap, the general Matérn covariance function is defined as,

$$k(r) = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}}{\ell} r \right)^{\nu} \mathcal{K}_\nu \left(\frac{\sqrt{2\nu}}{\ell} r \right). \tag{6.13}$$

$\mathcal{K}_\nu$ is the modified Bessel function of the second kind. Choosing values of $\nu = p + 1/2$ for p as any non-negative integer lead to simple expressions for Matérn covariance functions. In the case with smoothness parameter $\nu = 3/2$, the covariance function can be written as,

$$k(t, t') = \left(1 + \frac{\sqrt{3}r}{\ell} \right) \exp \left(-\frac{\sqrt{3}r}{\ell} \right). \tag{6.14}$$

For the general form of the Matérn covariance, the spectral density can be written down, by the Wiener-Khinchin theorem,

$$S(\omega) = \sigma_f^2 \frac{2\pi^{\frac{1}{2}} \Gamma\left(\nu + 1/2\right)}{\Gamma\left(\nu\right)} \lambda^{2\nu} \left(\lambda^2 + \omega^2 \right)^{-(\nu - 1/2)}. \tag{6.15}$$

Setting $\lambda = \sqrt{2\nu} \Big/ \ell$ and $\nu = p + 1/2$, in the Matérn class of covariance functions, as before, it can be seen that,

$$S(\omega) \propto \left(\lambda^2 + \omega^2 \right)^{-(p+1)}. \tag{6.16}$$

This can be factorised simply, shown in Equation (6.17), to recover the transfer function needed to convert to a state-space model.

$$H\left(\mathrm{i}\omega\right) = \left(\lambda + \mathrm{i}\omega\right)^{-(p+1)}. \tag{6.17}$$

The spectral density of the white noise process $S_w\left(\omega\right) = q$ is equal to the constant in $S\left(\omega\right)$ (Equation (6.15)) which is,

$$q = \frac{2\,\sigma_f^2\,\pi^{1/2}\lambda^{(2p+1)}\Gamma\left(p+1\right)}{\Gamma\left(p+1/2\right)}. \tag{6.18}$$

For the case of the Matérn covariance with $\nu = 3/2$ as in Equation (6.14) the equivalent state-space model is shown in Equation (6.19).

$$\frac{\mathrm{d}\mathbf{f}(\mathbf{t})}{\mathrm{d}t} = \begin{pmatrix} 0 & 1 \\ -\lambda^2 & -2\lambda \end{pmatrix} \mathbf{f}(\mathbf{t}) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} w(t). \tag{6.19}$$

It is then possible to convert this continuous time state-space model to a discrete time form and solve with a Kalman filter and RTS-smoother [181]. Since the LFM model contains a temporal Gaussian process as the latent function, and the mechanistic portion of the model has a simple state-space representation, it is natural to form a state-space representation of the LFM and this was shown in [227]. This mechanistic model can be shown to include all single degree of freedom (SDOF) and multi-degree of freedom (MDOF) linear oscillators. The spectral density of the white noise process $w(t)$ in Equation (6.19) is given by Equation (6.18) substituting in $p = 1$.

As a demonstration of the novel application of this technique for OMA. This methodology is applied to an SDOF oscillator as the mechanistic model, although it can be easily extended to the MDOF case — or indeed systems with different order ODEs. First, the equation of motion is converted to its state space form, which is again a first order vector Markov process.

Taking the SDOF equation of motion,

$$m\ddot{x} + c\dot{x} + kx = \mathcal{F}, \tag{6.20}$$

dividing through by m, and setting $x_1 = x$, $x_2 = \frac{\mathrm{d}x_1}{\mathrm{d}t} = \dot{x}$ in a state vector $\mathbf{x}$, recovers a state-space representation of the oscillator:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{pmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{pmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{pmatrix} 0 \\ \frac{1}{m} \end{pmatrix} \mathcal{F}. \tag{6.21}$$

The formulation shown in Equation (6.21) is useful where there are known external inputs to the system $\mathcal{F}$ and the effect of the forcing on the system is through the input matrix $B = [0,\ 1/m]^\mathsf{T}$. In the case of OMA these inputs are unknown but can be considered as one or more additional latent states in the model. When the transition for f is linear and described by a matrix U_f this would be written,

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\mathbf{f}} \end{bmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ -\frac{k}{m} & -\frac{c}{m} & \frac{1}{m} \\ 0 & 0 & U_f \end{pmatrix} \begin{bmatrix} x \\ \dot{x} \\ \mathbf{f} \end{bmatrix} + Lw(t). \tag{6.22}$$

The number of additional states that are introduced is dependent upon the model chosen for the transition of $\mathbf{f}$, which is now also in first order vector Markov process form. When using the LFM, it is clear from Equations (6.9) and (6.17) that the choice of covariance function for the GP chosen to represent F will affect the number of additional states required. Stein [92] suggests that Matérn kernels are a more appropriate choice of covariance function since the Squared Exponential imposes an unrealistic smoothness assumption. A choice of $\nu = 3/2$ as in Equation (6.14), is a common choice which appears to perform well, this leads to the addition of two states to the model as in Equation (6.19). This procedure leads to the full four dimensional state space model shown in Equation (6.23).

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{f} \\ \ddot{f} \end{bmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -\frac{k}{m} & -\frac{c}{m} & \frac{1}{m} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\lambda^2 & -2\lambda \end{pmatrix} \begin{bmatrix} x \\ \dot{x} \\ f \\ \dot{f} \end{bmatrix} + Lw(t). \tag{6.23}$$

The model including the noise $Lw(t)$ must now be discretised to perform inference.

At this point any observation model for the observed variable y can be adopted in the process, for instance if measuring displacement it is simply Equation (6.24a), or if measuring acceleration it is Equation (6.24b), for both the observation matrix is multiplied by the discrete states. If other observations were available these could

also be incorporated if a suitable observation model could be formed.

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_t \\ \dot{x}_t \\ f_t \\ \dot{f}_t \end{bmatrix} + \sigma_n^2. \tag{6.24a}$$

$$y = \begin{bmatrix} -\frac{k}{m} & -\frac{c}{m} & \frac{1}{m} & 0 \end{bmatrix} \begin{bmatrix} x_t \\ \dot{x}_t \\ f_t \\ \dot{f}_t \end{bmatrix} + \sigma_n^2. \tag{6.24b}$$

Now, in possession of a model which can represent the system, attention turns to the problem of inference. While methods such as the Expectation-Maximisation (EM) algorithm exist for this form of model [232], these maximum likelihood solutions do not give information about the parameter uncertainties in the model. Instead, a fully Bayesian solution is adopted using Markov Chain Monte Carlo (MCMC) to perform inference over the system parameters and the GP hyperparameters. MCMC is an inference method with guaranteed convergence to the true posterior distribution of interest in the limit [33]. The parameter vector for this model, therefore, is $\theta = \begin{bmatrix} m, k, c, \sigma_f^2, \ell \end{bmatrix}$. It is assumed that the process noise in the dynamics is small (i.e. the system is well described by a linear dynamic model) and that the observation noise is known.

The likelihood of the parameters in the model given the data observed for T time points, $p\left(\theta \,|\, \mathbf{y}_{1:T}\right)$, is related to a quantity called the energy function $\varphi_T\left(\theta\right)$ through a proportionality relationship — the energy function is the negative log likelihood of the filter scaled by a constant. All of the statements made regarding the Bayesian Occam's Razor in relation to the GP are also true of using this quantity in, for instance, an optimisation routine.

$$p\left(\theta \,|\, \mathbf{y}_{1:T}\right) \propto \exp\left(-\varphi_T\left(\theta\right)\right). \tag{6.25}$$

The energy function $\varphi_T\left(\theta\right)$ can be well approximated by the following recursion as the filter runs [182],

$$\varphi_t(\theta) \simeq \varphi_{t-1}(\theta) + \frac{1}{2}\log|2\pi S_t(\theta)| + \frac{1}{2}\mathbf{v}_t^{\mathsf{T}} S_t^{-1}\mathbf{v}_t. \tag{6.26}$$

$\mathbf{v}_t$ and S_t are defined as,

$$\mathbf{v}_t = \mathbf{y}_t - C(A\mathbf{x}_{t-1}), \tag{6.27a}$$
$$S_t = C\left(AP_{t-1}A^{\mathsf{T}} + Q\right)C^{\mathsf{T}} + R, \tag{6.27b}$$

when P_{t-1} is the state covariance at $t-1$. The recursion for φ_T is started at $\varphi_0 = -\log p(\theta)$ [182]. Inference can then be performed using the standard MCMC Metropolis random walk algorithm [32], which for this problem is shown in Algorithm 7.

Algorithm 7 MCMC Metropolis Random Walk for Parameter Inference in State-Space LFM

$n_b, n_s \leftarrow$ Length of burn in and desired number of samples
$\theta_0 \leftarrow \left\{m_0, k_0, c_0, \sigma_{f0}^2, \ell_0\right\}$ ▷ Set Start Points for Markov Chain
$n_a \leftarrow 0, k \leftarrow 0$ ▷ Number of accepted θ, Number of Steps
$p(\theta' \mid \theta_k) = \mathcal{N}(0\,,\ \Sigma_p)$ ▷ Proposal is random walk with diagonal matrix Σ_p
$p(\theta \mid \mathbf{y}) \propto \exp\{-\varphi_T(\theta)\}$ ▷ Posterior Likelihood is Proportional to the Energy Function
while $k < (n_b + n_s)$ **do**
  Sample θ' from $p(\theta' \mid \theta_k)$
  Calculate $\varphi_T(\theta')$ ▷ Equation (6.26)
  $\log(\alpha) = \min\{-\varphi_T(\theta') + \varphi_T(\theta_k), 0\}$ ▷ Log Acceptance Ratio
  $\theta_{k+1} \leftarrow \begin{cases} \theta', & n_a{+}{+} \quad \text{with probability } \alpha \\ \theta_{\mathbf{k}}, & \text{otherwise} \end{cases}$
  $k \leftarrow k + 1$
end while

In this way, it is possible to generate samples from the true posterior of the parameter vector θ. These samples can be used for further uncertainty quantification steps or can be used for model scrutiny. The strength of this method, alongside recovering Bayesian posteriors over the system properties, is to recover, as a latent state of the system, the time series of the forcing signal applied to the system.

6.3 Application to Operational Modal Analysis

It is clear from the formulation that this model is directly applicable to the problem of modal analysis. One of the strengths of the model presented here is the flexibility available due to the non-parametric form of the forcing function as defined by the GP. This also presents a problem in the practical application of the model. As well as the normal system identification problems with the scaling of the mass, stiffness, and damping parameters which lead to non-identifiability in the model when the forcing level is unknown; the ability of the GP to model behaviour very similar to the dynamics (it is not restricted to a Gaussian white noise form) means that it can mask dynamics in the system. For this reason it is necessary to place prior distributions over the system parameters. Prior distributions can also be placed over the hyperparameters of the GP model to control its behaviour without breaking the Bayesian paradigm within which the model has been set up.

In the current work it was found to be necessary to constrain at least one of the system parameters to a fixed value to resolve these problems. In this case it is assumed that one of the system parameters is known with certainty *a priori*, this can be interpreted as a delta prior on that parameter. The other prior distributions can be elicited from knowledge of the system. This could be from a finite element (FE) model where, if necessary, reduced order modelling techniques can be employed [233]. This would be similar to the approach seen in, for example [215]. If in possession of a verified and validated model of a complex system this could be a powerful approach although it remains an open question as to whether it would be better to adopt a 'ground up' approach to building this model. It should be noted that when using MCMC for inference there is no restriction on the distribution of these priors, therefore, empirical distributions can be used if available from testing — the author has not found an instance of this in the literature but it is a simple intuitive jump.

The choice of prior over the hyperparameters of the GP poses a more difficult problem as they are less interpretable in a physical sense. It is possible to set a formal uniform prior, for example $p\left(\sigma_f^2\right) = 1$ which is a form of improper prior. However, it can be helpful (in the author's experience!) to set the prior over the signal variance σ_f^2 around the expected variance of the loading and the length-scale ℓ based upon the expected frequency content of the loading. This is an intuitive leap from understanding the role of the signal variance in the specification of the Matérn

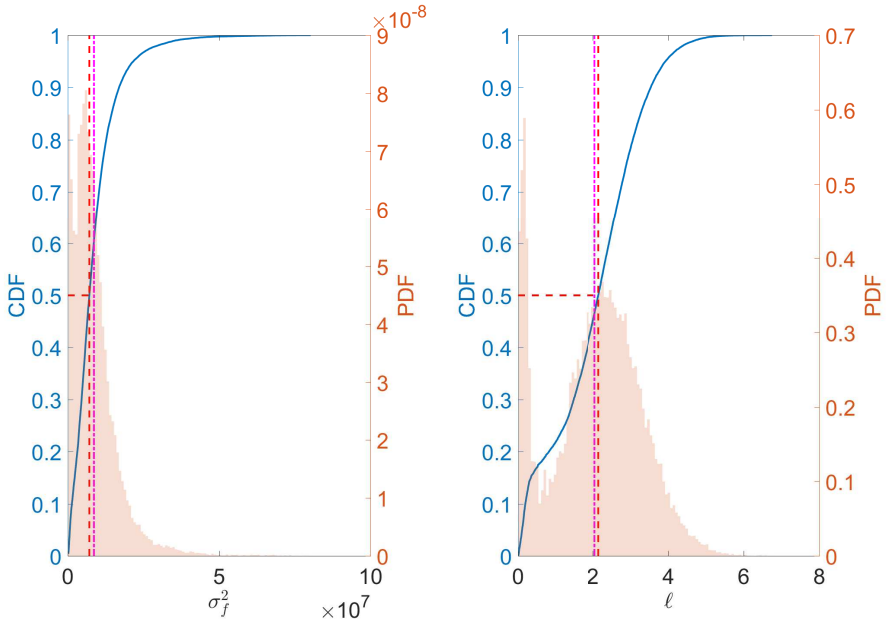


Figure 6.1: Plots showing the histograms and CDFs for the parameters being inferred when system parameters are known. The mean values are shown with the dot-dash $(\cdot-)$ line in magenta, and the values estimated from the empirical CDF with the dashed $(--)$ line in red.

kernel.

6.4 Results

The results presented here paper are from a simulated SDOF system response to a measured wave force input where the parameters of the model are chosen such that the natural frequency of the system is close to the narrowband loading frequency. The wave loading data was measured as a part of the Christchurch Bay project previously introduced in Chapter 4. This represents, only, the initial work into the application of LFM state-space models for OMA.

The parameters of the SDOF model are chosen to be: $m = 2000$kg, $k = 100$N/s and $c = 5$Ns/m. These values correspond to modal parameters $\omega_n = 0.2$Hz and $\zeta = 0.0079$. This presents a problem that is normally difficult in a OMA setting where the loading frequency is close to the resonant frequency of a lightly damped mode.

All of the parameters of the model must take only positive values for them to be valid,

therefore, the random walk of the Markov-Chain must take place in a transformed space which ensures this. The values in the transformed space are denoted $(\hat{\cdot})$, with the transformation defined as:

$$\theta = \log\left\{\exp\left(\hat{\theta}\right) + 1\right\}, \tag{6.28}$$

where θ is the parameter of interest. Prior probability distributions over the system parameters are chosen with a random perturbation on the mean values in the system parameters to simulate priors elicited from an incorrect model (e.g. an uncalibrated FE model). To constrain the problem it is assumed that the mass of the system is known *a priori* and as such this is modelled with a delta prior for all tests.

Inference was performed as in Algorithm 7 with the burn-in set to 1×10^3 samples and the total number of samples as 1×10^5. Since the distributions may be non-Gaussian, and the distributions over the Gaussian Process hyperparameters are expected to be multi-modal, the parameter estimates used to calculate the example force are taken to be the values when the empirical cumulative density function (CDF) is equal to 0.5.

6.4.1 LFM With Known System Parameters

As an initial test the model is run with the system parameters fixed at the true values to demonstrate the ability of the LFM to model the input to the system. This may be of interest if it is believed that the parameters of the system are known with almost certainty and it is decided that performing inference over them is not of value. The priors used for this model are summarised below:

$$p\left(\hat{\sigma}_f^2\right) = 1, \tag{6.29a}$$
$$p\left(\hat{\ell}\right) = \mathcal{N}\left(0\,,\ 2\right), \tag{6.29b}$$
$$p\left(m\right) = \delta\left(2000\right), \tag{6.29c}$$
$$p\left(k\right) = \delta\left(100\right), \tag{6.29d}$$
$$p\left(c\right) = \delta\left(5\right). \tag{6.29e}$$

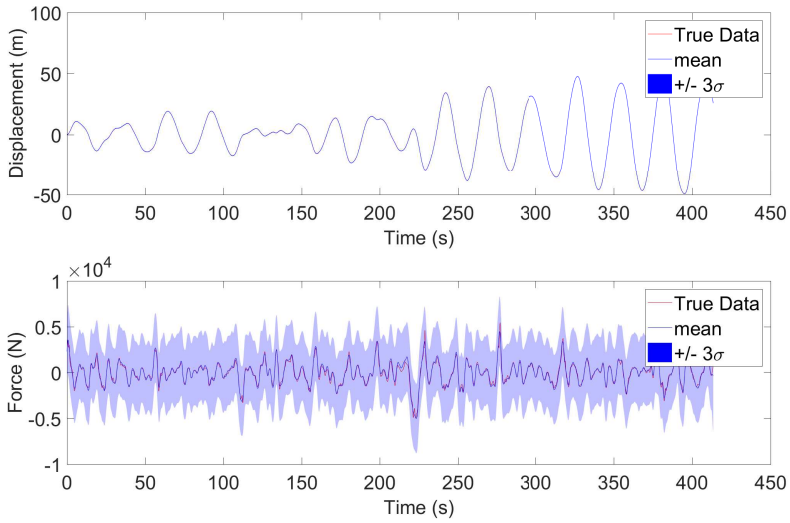


Figure 6.2: Smoother estimate of displacement and forcing signal on the system when m, k, and c are known. True values are shown in red, mean estimates as a solid blue line and three standard deviation confidence in the shaded regions

Parameter	True Value	Prior Mean	Posterior Mean	CDF Estimate
σ_f^{2*}	-	-	8.64×10^6	7.12×10^6
ℓ^*	-	0.693	1.760	2.148

Table 6.1: Table showing results for the GP hyperparameters when the system parameters are known and fixed. (* denotes that the true values for these parameters are unknown.)

where $\delta(\cdot)$ indicates a value drawn from a delta distribution, i.e. if sampled this variable will always be equal to the parameter given. Using this procedure, samples from the posteriors over the signal variance and length-scale hyperparameters were drawn. Plots of the PDF normalised histogram and CDFs are shown in Figure 6.1 along with mean estimates and CDF estimates of the values. A summary of the results is shown in Table 6.1. For the GP modelling the latent force on the system, the signal variance of the forcing σ_f^2 is assumed unknown and a uniform prior is used, the length-scale parameter has a prior to discourage very long or short length-scales. Long length-scales will cause the function in the forcing to become very smooth, the extreme of which is that it remains at the mean, in this case the signal variance increases and the solution returned is one in which all the forcing is considered to

be noise. This, although a valid solution, is not helpful in the setting of modal analysis where retrieval of the forcing time series adds a significant amount of valuable information. If the length-scale is allowed to become very small, then the GP will attempt to model the noise in the signal as if it were functional, this is also clearly not a desirable situation. Since the distributions are skewed and multi-modal, the mean estimates are not a good value to take and instead the CDF estimate is used to predict the forcing experienced by the structure. This multi-modal nature of the distribution over length scales is discussed in [87] and is related to the Type-I and Type-II solutions observed from the GP. Figure 6.2 shows the smoother prediction over the forcing as well as the observed state. As expected the prediction over the displacements is modelled well with very small variance. The prediction over the forcing is good, and there is a normalised mean square error value of 3.3 with the mean prediction, however, one of the strengths of this model is that in addition to this there are also confidence intervals which represent the uncertainty in the forcing.

6.4.2 OMA With The LFM

The test was run again, this time assuming that the system parameters were not known *a priori* and are to be determined as part of the inference procedure. For the case of performing OMA with the GP LFM, the priors used in this model are shown below:

$$p\left(\hat{\sigma}_f^2\right) = 1, \tag{6.30a}$$

$$p\left(\hat{\ell}\right) = \mathcal{N}\left(0\,,\ 2\right), \tag{6.30b}$$

$$p\left(m\right) = \delta\left(2000\right), \tag{6.30c}$$

$$p\left(\hat{k}\right) = \mathcal{N}\left(105.9\,,\ 2500\right), \tag{6.30d}$$

$$p\left(\hat{c}\right) = \mathcal{N}\left(4.8\,,\ 6.25\right). \tag{6.30e}$$

When running the model where only the mass is known, samples are generated from the posteriors over the GP hyperparameters and the system parameters, k and c. Plots of the histograms and CDFs of the parameters are shown, as before,

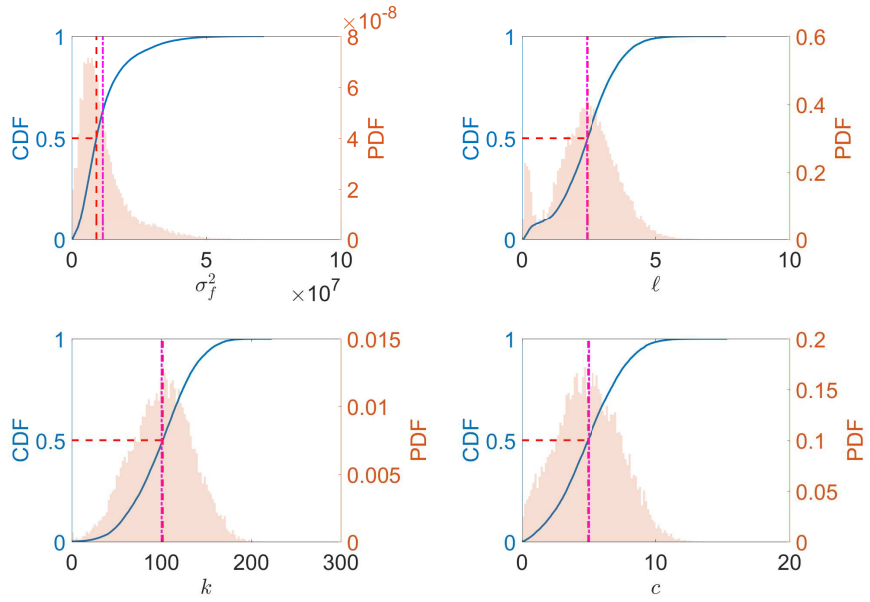


Figure 6.3: Plots showing the histograms and CDFs for the parameters being inferred in the full LFM OMA procedure. The mean values are shown with the dot-dash $(\cdot-)$ line in magenta, and the values estimated from the empirical CDF with the dashed $(--)$ line in red.

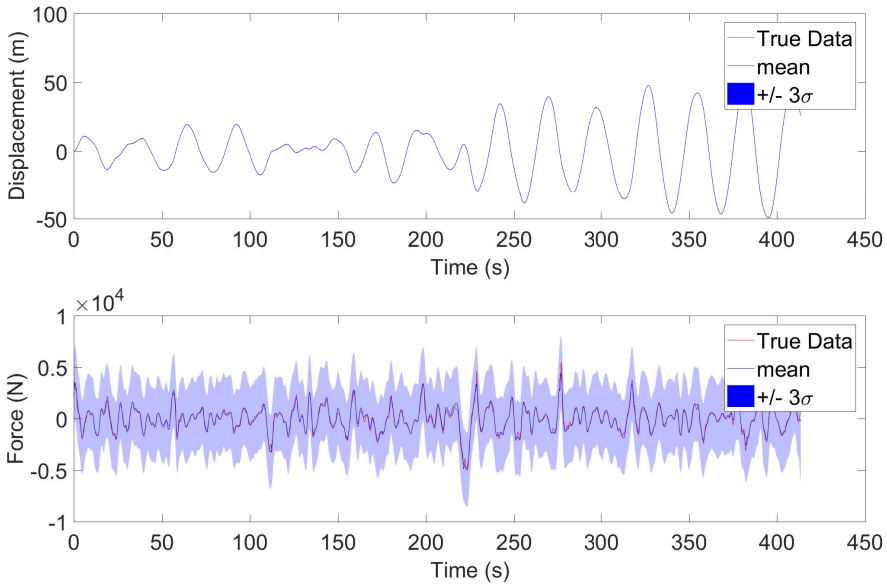


Figure 6.4: Smoother estimate of displacement and forcing signal from the LFM OMA on the system with true values shown in red, mean estimates as a solid blue line and three standard deviation confidence in the shaded regions

Parameter	True Value	Prior Mean	Posterior Mean	CDF Estimate
σ_f^{2*}	-	-	1.153×10^7	9.154×10^6
ℓ^*	-	0.693	2.283	2.443
k	100	105.9 (5.9%)	100.2 (0.2%)	101.4 (1.4%)
c	5	4.81 (-3.8%)	4.91 (-1.8%)	4.93 (-1.4%)

Table 6.2: Table showing results from the full LFM OMA model, values in brackets denote the percentage error versus the true values. (* denotes that the true values for these parameters are unknown.)

in Figure 6.3 and a summary of the results is shown in Table 6.2. It can be seen that the parameter estimates of the system parameters are within 2% for both the mean values of the distributions and the CDF estimates, this would indicate that the distributions are uni-modal and are not heavily skewed. When the values from the CDF estimates are used to run the filter-smoother over the full dataset, the predictions in Figure 6.4 are made. The prediction over the force continues to be accurate with a normalised mean-squared error on the mean of the signal of 3.1. This result is comparable to the results obtained when the system properties are known, which is unexpected and encouraging. It follows that the estimates of the signal variance and length-scales are similar for this model and the model with known system parameters. It is also reassuring that the values for the GP hyperparameters are similar to those from the case where the system is known. This would suggest that the solution is stable and that the use of priors has alleviated some of the identifiability issues arising from the flexibility of the GP prior.

Stepping back, these results are in some ways remarkable. The quality of prediction of the forcing time history is nearly identical in the case of both known and unknown parameters. The restriction of fixing the mass in the model is similar to limitations seen in standard OMA analysis [38]. The results seen here would suggest that it is in fact possible to perform OMA when the forcing is non-Gaussian and close to resonant frequencies by replacing the assumption that the forcing can be modelled by a white noise sequence with the assumption that it can be modelled by a Gaussian process. What is necessary is to robustly determine the model order, however, this is normally accessible from other models of the system e.g. an FE model.

6.5 Discussion

The work presented here has demonstrated, through application to a simple system the benefit of using a Gaussian Process Latent Force Model in the task of operational modal analysis. Specifically, the state-space formulation of this model has allowed for efficient inference to be made and as such Markov-Chain Monte-Carlo methods can be used to recover distributions over the system parameters and the GP hyperparameters — which in turn returns the distribution over the time history of the forcing. It has been shown that these distributions can be used to elicit accurate point estimates of the parameters and also of the loading, something that is normally not possible in an OMA setting.

It is worth noting, however, that the tasks shown here have been performed on simulated data under a number of significant assumptions. First it is assumed that the linear model of the system can represent the behaviour well with low process noise — i.e. the system has a known order and is linear. Second the model is tested with a low level of artificial Gaussian white noise added. Thirdly there exist priors over the system parameters in which there is non-negligible probability mass at the true values. It is normally possible to obtain these priors, either from a first principles physics model or from a reduced order FE model. Finally, it is assumed that the mass of the system is known accurately *a priori*, this is required to constrain the unidentifiability inherent in the model. This unidentifiability stems from the scaling of the forcing and the system parameters by the mass, that is, a system with the same response can be recovered if all the system parameters are multiplied by some constant α and if the forcing magnitude is also scaled by α. The flexibility of the GP as a prior over the forcing function only increases this issue. By fixing the mass and placing informative priors over the system parameters the model can be constrained in such a way that it is more likely to converge to the true parameter posteriors.

This remains a powerful *grey-box* approach to the problem of OMA. It is clear that the model is capable of returning good estimates of the forcing, and the system parameters under the constraints already discussed. The results presented here motivate further work into the use of GP LFM models in the modal analysis setting. This includes the extension of the current model to a multi-degree of freedom case. This extension merely requires increasing the state vector to include the additional degrees of freedom and including the interactions between these states. Additionally,

in nonlinear systems where the structure of the nonlinearity is known (e.g. a cubic stiffness term), this model is also applicable although the Kalman filtering formulation must be replaced with a nonlinear counterpart. This would require a combination of the techniques presented in this chapter with those seen in the previous chapter. The challenge in inference becomes significantly harder in this case and is practically impossible due to identifiability when the form of the nonlinearity is not known. This can be a major challenge on full-scale structures, it might be known that a nonlinearity exists in the system, however, it can be difficult to define this parametrically to allow it to be modelled. It may be an alternative to, in this case, combine this approach with a more flexible model such as the Gaussian process state-space model. However, here the flexibility of the state-space model may make it impossible to separate the dynamics of the system from the loading experienced. Therefore, it will likely be necessary to define the nonlinearity in the system *a priori*.

The results presented in this chapter would suggest that this LFM approach to modelling the wave loading on structures is preferable to the direct modelling approach in Chapter 4. However, this should be considered carefully. While the results presented here have been shown to more accurately predict the wave force on the structure than the experiments shown in direct modelling, these results are shown for a simulated system where the model order is known. Although it is outside the scope of this thesis, forming reduced order models of physical structures which are subject to these kinds of loading is no simple task. The selection of model order will have a significant impact on the effectiveness of the algorithm and the flexibility of the GP may make it difficult to distinguish when the method is not performing as expected. The methodology here can be considered a form *grey-box* model where physical knowledge about a system can be exploited within a machine learning methodology.

Chapter 7

Online Bayesian Clustering for Damage Detection

Highlights:

- *A Dirichlet process model is introduced for online Bayesian clustering of SHM data*
- *The technique removes the need to pre-collect a training dataset or add information on expected damage conditions*
- *Random Projection is exploited to allow online unsupervised dimensionality reduction*
- *A framework which allows for incorporation of prior knowledge of damage states leads to a flexible and practical model*

This chapter switches focus from a regression based approach to SHM where the system is modelled to attempt to understand more about its behaviour to a classification problem where SHM is approached as a data labelling problem. In many ways this is a more direct approach to solving SHM problems in which the fundamental questions are addressed head on. These being,

- Is a given structure damaged?

- If so, what is the type, location and extent of that damage?
- Given the information regarding the structure what is its expected remaining useful life (prognosis)?

These steps are well summarised in Rytter's hierarchy [6, 7], discussed earlier in this thesis. For the first two levels of the hierarchy, detection and classification of damage, it is more natural to use a classification technique as these situations have discrete labels. Often these may be the first tasks that an end user may expect of an SHM system when applying it to a physical structure. This can help inform a Risk & Reliability based inspection scheme, where inspections are carried out based on the expected cost of that particular action relative to the cost of differing actions or indeed inaction. Given the maturity and availability of sensing hardware, a data-driven approach is commonly adopted when attempting to solve these classification problems. Here, statistical models can be used to detect similarity (or difference) between sets of data collected from a structure, which is, in turn, used to infer its health/condition. Data-driven approaches, if wanting to achieve more than novelty detection, require training data from multiple healthy and damage states which is a significant limitation.

In many cases, it will not be possible to acquire data covering all healthy conditions and damage scenarios, the main limitation being the cost of producing and subsequently damaging large valuable structures, e.g. within the aerospace industry or civil infrastructure. A particular challenge in civil infrastructure stems from the fact that structures are often unique. The existence of a number of different damage scenarios comes from the multiple mechanisms for damage that a structure might experience. For example, in an aerospace structure it would be desirable to detect degrading performance from fatigue damage accrual, but damage introduced by low-velocity impact is also of concern. In certain cases it will be unsafe to operate the structure with a given type of damage present, meaning that collection of data from this damage state prior to operation of the structure is not possible. Additionally, a structure will operate in a number of different operational and environmental conditions, which result in significant changes to the measured dynamic behaviour. Continuing with the example of an aerospace structure, it is clear that there will be significant changes in behaviour between flight and taxiing. It is less obvious, however, that there may be other confounding influences such as crosswinds on landing or freezing temperatures which will affect the behaviour of the structure.

One could go on attempting to imagine all the scenarios possible for changes in operating condition, but this is a fruitless exercise, as it quickly becomes apparent that collecting data from all these conditions is not feasible [6, 51], not least because the operator normally has little or no control over these factors.

It is desirable, therefore, to consider methods which will allow the incorporation of operational data into the training of a given algorithm, which adapts, as time progresses. These methods are commonly referred to as *online learning* [75]. Rather than pure novelty detection from a known, healthy, baseline state, it would be beneficial to be able to first detect a new regime, then label it and be able to recognise that behaviour, should it occur, in the future. This is also sometimes called the *semi-supervised* learning approach [234], where new regimes are discovered in the data which are labelled in operation and incorporated into future analysis, this process of inspecting online leads to a partially labelled dataset.

This chapter will introduce the use of the finite Gaussian Mixture Model (GMM) for clustering SHM as a stepping stone to presenting the use of a Dirichlet Process (DP) model as a Bayesian approach to online clustering of SHM data. It will be shown how this methodology allows operation of the SHM system from day one without the need to pre-collect a training dataset. It also will be shown that this algorithm does not require a pre-specified expected number of damage states but instead the number of clusters in the model is learnt online in a Bayesian manner along with the cluster parameters (means and variances). This methodology is applied to two case studies, the first a lab structure, and the second a popular benchmark dataset collected from a full scale bridge structure.

It is useful to provide a short overview of a number of alternate methods for the classification task in order to highlight the effectiveness of the newly proposed method. The machine learning perspective has, generally, considered only *unsupervised* and *supervised learning* tasks[1] [6]. *Unsupervised learning* applications are dominated by one-class classification tasks [10] based on outlier analysis [12, 16]. A baseline healthy state is used to define a "normal" condition and then deviations from this can be detected in an online manner. The problem of *supervised learning* in SHM is usually concerned with regression or classification tasks which provide information

[1]For clarification, here, *unsupervised learning* is defined as a situation in which data is available without any labels or outputs. This can include the case where a dataset is collected from what is assumed to be a *normal* condition. The *supervised learning* task is treated as one where data is available with both the inputs and outputs (either continuous or labels) from which methods for classification or regression can be trained.

regarding the type, location, or severity of damage in a structure [7].

Treatment of SHM as an *unsupervised learning* task has been mainly limited to an outlier detection problem, usually in a laboratory setting [13, 235]. The challenge in this research has been in building algorithms that are robust to false alarm and environmental changes. A number of methods have been developed which handle this problem well [17, 52]. However, a drawback to the most common approaches to dealing with confounding influences, is that they reduce SHM to a one-class problem [10], where distinction is only made between normal (previously observed) and abnormal (newly observed) states. This fails to give additional information about the operating conditions of the structure, which would be useful for an operator to know, or, indeed, about any damage or performance anomalies that occur. To counteract this, a popular approach has been to consider clustering in an unsupervised manner. The most common approaches employ Gaussian Mixture Models [50, 236–238], or other clustering techniques [239–242] in an offline manner. Tibaduiza *et al.* [243] present another unsupervised methodology based on self-organising maps of features from ultrasonic pitch-catch data.

The alternative to this, and preferred option when interested in additional information, is the *supervised learning* task. Here, a training dataset is formed which has information from all possible structure states; inference about the current state can now be made via pattern recognition or machine learning methods, where new observations/data are compared to the training set.

Although tools exist which perform very well in the supervised learning problem, a common stumbling block is the lack of availability of complete datasets for algorithm training. It is usually prohibitively expensive to acquire training data from all environmental conditions and damage states. For this reason, the development of algorithms which can be established/learn fully or partially online is of particular interest. Langone *et al.* [244] propose an adaptive learning algorithm based on a kernel PCA transformation, they demonstrate this by performing damage detection on a benchmark dataset — the Z24 bridge. The algorithm performs well on benchmark data but requires an initialisation and calibration phase before being fully operational; in this phase, the structure is assumed undamaged. The method also requires user input regarding thresholds and the expected number of clusters. Chen *et al.* [245] present a semi-supervised algorithm for damage detection based on a multi-resolution classification with adaptive graph filtering; the features are extracted by passing the input signals through a filter bank. A graph-filtering algorithm estimates the labels

for unknown data given previously labelled features, and a regression step is able to compensate for missing data in the problem informed by the graph filter.

Finite Gaussian Mixture Models (GMMs) in SHM have been used previously with promising results [246–248]; the strength of the GMM is in the ability of training data to shape clusters and form a probabilistic representation of the different possible states that the structure could be in, undamaged or damaged, with the possibility for multiple examples from each. The key difficultly in implementing a finite GMM without a complete training set is the specification of the number of Gaussians in the mixture. The method proposed in this paper uses a DP clustering model to remove the need to pre-specify the number of clusters that are expected, while retaining a Bayesian formulation as opposed to methods such as affinity propagation [249].

DP models have been employed in a number of machine learning tasks including: Natural Language Processing [250] and topic modelling [251, 252], where documents can be grouped according to thematic similarities. In image analysis, the model has been used to generate captions for images [253]; it has also found use in medical image analysis [254, 255], for clustering regions of the brain from data collected by MRI or fMRI, and in genetic analysis [256, 257]. In other medical applications, DP mixture models have been used for sorting neural spike data [258].

Previously, a DP mixture model has been shown to be effective in the feature selection step in SHM [259]. In that work, the outputs of the DP clustering model are used as features in a further analysis step — a particle-filter based damage progression model — where they are combined with a physical model. Only the number of clusters identified by the DP is used as a feature, which does not make full use of the Bayesian nature of the DP clustering method.

The approach adopted here makes use of the Bayesian properties of the DP to allow incorporation of prior knowledge and updates of belief given observed data. The aim is to avoid the need for a training dataset before the process begins, but retain flexibility to include any training data as a formal prior belief. In addition there is a reduction in the number of required user-tuned parameters in the model. In this way, a model is developed which can perform powerful *online learning* with minimal required *a priori* knowledge in terms of access to data or a physical model. The work in this paper aims to show how such a model can be implemented online for use in SHM. To achieve this, a novel feature selection approach is also explored, making use of Random Projection [260] of high dimensional frequency domain features.

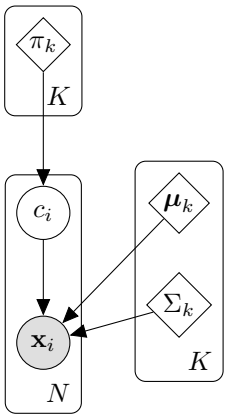


Figure 7.1: Graphical model of a finite Gaussian Mixture Model with K components in the mixture.

7.1 Finite Gaussian Mixture Models

A helpful introduction before presenting the full model is to recap the use of finite GMMs. Modelling data which are inherently non-Gaussian — which many collected datasets are — poses a challenge, as typically the inference becomes harder. It is possible to imagine that the data has been generated, not by some complex non-Gaussian process, but from a mixture of independent Gaussian distributions. In SHM, one could assume that during normal operation, features are clustered according to one or more Gaussian distribution; however, when damage occurs, the features are drawn from a separate set of Gaussians with different parameters. More Gaussians can be added to cover many different scenarios relating to changing operating conditions or different damage cases.

It is possible to construct a probabilistic model which describes this behaviour. First, one proposes a multinomial distribution $\boldsymbol{\pi}$, in which each element π_k is the probability that a data point comes from each class, $k = 1, \ldots, K$ for K classes, and $\sum_{k=1}^{K} \pi_k = 1$. In other words, $\boldsymbol{\pi}$ is merely the probability that the structure is in each state.

Each state of the structure is defined by its own Gaussian distribution which has a mean, $\boldsymbol{\mu}_k$, and covariance, Σ_k. The graphical model of this is shown in Figure 7.1 and it is possible to write it down as below:

$$\begin{aligned} \mathbf{x}_i | c_i &\sim \mathcal{N}\left(\boldsymbol{\mu}_{c_i}, \Sigma_{c_i}\right) \\ c_i &\sim \text{Mult}\left(\boldsymbol{\pi}\right) \end{aligned} \tag{7.1}$$

In order to use this model, the parameters must first be determined. The parameters of the model include: the number of clusters K; the mixing proportions $\boldsymbol{\pi}$; and the cluster parameters, $\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K, \Sigma_1, \ldots, \Sigma_K\}$. This gives a total parameter vector, $\Theta = \{K, \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K, \Sigma_1, \ldots, \Sigma_K\}$. Additionally, θ_k is defined as $\theta_k = \{\pi_k, \boldsymbol{\mu}_k, \Sigma_k\}$. Determining these parameters can be accomplished quite efficiently via Expectation Maximisation [75] for $\theta_{1:K}$ and for K either the Bayesian Information Criterion [261] or Akaike Information Criterion [262] can be used. This will give the maximum likelihood solution to the model given the currently observed data, however, a Bayesian solution to the problem has also been explored for SHM [263] or more generally, for the GMM, in [264].

7.2 Dirichlet Process Gaussian Mixture Models

The desirable modification to this hierarchical finite GMM is to make the inference over Θ Bayesian. This will give more robust estimates of the parameters, $\theta_{1:K}$ (i.e. the parameters in θ_k for all clusters $k = 1, \ldots, K$), and allow a probabilistic selection of K through use of the Dirichlet Process prior. The Bayesian approach allows incorporation of prior knowledge, such as the expected effects of damage, in a formal manner. This can be done by fixing labels to certain data points *a priori* rather than learning them or by grouping certain cluster labels using some form of meta-labelling (a label which is separate from the clustering procedure but defines multiple clusters as having the same class — e.g. for handling multiple undamaged conditions). Conversely, it also allows the data observed to shape the model belief through calculation of posterior distributions.

Firstly, as is standard in any Bayesian analysis, priors are placed over the cluster parameters $\boldsymbol{\mu}_k$ and Σ_k. To help with inference over the model, these priors are chosen to be conjugate with the Gaussian distribution which is the likelihood, therefore, the prior over the means is a multivariate Gaussian and the prior over the covariances is an Inverse-Wishart ($\mathcal{IW}$). The choice of conjugate priors ensures that updates to the

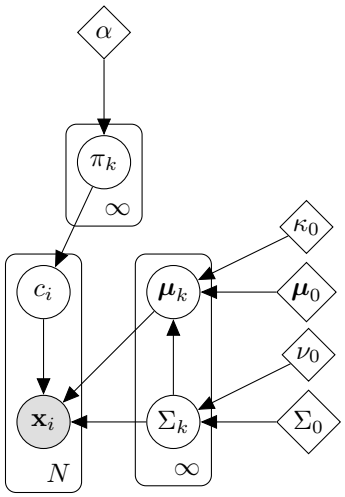


Figure 7.2: Graphical model of the Infinite Gaussian Mixture Model.

cluster parameters can be computed in closed form given the observed data in each cluster. This primarily is a computational advantage, however, it also avoids any error that may occur from approximate the posteriors of these parameters for example by MCMC. These prior distributions have their own hyperparameters associated with them which are, $\boldsymbol{\mu}_0, \kappa_0, \Sigma_0, \nu_0$. It is usual to combine these into a single prior distribution over the cluster parameters H.

$$\begin{aligned} H &= \mathcal{NIW}\left(\boldsymbol{\mu_0}, \Sigma_0, \kappa_0, \nu_0\right) \\ &= \mathcal{N}\left(\boldsymbol{\mu}|\boldsymbol{\mu_0}, \frac{\Sigma}{\kappa_0}\right) \mathcal{IW}\left(\Sigma|\Sigma_0, \nu_0\right) \end{aligned} \tag{7.2}$$

To perform Bayesian inference over the mixing proportions $\boldsymbol{\pi}$, as well as the cluster parameters, another prior must be specified. The sensible choice again is to choose the conjugate prior to the Multinomial distribution, which is a Dirichlet distribution governed by a strength parameter α, which is a single number when a symmetric Dirichlet distribution [76], is used, as in this case. Following [265], it is possible to take the limit of $K \to \infty$ and form an infinite Gaussian mixture model (IGMM) for which the generative model is shown in Equation (7.3) and the graphical model is seen in Figure 7.2.

$$\mathbf{x_i}|c_i \sim \mathcal{N}(\mathbf{x_i}|\boldsymbol{\mu}_{\boldsymbol{c_i}}, \Sigma_{c_i}) \tag{7.3a}$$

$$\boldsymbol{\mu}_{\boldsymbol{c_i}}|\Sigma_{c_i}, c_i \sim \mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{c_i}}|\boldsymbol{\mu_0}, \frac{\Sigma_{c_i}}{\kappa_0}\right) \tag{7.3b}$$

$$\Sigma_{c_i}|c_i \sim \mathcal{IW}\left(\Sigma_{c_i}|\Sigma_0, \nu_0\right) \tag{7.3c}$$

$$c_i|\boldsymbol{\pi} \sim \text{Mult}\left(\boldsymbol{\pi}\right) \tag{7.3d}$$

$$\boldsymbol{\pi} \sim \text{Dir}\left(\alpha\right) \tag{7.3e}$$

The strength of this formulation for a mixture model, in the SHM context, is that only the hyperparameters need to be specified to use the model, there is no tuning of thresholds or calibration phase. Practically, this means that to implement the model, the operator does not need to specify a number of expected normal or damage conditions, which is difficult or impossible for a structure in operation. Nor does the user need to specify the expected changes that damage on the structure will introduce to the data (derived from the physical mechanism of damage or a large number of expensive tests); although in the presence of training data, this can be easily introduced by including clusters in the model where the prior parameters of those clusters are the posteriors of the parameters when the known data is added to the cluster. Or by fixing certain data points assignment to certain clusters the algorithm is used in a semi-supervised manner.

A collapsed Gibbs sampler can be used to make efficient online inference over this model [266]. The collapsed Gibbs sampler refers to the process of analytically marginalising certain variables in the model — this is made possible by the choice of priors in the model. For the DPGMM, the cluster parameters, means and covariances, can be marginalised analytically removing the need to sample them. This gives a posterior distribution over each of the parameters against which new data can be assessed via the posterior predictive distribution, $p(\mathbf{x_i}|\mathcal{D}_{-i})$[2], the likelihood of point $\mathbf{x_i}$ given the rest of the observed data. Although potentially faster algorithms for variational inference in the Dirichlet Process mixture model exist [267, 268], it is more practical to implement the Gibbs sampler when performing inference online. The nature of the Gibbs sampling solution is that each data point is assessed marginally in the sampler, this allows the addition of new points online rather than requiring batch updates. It also makes the incorporation of prior knowledge simple, as points

[2]The notation $-i$ is used to indicated all points except for point i

which have known labels can be excluded from the Gibbs sampler.

For the case of a Gaussian base distribution, the Gibbs sampler proceeds as follows. The already collected data (if present) are initially assigned to random clusters, then at each iteration one of the data points is chosen to be (re)assessed. This point is removed from its current cluster assignment, c_i, and the parameters of that cluster are updated. If that data point was the only point assigned to that cluster, it is destroyed and the total number of clusters, K, is updated. For each cluster, $k = 1, \ldots, K$, the prior likelihood that the point was drawn from that cluster k, is assessed. The prior is a Dirichlet Process prior, which for an existing cluster is equal to:

$$p\left(c_i = k \,|\, \mathbf{c}_{-i}, \alpha\right) = \frac{N_{-i,k}}{N + \alpha - 1} \tag{7.4}$$

It can be seen that the prior likelihood is governed by the hyperparameter, α, and the number of points currently assigned to that cluster, $N_{-i,k}$. The prior encourages clusters to grow, increasing α will make a higher number of clusters more likely. Since the information from the other data points should also be included in the clustering process, the likelihood term must be computed to get the posterior likelihood of the point belonging to each cluster, up to a constant. That is, compute: $p\left(c_i = k \,|\, \mathbf{x_i}, \mathbf{c}_{-i}, X_{-i,k}, \alpha, \boldsymbol{\beta}\right) \propto p\left(\mathbf{x_i} \,|\, X_{-i,k}, c_i = k, \boldsymbol{\beta}\right) p\left(c_i = k \,|\, \mathbf{c}_{-i}, \alpha\right)$ where $\boldsymbol{\beta} = \{\boldsymbol{\mu}_0, \Sigma_0, \kappa_0, \nu_0\}$, the prior constants of the base distribution. This is the posterior probability for the assignment of data point i to cluster k, given the data value, $\mathbf{x_i}$, the current cluster assignments, $\mathbf{c}_{-i}$, the data already assigned to that cluster, $X_{-i,k}$, and the hyperparameters α and $\boldsymbol{\beta}$.

The computation of the likelihood term, $p\left(\mathbf{x_i} \,|\, X_{-i,k}, c_i = k, \boldsymbol{\beta}\right)$, involves calculating the posterior predictive likelihood of that data point $\mathbf{x_i}$ being in cluster k. As data are added to each cluster the parameters of that cluster are updated via conjugate (closed form) updates to the Gaussian which defines it. The model requires a posterior distribution over the parameters of each Gaussian cluster: $\boldsymbol{\mu}_k$ the mean and Σ_k the

variance. This leads to a prior over the cluster parameters[3],

$$\Sigma \sim \mathcal{IW}_{\nu_0}(\Sigma_0) \tag{7.5a}$$

$$\boldsymbol{\mu}|\Sigma \sim \mathcal{N}(\boldsymbol{\mu}_0, \Sigma/\kappa_0) \tag{7.5b}$$

$$p(\boldsymbol{\mu}, \Sigma) \equiv \mathcal{NIW}(\boldsymbol{\mu}_0, \kappa_0, \Sigma_0, \nu_0) \tag{7.5c}$$

$$\propto |\Sigma|^{-((\nu_0+d)/2+1)} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\Sigma_0^{-1}\Sigma^{-1}\right) - \frac{1}{2}\kappa_0(\boldsymbol{\mu}-\boldsymbol{\mu}_0)^{\mathsf{T}}\Sigma^{-1}(\boldsymbol{\mu}-\boldsymbol{\mu}_0)\right) \tag{7.5d}$$

The updates to the posterior parameters of the cluster are efficient since the priors have been chosen to be conjugate. The conjugate updates when n data points have been observed, are computed as shown,

$$\boldsymbol{\mu}_n = \frac{\kappa_0}{\kappa_0+n}\boldsymbol{\mu}_0 + \frac{n}{\kappa_0+n}\bar{x} \tag{7.6a}$$

$$\kappa_n = \kappa_0 + n \tag{7.6b}$$

$$\nu_n = \nu_0 + n \tag{7.6c}$$

$$\Sigma_n^{-1} = \Sigma_0^{-1} + S + \frac{\kappa_0 n}{\kappa_0+n}(\bar{\mathbf{x}}-\boldsymbol{\mu}_0)(\bar{\mathbf{x}}-\boldsymbol{\mu}_0)^{\mathsf{T}} \tag{7.6d}$$

Here, S is defined as the sum of squares matrix around the sample mean, $\bar{x}$,

$$S = \sum_{i=1}^{n}(x_i-\bar{x})(x_i-\bar{x})^{\mathsf{T}} \tag{7.7}$$

It can be shown that when updating a cluster by a single data point (adding or removing a single point), the updates can be carried out as Rank 1 updates to a Cholesky decomposition of the covariance matrix of the posterior, which significantly improves the speed of the computation [269, 270]. The distribution of interest for calculating the likelihood term in the DP mixture model is sometimes referred to as the posterior predictive distribution $p(x \mid \mathcal{D}_k)$; the likelihood that a new point x

[3]Here $\mathrm{tr}(\cdot)$ indicates the trace operator

was drawn from the posterior distribution of the currently observed data $\mathcal{D}_k$, in that cluster under the assumed prior. For the model being considered, this is given by a multivariate-t distribution with $\nu_n - d + 1$ degrees of freedom,

$$\begin{aligned} p\left(\mathbf{x} \,|\, \mathcal{D}_k\right) &= t_{\nu_n - d + 1}\left(\boldsymbol{\mu}_n, \frac{\Sigma_n^{-1}\left(\kappa_n + 1\right)}{\kappa_n\left(\nu_n - d + 1\right)}\right) \\ &= Z\left(1 + \left(\nu_n - d + 1\right)^{-1}\left(\mathbf{x} - \boldsymbol{\mu}_n\right)^{\mathsf{T}}\left(\frac{\left(\kappa_n\left(\nu_n - d + 1\right)\right)}{\Sigma_n^{-1}\left(\kappa_n + 1\right)}\right)\left(\mathbf{x} - \boldsymbol{\mu}_n\right)\right)^{-(\nu_n - d + 1)/2} \end{aligned} \tag{7.8a}$$

Where,

$$Z = \frac{\Gamma\left(\left(\nu_n + 1\right)/2\right)}{\Gamma\left(\left(\nu_n - d + 1\right)/2\right)\left(\nu_n - d + 1\right)^{d/2}\pi^{d/2}}\left|\frac{\left(\kappa_n\left(\nu_n - d + 1\right)\right)}{\Sigma_n^{-1}\left(\kappa_n + 1\right)}\right|^{1/2} \tag{7.8b}$$

As the degrees of freedom of this distribution increases, it tends towards a Gaussian. Since the student's-t distribution has similar shape to a Gaussian but with heavier tails, this has an interesting interpretation in the clustering model. When clusters have fewer points, a new point which is assessed in the tails of the distribution will have a higher likelihood than if a Gaussian were used. Practically, this will allow small clusters to still accept new points and reduce bias introduced from the small number of points defining the cluster.

Having computed the prior and likelihood for each of the existing clusters in the model, $k = 1, \ldots, K$, the prior and likelihood are calculated to account for the creation of a new cluster $k^\star$. The likelihood is calculated as in Equation (7.8), where the parameters of the t distribution are equal to the prior parameters $\boldsymbol{\beta}$. The prior is calculated as,

$$p\left(c_i = k^\star \,|\, \mathbf{c}_{-i}, \alpha\right) = \frac{\alpha}{N + \alpha - 1} \tag{7.9}$$

Equations (7.4), (7.8) and (7.9) allow the calculation of a value proportional to the posterior likelihood that the data point of interest $\mathbf{x}_i$, was a sample from any existing cluster or a new cluster. These likelihoods need to be scaled by the marginal likelihood, $\sum_{k=1}^{K+1} \tilde{p}\left(c_i = k \mid \mathbf{x_i}, \mathbf{c}_{-i}, X_{-i,k}, \alpha, \boldsymbol{\beta}\right)$, where,

$$\tilde{p}\left(c_i = k \mid \mathbf{x_i}, \mathbf{c}_{-i}, X_{-i,k}, \alpha, \boldsymbol{\beta}\right) = p\left(\mathbf{x_i} \,|\, X_{-i,k}, c_i = k, \boldsymbol{\beta}\right) p\left(c_i = k \,|\, \mathbf{c}_{-i}, \alpha\right) \tag{7.10}$$

Practically, this means summing $\tilde{p}\left(c_i = k \mid \mathbf{x_i}, \mathbf{c}_{-i}, X_{-i,k}, \alpha, \boldsymbol{\beta}\right)$ for every existing cluster and the new cluster ($c_i = k^\star = K{+}1$) and dividing each $\tilde{p}\left(c_i = k \mid \mathbf{x_i}, \mathbf{c}_{-i}, X_{-i,k}, \alpha, \boldsymbol{\beta}\right)$ by this sum. This gives a multinomial distribution for the cluster label c_i of point i.

Sampling a cluster label c_i, from this distribution, the point is assigned to this cluster, either an existing cluster or a new cluster. If the point is added to an existing cluster then the parameters of that cluster are updated according to Equation (7.6). If the point is assigned to a new cluster, that cluster is initialised from the $\mathcal{NIW}$ prior and the single point is added to it according to Equation (7.6). The total number of clusters is also updated to reflect the increase, $K = K + 1$. Once these updates are made, another point is sampled and the process repeats itself.

Since the Gibbs sampler is a valid Markov Chain Monte Carlo (MCMC) method it is guaranteed that the normalised posterior distribution over the cluster labels will converge in the limit to the true posterior conditioned on α and $\boldsymbol{\beta}$ provided that the target distribution of the Markov Chain is that true posterior [76].

7.3 Online Inference in the SHM Context

Using a Gibbs sampling approach to assign cluster labels has a key advantage; each data point is assessed marginally. This means that new data points can be added into the data set and inference can proceed uninterrupted. Since the data in each cluster update the posterior parameters of that cluster, the cluster posterior distributions are refined by increasing the amount of data. The addition of data also allows for the creation of new clusters in a probabilistic (specifically Bayesian) manner without needing to pre-specify the total expected number of clusters or the parameters of those clusters, all without relying on heuristic measures. The learning of the number of clusters is a direct consequence of the Bayesian model form, it does not require expert knowledge or collection of a large training data set.

This behaviour can be exploited for use in an SHM context in three ways:

1. All data observed by a monitoring system refines the parameters of already known states, e.g. the normal condition, thus reducing false alarms.

2. When the behaviour of a structure changes, a new cluster is formed, triggering an alarm.

3. If, upon investigation, first of other available data (i.e. operational and environmental data) and if necessary of the structure itself, this alarm is not a result of damage, the cluster is given a label that allows classification of this separate undamaged state in the future.

This type of semi-supervised methodology allows the model to be continually updated so that all data collected are used to refine the model; this avoids the need to conduct many expensive long-term tests to acquire multiple normal state conditions and to observe the effects of all type of damage. Here, the semi-supervised nature of the process is to label online the discovered clusters from the DP model. The underlying clustering algorithm is considered unsupervised cluster discovery but the addition of labels following investigation is what adds value in an SHM context and this requires online supervision (assigning meaningful labels to those implicitly assigned by the algorithm, this is part of the meta-labelling procedure). It also allows all data collected by the monitoring system to be used as additional information when making inference in the future. Therefore, the value of collecting data increases, as it is not only used for assessment of the structure, but also improves future operation of the SHM system.

It is usual that an SHM system will be operational for an extended period of time, therefore, the size of the training dataset being considered in an *online learning* setting is constantly increasing. This introduces a challenge if the standard Gibbs sampling algorithm for inference in a DP mixture model were to be used. Since the Gibbs sampler would reassess all of the data points (calculating the posterior likelihood of each cluster label), at each iteration the algorithm would become progressively slower, to the point where it would not be feasible to continue. The proposed solution is to window the process so that only the previous o_{max} points that are added to the training set are considered in the Gibbs sampler. The value of this *forgetting factor* should be determined to be the maximum possible, given available computational power, since the early stopping of the Markov Chain may mean that the chain has not converged to the target distribution. It is worth considering that this is the case for all MCMC methods, whose convergence to the stationary distribution is guaranteed in the limit using the Strong Law of Large Numbers. The usual convergence checks for MCMC can be used, such as the $\hat{R}$ statistic [33]; it is recommended, however, that the sampler is run for as many iterations as is computationally feasible. In an online setting, this is limited by the rate at which new data is being added to the process; the algorithm should be able to sample every

Algorithm 8 A Gibbs Sampler for DP Clustering SHM Data with Forgetting and a Gaussian Base Distribution

function DP-FGS($\alpha, \boldsymbol{\mu_0}, \Sigma_0, \kappa_0, \nu_0, o_{max}$)
  $\boldsymbol{\beta} \leftarrow \{\boldsymbol{\mu_0}, \Sigma_0, \kappa_0, \nu_0\}$
  $N \leftarrow 0$ ▷ The Number of Points Observed
  $C \leftarrow 0$ ▷ Start with No Clusters
  for Each New Point Observed **do**
    $N \leftarrow N + 1$
    $o = \max(N - o_{max}, 0)$
    for randperm($i = o$ to N) **do**▷ Random Permutation of Last o Datapoints
      Remove point $\mathbf{x_i}$ from cluster c_i
      Update $\boldsymbol{\mu}_{\mathbf{c_i}}, \Sigma_{c_i}, K$
      for $k = 1$ to K **do**
        Calculate $p(c_i = k|\mathbf{c}_{-i}, X, \boldsymbol{\beta}, \alpha)$ ▷ Predictive Posterior for Each Cluster
      end for
      Calculate $p(c_i = k^\star|\mathbf{c}_{-i}, X, \boldsymbol{\beta}, \alpha)$▷ Predictive Posterior of a New Cluster for $\mathbf{x_i}$
      Sample new c_i from normalised $p(c_i|\mathbf{c}_{-i}, X, \alpha, \boldsymbol{\beta})$
      Add point $\mathbf{x_i}$ to cluster c_i
      Update $\boldsymbol{\mu}_{\mathbf{c_i}}, \Sigma_{c_i}, K$
    end for
  end for
end function

point in the Gibbs sampler at least once between every new reading.

Pseudocode for the algorithm is shown in Algorithm 8, here it can be seen that only data points o_{max} samples back in time are reassessed. This introduces the additional hyperparameter to the model of how far back in time the sampler assesses. This parameter must be chosen *a priori* and is dependent on the system used with regard to the expected rate of change of behaviour, and computational requirements. Once datapoints will no longer be reassessed it is possible to discard them as the information can be contained in the cluster parameters thus leading to a more computationally and memory efficient implementation.

7.3.1 Hyperparameter Selection

In many cases the choices of hyperparameters in the process, including $\boldsymbol{\mu}_0$, Σ_0, ν_0, κ_0, must be driven by prior knowledge of the system which can only come from an

understanding of the structure as an engineering problem; additionally, the available computational resources will govern the range of feasible values.

If there is a case where no training data are available, it can pose problems in setting the hyperparameters for the clusters $\boldsymbol{\beta}$, and also the strength parameter α. In this case, pragmatism must take over. Normalisation of the data would allow the parameters in $\boldsymbol{\beta}$ to be set such that the prior cluster is a zero mean, unit variance Gaussian. It is clearly not possible to perform this normalisation in the absence of any training data. A sensible solution to this would be to implement a standard normalisation scheme, removing the mean and scaling by the standard deviation, where these quantities are calculated based on samples from a fixed period at the beginning of operation, either using the sample statistics, or by bootstrapping [271].

The choice of α poses a more difficult problem. This hyperparameter controls the likelihood that new clusters will be generated. It is not possible *a priori* to choose an optimal value for this parameter, since the spacing of the data in the feature space is unknown. For many applications, there is a sensible range of values from which α can be set. Based on the author's experience, it is recommended that α is set between one and 20 for most applications. Should problems be found with the process in operation, it is possible to repeat the analysis with a different value for α and if desired inference can be performed over α by placing a Gamma prior on the parameter [268].

7.3.2 A Suggested Decision Making Process

The algorithm shown here returns more information than a usual novelty detection scheme due to its ability to cluster recurring feature sets into previously observed behaviour. Outlined here is one way in which this process could be used to aid decision making for SHM, as well as some of the considerations that should be made.

The simplest method to choose as the point at which an alarm is triggered is the creation of a new cluster, which in theory corresponds to the emergence of, as yet, unobserved behaviour. However, as the method progresses and clusters data online, for each assessment in the Gibbs sampler there is a non-zero probability that a new cluster will be created, although this probability can be very small. To protect against an unacceptable rate of false positives, a threshold can be introduced to ensure that alarms are not raised until a number of points are added to a new cluster.

This threshold can be refined over the operation of the system as it does not affect the process of clustering the data itself. As a rule of thumb this can initially be set to be around five points for the critical mass in a cluster; this ensures that the process remains sensitive to changes in behaviour, but protects against small clusters being formed which don't correspond to actual structural changes, but are artefacts of the Gibbs sampler. The value of this threshold does not affect the progression of the algorithm and will likely be specific to individual use cases, its alteration online does not interrupt the algorithm.

A more robust system can be developed working on the assumption that damage causes ongoing changes in the behaviour of the system and that the structure cannot, of itself, return to an undamaged state. The affect of damage will not only cause a new cluster to be formed but points will continue to be added to this cluster as long as the structure is damaged. In view of this, it is possible to use the rate of growth of the clusters as indicators of the structures condition or operating behaviour. If a new cluster is created and grows at a significant rate (the extreme of which being all new points are added to it) this indicates a permanent shift in behaviour which could be associated with damage.

The problem remains of determining whether the change in behaviour is associated with damage to the structure or a change in operation which is ultimately benign. The primary method to separate damage from environmental variation is the choice of appropriate features to cluster [235]. Before the structure is inspected when an alarm is triggered, it is important to use all available data to assess reasons for changes in behaviour. The obvious suspects would be changes in environmental conditions: temperature, precipitation, etc. Other factors which will strongly influence the operational behaviour will include changes in use of the structure, such as change in loading or in the structural properties (e.g. changing topside mass on an offshore platform). It is worth considering at this point, the difference between observing a correlation with another measured variable related to the environmental conditions and establishing causation before deciding that the cause of a new cluster is benign. Discussion regarding this point can be found in [272]. Methods such as the Granger test [273] may help to provide insight as to whether a cause of the change in behaviour can be explained by other measured data.

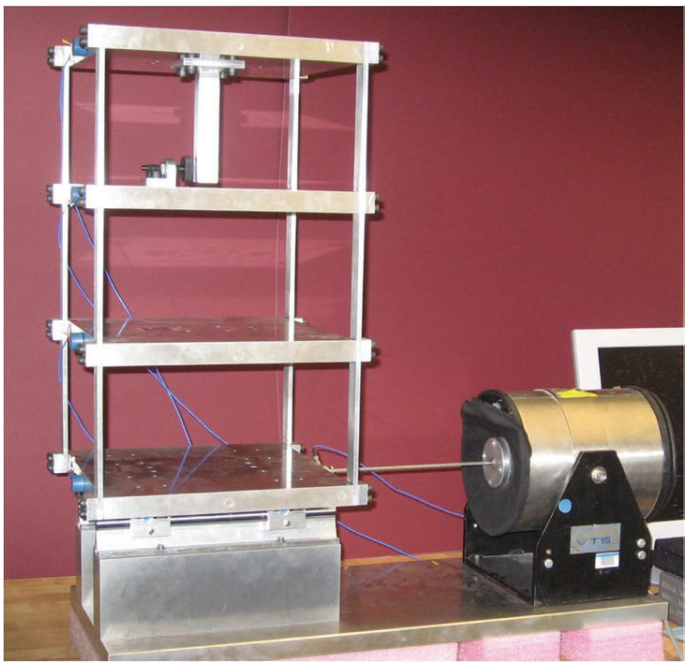


Figure 7.3: Image showing setup of the three storey building structure, image reproduced from [274]

7.4 Results

7.4.1 Three-Storey Building Structure

The application of the DP mixture model is explored here using a benchmark dataset from a three-storey building structure (Figure 7.3), produced by Los Alamos National Laboratory [274] for identification of damage under changing system behaviour. The experiment is a simplified three-storey building structure undergoing base excitation. Damage is simulated using a bumper attached between the second and third floors with the aim of representing a breathing crack in the structure. The structure is excited, nominally, along one axis only and mounted on linear bearings to minimise any torsional behaviour. The structure was tested in 17 states, aiming to represent a mixture of damaged and undamaged conditions, a summary of these states is shown in Table 7.1.

In the original report [274], a number of methods for feature extraction and classification are discussed and shown to be effective at detecting damage on this structure. Zhou *et al.* [275] show the use of an output-only approach to detect the introduction of damage. Figueiredo *et al.* [276] discusses the selection of AR model order in

Label	State Condition	Description
State#1	Undamaged	Baseline condition
State#2	Undamaged	Added mass (1.2kg) at the base
State#3	Undamaged	Added mass (1.2kg) on the 1st floor
State#4	Undamaged	Stiffness reduction in column 1BD
State#5	Undamaged	Stiffness reduction in column 1AD and 1BD
State#6	Undamaged	Stiffness reduction in column 2BD
State#7	Undamaged	Stiffness reduction in column 2AD and 2BD
State#8	Undamaged	Stiffness reduction in column 3BD
State#9	Undamaged	Stiffness reduction in column 3AD and 3BD
State#10	Damaged	Gap (0.20 mm)
State#11	Damaged	Gap (0.15 mm)
State#12	Damaged	Gap (0.13 mm)
State#13	Damaged	Gap (0.10 mm)
State#14	Damaged	Gap (0.05 mm)
State#15	Damaged	Gap (0.20 mm) and mass (1.2kg) at the base
State#16	Damaged	Gap (0.20 mm) and mass (1.2kg) on the 1st floor
State#17	Damaged	Gap (0.10 mm) and mass (1.2kg) on the 1st floor

Table 7.1: Table reproduced from [274] showing 17 different states under which the structure was tested.

the context of damage detection on this structure, using the time series collected; whereas, Bandara *et al.* [277] show how frequency domain features (PCA projections of the FRF and coherence) can be used in the feature selection step and provide good classification results when used as inputs to a neural network.

The states where the changes made to the structure did not introduce nonlinearity (mass or stiffness changes) are considered to be environmental variation, and those which introduced nonlinearity (impacts of the bumper) are considered damage states. It can be seen that, in addition to the baseline condition, there are eight states representing environmental changes and eight representing damage.

50 measurements were made in every state, each comprising of a time series of 8192 data points, corresponding to 25.6 seconds of data. Frequency domain features are extracted from the data. It is important here that the clustering algorithm is also sensitive to features which can be extracted online. Although this limitation is minor, it does require some consideration when designing the identification algorithm.

Prior to the implementation of an SHM system, the use of such a system is justified, and the design of the system must be informed by operational evaluation [6]. This

process considers the added benefit of investing in SHM; it also defines the parameters under which the system operates. These include considering the conditions in which the structure will operate, and the effect of this on any data acquisition scheme. A key step in SHM is feature selection; the challenge in this case is that many of the usual tools for feature selection are unavailable due to the lack of a training phase. It is necessary, therefore, to design the feature selection in such a way that it can: firstly, be computed online for all data that will be collected by the system; secondly, will give rise to features that are sensitive to changes in the structure that are of interest. In general this will be sensitivity to damage in the structure but not to environmental conditions. As is usual when dealing with measurements of acceleration of a dynamical system, data is first transformed into the frequency domain in batch. For vibration data, damage sensitive features are predominantly extracted in the frequency domain [278, 279]. The additional benefit of using frequency domain features is that they can be invariant to the input to the system, e.g. the natural frequency (of a linear structure) is not affected by the forcing on the structure. This plays some role in the removal of environmental and operational changes.

Transformation of the blocks of 8192 time points into the frequency domain gives feature vectors which are 1024-dimensional real values in the Power Spectral Density (PSD), using Welch's method [280]. This high dimensionality is a significant hindrance to many algorithms, including the one presented in this paper. Not only does it add significant computational burden, in this case $\mathcal{O}(D^3)$ for D dimensions, this complexity comes from the inversion of the covariance matrices which are size $D \times D$. But also many algorithms suffer from lack of sensitivity in high-dimensional spaces due to reliance on Euclidean distance metrics [179, 281]. To avoid this it is possible to only consider other features which summarise the key properties of these high dimensional features, e.g. the natural frequencies and damping ratios of a system. However, a significant amount of information is lost when only these simple quantities are considered. It is desirable, therefore, to retain as much information as possible while also reducing the dimensionality of the feature space.

The usual manner to deal with this high dimensionality is to perform some type of dimensionality reduction such as Principal Component Analysis [75]. Principal Component Analysis (PCA), among other dimensionality reduction techniques, requires a representative training set of data which can be used to learn a linear projection onto a lower-dimensional space by accounting for maximum variance in each direction as the dimensionality increases. When designing an online SHM

system, this does not represent a feasible approach since data are required to learn the optimal projection prior to any analysis using PCA, via the expectation maximisation method. The use of an online PCA projection — for example via online expectation maximisation [282] — also causes problems since the projection into the low dimensional space would be changing online; requiring the algorithm to fully recompute at each time step (running the Gibbs sampler multiple times to ensure convergence) which is not computationally feasible.

An alternative approach is to leverage a technique that has found widespread use in the compressive sensing community [283] — Random Projection (RP). The Johnson-Lindenstrauss theorem states that, when a set of high-dimensional data in Euclidean space is projected using a random matrix, the pairwise distances between the data are preserved with an error that can be quantified, allowing signals to be significantly compressed using RP [284, 285]. By adopting a dimensionality reduction technique, which, rather than manually selecting features, does not require expert knowledge or a representative training set offers a number of advantages. The foremost of which (in this case) is the ability to begin operation of the SHM system immediately without a training phase and while preserving the pairwise distances between the full magnitude FRFs/coherences. In this way more information can be retained as opposed to the selection of some other low dimensional feature e.g. modal properties.

For this dataset, initially, the FRF and coherence at the top floor are considered; each of these is projected down onto ten dimensions using a random projection, where each element of the random matrix is an i.i.d. sample from the distribution $\mathcal{N}(0,1)$. These features are augmented with the area under the magnitude FRF at each floor including the base, giving a 24-dimensional feature vector. The addition of this feature is to capture the change in total energy being transferred to each floor as the structure state changes.

The algorithm was run with the parameters set as: $\alpha = 10$, $o_{max} = 200$, α is chosen by engineering judgement before looking at the data (although varying this parameter is explored later) and o_{max} is limited by computation speed; therefore the Gibbs sampler reassessed only the previous 200 points, to save on computational burden. Figure 7.4 shows the progression of the algorithm over time with each observation being a block of 8192 time series points from which the features are extracted. Vertical lines show the initiation of a new cluster at which point an intervention is triggered to label the newly-observed behaviour. By studying Figure 7.4, one can see that 16 clusters have been detected. The damage introduced at observation 450, immediately triggers an

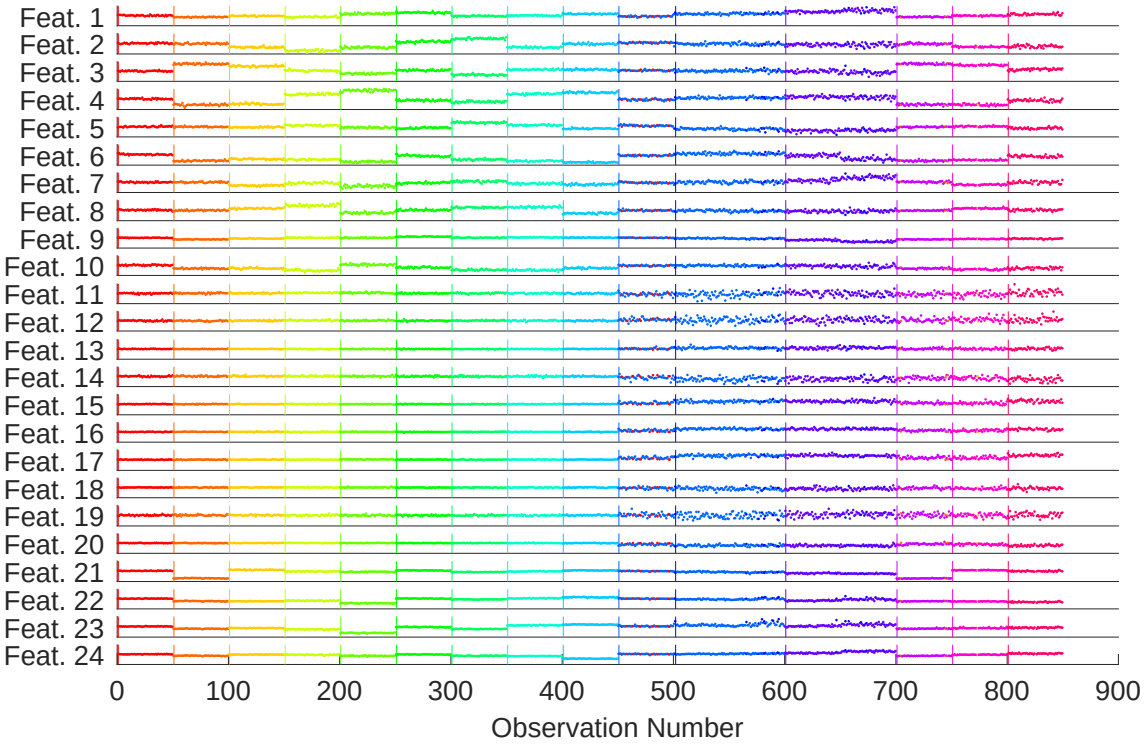


Figure 7.4: Plot showing features used in clustering with colours indicating the clusters to which datapoints have been assigned for the online Dirichlet Process clustering with the full 24 dimensional feature space. The vertical lines of each colour indicate the initiation of that cluster.

intervention as a new cluster is formed.

Figure 7.5 shows the confusion matix between the implicit states from the DP clustering and the known true states. For the initial nine states, baseline and eight environmental changes, there is perfect classification using the online DP clustering, the 9×9 matrix in the upper left is diagonal. This shows that while the algorithm would require further investigation when there is a change in environmental behaviour, the reappearance of these changes would then be correctly classified. For example, if there were seasonal changes in behaviour these would be classified correctly after the first appearance of the behaviour.

States 10 to 14 in the dataset correspond to increasing damage severity. It can be seen that for the smallest damage extent, despite triggering at the first damage observation, there is some confusion with the baseline state. Given these fifty observations it suggests that damage is occurring while the structure is operating under environmental conditions equivalent to State 1. As the severity of damage increases, the states are correctly classified into one of three clusters. State 15

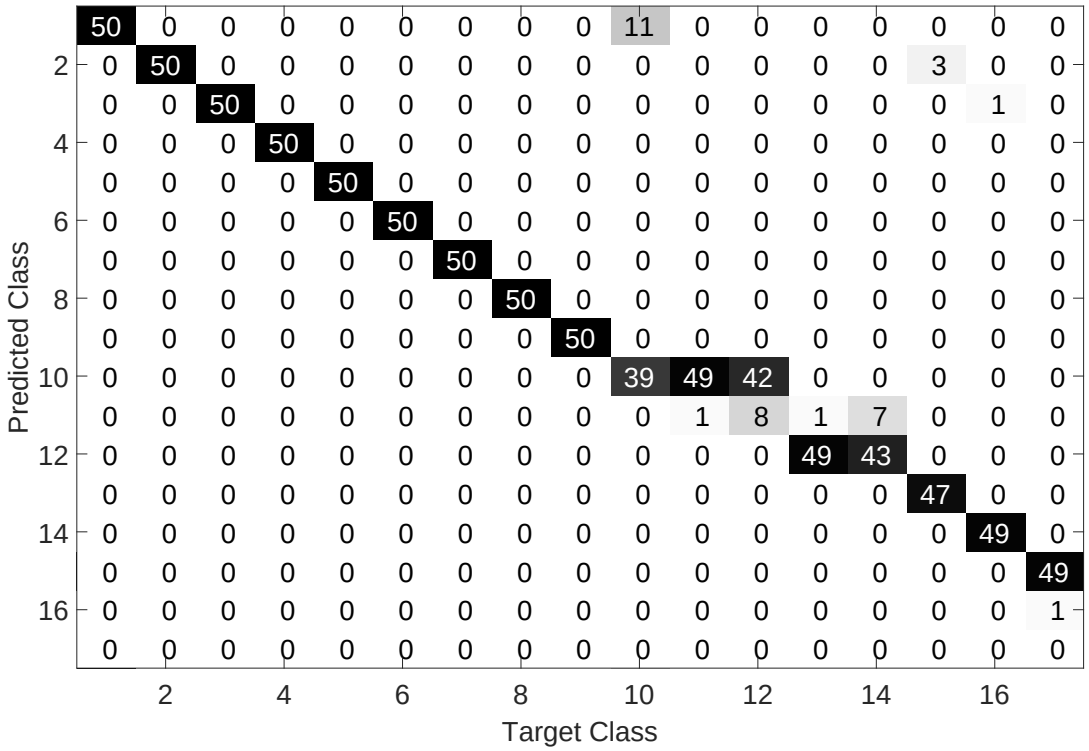


Figure 7.5: Figure showing the confusion matrix for the implied states (cluster indices) from the online DP clustering when compared to the 17 known states for which the structure is tested using the full 24 dimensional feature space.

corresponds to the lowest damage extent with the environmental change from state 2, which is classified well as a new damage case with only a small number of misclassifications into state 2. State 16 is equivalent to 15 except the environmental change is that seen in state 3, with similar results. State 17 corresponds to a larger damage extent with the environmental change from state 3. This is well classified as a new damage class.

It is useful, however, to consider how varying the alpha parameter would affect the results shown for this case. For this reason the algorithm was additionally run with a number of different α values. If the system were running offline, inference could be performed over α to either select an optimal value or to learn the distribution in a Bayesian manner. Instead, the algorithm has been run with ten different fixed α values for a hundred different runs. Since the algorithm is stochastic, it is important to consider the distributions at different α values, not just a single result.

Figure 7.6 shows the development of the false negative (FN) rate for increasing α. The boxplot shows the 25th and 75th percentiles as the top and bottom of each box,

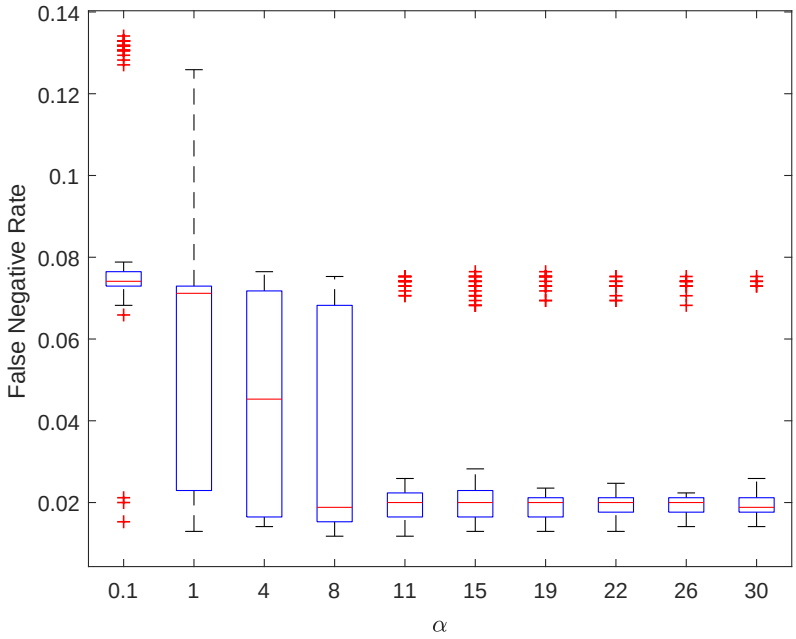


Figure 7.6: Boxplot showing distribution of False Negative rates for 100 runs at the given levels of α

the sample median is shown by the red line. The "whiskers" show the interval of $+/-2.7\sigma$ and outliers from this range are denoted by red crosses. The FN rate is defined here as the number of points in damage classes classified into an undamaged class. As seen in Figures 7.4 and 7.5, for the progressing damage scenarios in States 10 to 14, three clusters are created; distinction between these clusters is not included in the calculation of the FN or false positive (FP) rate. The FP rate was zero for all tests across all values of α, where an FP was defined as a point being classified into a cluster greater than 9 if it was in one of the first 9 states. For the results shown in Figure 7.6, it can be seen that the FN rate is low across all levels of α. There is an increase in the FN rate as α tends to zero which is associated with data in the lowest damage extents, States 10 and 15, being misclassified as belonging to the healthy clusters associated with those environmental conditions. As the α value increases past 10, the FN rate has little variation with α, as the clusters are well separated; this stops the formation of more clusters. Figure 7.7 shows this in a box plot where the distributions in the number of clusters are considered with varying α. The fact that the clusters are well separated means that the number of clusters plateaus in this range of α values.

For the feature set shown in this experiment, which gives relatively good separation

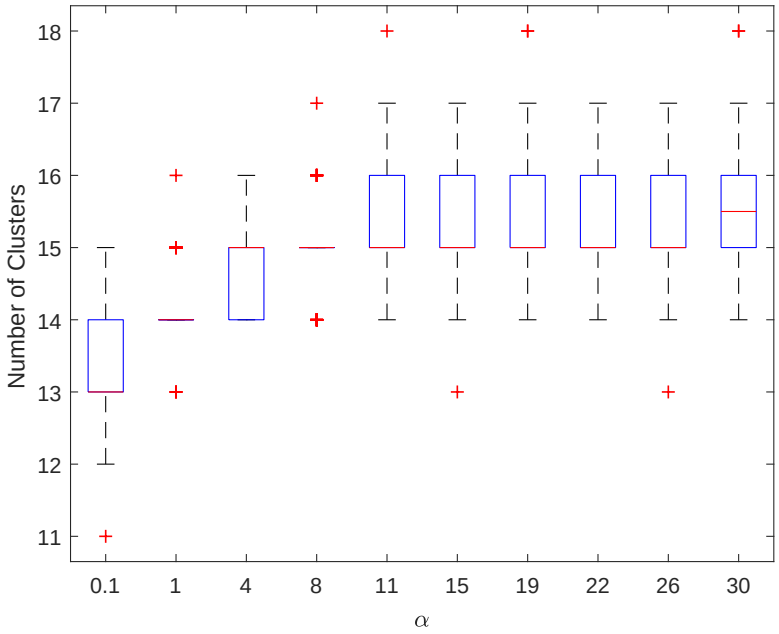


Figure 7.7: Boxplot showing distribution of the number of clusters created for 100 runs at the given levels of α

of clusters, the performance of the process is not significantly impacted by the choice of alpha within the range $\alpha \in [0.1, 30]$. This supports the *a priori* selection of $\alpha = 10$ as a starting point in engineering problems, where the data can be normalised to zero mean and unit variance and the parameters of the $\mathcal{NIW}$ prior are set $\boldsymbol{\mu_0} = \mathbf{0}, \Sigma_0 = \mathbb{I}, \nu_0 = D, \kappa_0 = 1$ which corresponds to a unit Gaussian in D dimensions as a prior.

Feature Selection to Remove Sensitivity to Environmental Changes

Should one wish to build a damage detection system that is insensitive to changes in the environmental conditions, it is possible to omit the features that are sensitive to this and perform the same inference procedure on a reduced feature set. The algorithm is re-run with a reduced feature set, where features are only sensitive to the damage condition not the environmental changes, with the same parameters as the previous analysis. This follows from the feature selection methodology shown in [235]; however, in the case of *online learning* these features must be chosen *a priori* based on engineering judgement.

Figure 7.8 shows the confusion matrix when only 10 features which are damage-

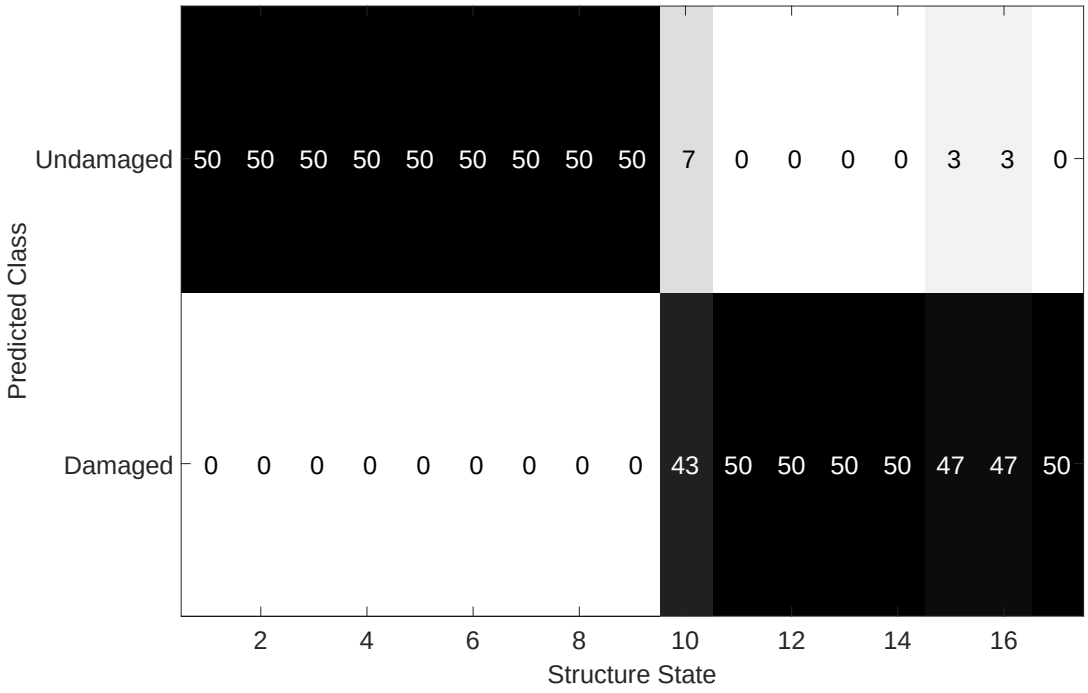


Figure 7.8: Figure showing the confusion matrix for the implied states (cluster indices) from the online DP clustering with the reduced feature space (10 features). This compared to the 17 known states of the structure, states 1-9 correspond to undamaged behaviour and 10-17 correspond to damage, as is shown in Table 7.1.

sensitive are used to perform the clustering. The algorithm is attempting to separate only the damaged and undamaged classes from the dataset as defined in column two of Table 7.1, where states one to nine are classified as undamaged and 10 to 17 as damaged. These ten features are chosen to be the randomly projected coherence of the top floor. This feature selection does not require the damage state data. It is intuitive that, since the system is designed to detect a breathing crack in a structure that is approximately linear, the damage will increase any nonlinear behaviour which will cause significant change in the coherence but not in the FRF [286]. The coherence should also be broadly insensitive to the environmental changes that are expected to occur.

The DP clustering algorithm creates only two clusters in this case, without any tweaking of the hyperparameters. These two implicit states, upon inspection, correspond to the undamaged and damaged states. If only considering whether the system is labelled damaged or undamaged (Table 7.1), there are no false positives and 14 false negatives across the dataset of 850 observations, a FN rate of 0.017, defined as before. These results correspond to a sensitivity of 1 and specificity of

0.965 giving a total accuracy of 0.984. All of the false negatives occur at the lowest damage state (0.20mm Gap), under differing environmental conditions. Despite this misclassification, the algorithm would raise a suitable alarm, even at the smallest damage extent, triggering an intervention.

The behaviour shown in the two cases above clearly demonstrates the ability of the algorithm to detect unknown states and to create new clusters to accommodate them. It also reveals that this does not remove the need for intelligent feature extraction based on sound engineering judgement. It is possible, given sufficient physical understanding of the structure, to imagine features *a priori* that will only be sensitive to changes of interest (e.g. insensitive to environmental conditions), and with the use of techniques such as RP to create feature spaces upon which the algorithm can operate. The choice of these features must be driven by engineering knowledge, in this case the assumption that a system whose behaviour is close to linear when undamaged will become more nonlinear with progressing damage but not with environmental changes [286].

Operating Online Without Input Information

In operation, an SHM system does not normally have access to measurement of the excitation source, as with a system tested under laboratory conditions. This is normally due to the difficulty in placing instrumentation in the load path of the structure, both practically and financially. In this case, features based on the FRF or coherence function become inaccessible due to their reliance on data regarding the forcing of the system — this would be another possible motivation for the work presented in previous chapters of this thesis.

It is desirable, therefore, to imagine a situation in which the proposed method would be applied on a dataset where this information is unavailable, the aim being to create a semi-supervised learning algorithm that is sensitive to damage on the structure. Again, using the intuition that the presence of damage on the structure will lead to increased nonlinearity in the structure [286], it is possible to determine a feature set that will be sensitive to damage; it is assumed here that the measurements of acceleration at all three floors are available, but not the forcing at the base.

In the same manner as before, the data arriving in windows of 8192 points can be converted into power spectra in the frequency domain with 1024 features. Operating

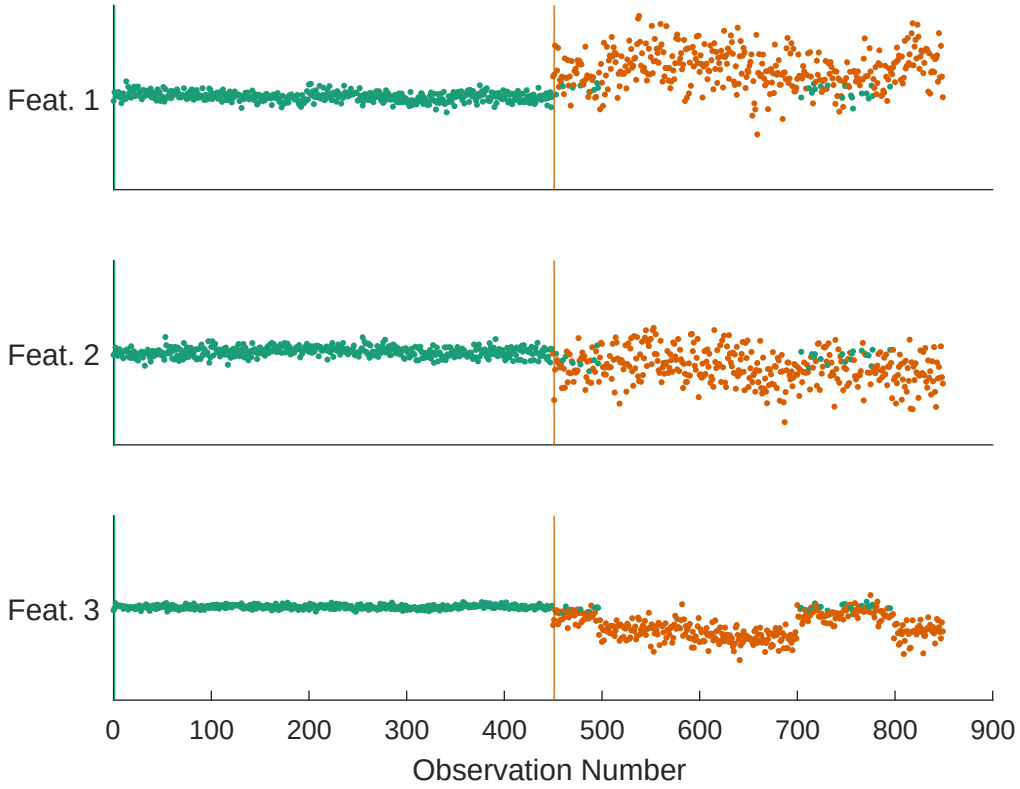


Figure 7.9: Figure showing operation of the algorithm on the three dimension feature space created by randomly projecting the coherence between the ground floor and floor three.

directly on these power spectra will not yield a high sensitivity to damage and will be sensitive to changing environmental conditions. It is possible, therefore, to calculate the coherence between two of these output spectra rather than the traditional input-output coherence. This approach has been explored in [275], although here further signal processing is applied to create a damage-sensitive index based on the sum of the coherence functions. This approach requires offline learning to set up a statistical control chart on this feature, a step that is not required in the current work.

This coherence between the two output spectra can be reduced in dimension in the same manner as previously, using RP, since using all spectral lines naïvely is not feasible computationally. The new algorithm here is tested using the projection of the coherence between the ground floor and the top floor (data channels 2 and 5) onto only three dimensions.

Figures 7.9 and 7.10 show the progression of the algorithm in time, and the confusion

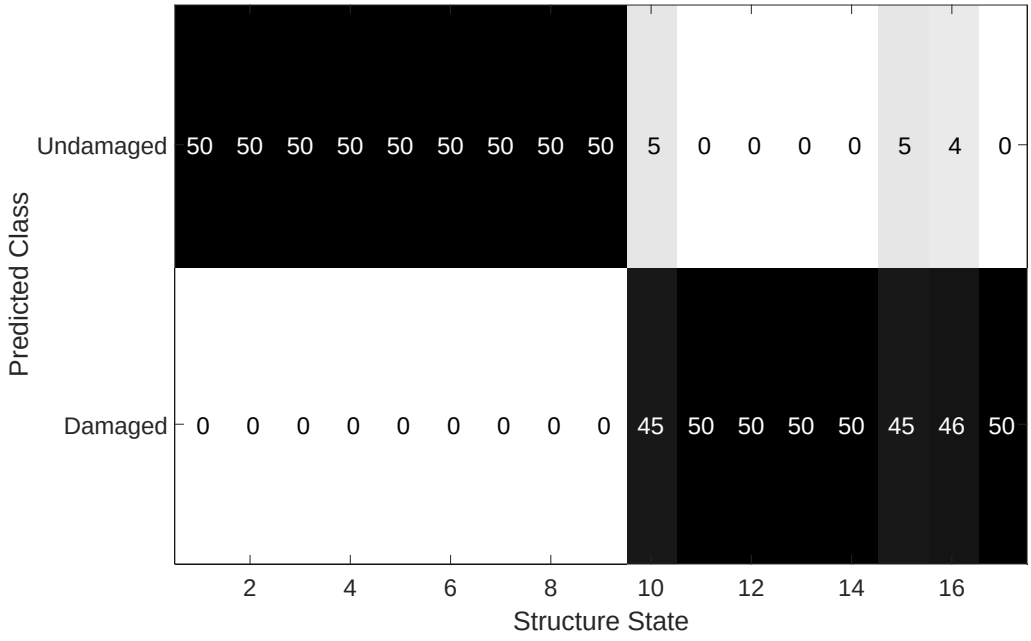


Figure 7.10: Figure showing the confusion matrix for the implied states (cluster indices) from the online DP clustering with the randomly projected output coherence features. As in Figure 7.8, the algorithm is attempting to classify between undamaged states(1-9) and damaged states (10-17) — see Table 7.1.

matrix when this limited feature set is used. The performance is comparable to when the input information is also available to the algorithm. It is clear that this methodology is capable of detecting changes in behaviour associated with damage in an online semi-supervised manner (in the absence of training data at the start of the process), which is efficient in terms of memory and data storage requirements. The information returned from the method regarding the creation of new clusters is a simple trigger for intervention from engineers operating the system. In addition, as the system continues to collect data, its ability to correctly classify new data is enhanced as the cluster parameters are refined in a Bayesian manner.

For this alternate feature set, the results shown in Figures 7.9 and 7.10 are for an α value of 0.1. The results were run for a range of α values as before; however, for this experiment significant dependence on the choice of α was seen for the number of clusters created. This is shown in Figure 7.11. To calculate the FN and FP, rates the problem was treated as a two class problem where, any points in cluster 1 were considered to be "normal" and points in any other cluster were considered "abnormal". This prescription leads to the true classes for each data point to be given

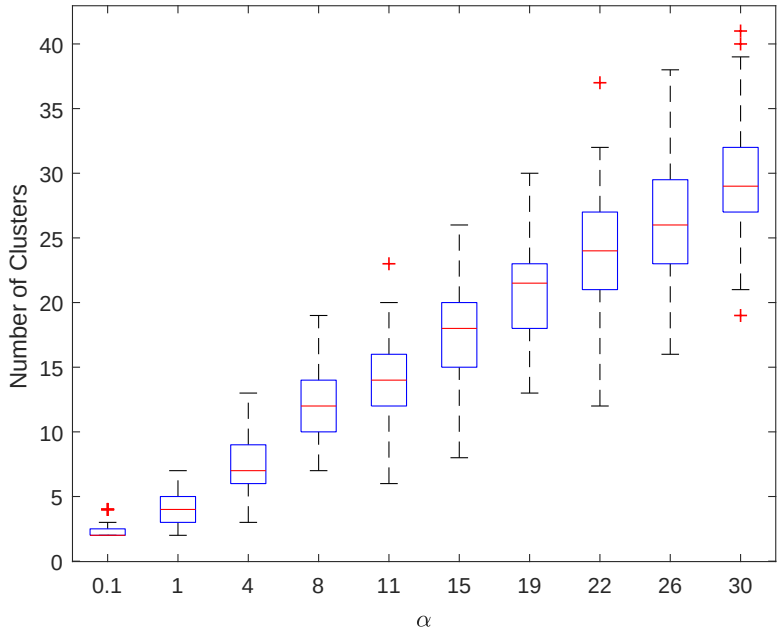


Figure 7.11: Boxplot showing distribution of the number of clusters created for 100 runs at the given levels of α

by the State Condition column in Table 7.1. Box plots of the FN and FP rates for one hundred repeats at each α are shown in Figures 7.12 and 7.13, both the FN and FP rates are very low for all values of α. The trend shown in Figures 7.12 and 7.13 is a decrease in FN and increase in FP with increasing α. This is expected since the α parameter encodes the prior belief that data will be drawn from new clusters; intuitively, this states that with a given value of α a new cluster is just as likely as a cluster with α points in it already, see Equations (7.4) and (7.9).

Shown in Figure 7.14 is the corresponding plot to Figure 7.9 for a randomly chosen run of the algorithm with $\alpha = 30$; it can be seen that the FP rate is very low with only a single point misclassified, but as soon as the structure has damage introduced (point 450) multiple new clusters are created very quickly. To understand this behaviour, it is helpful to consider the pairwise correlation plots in Figure 7.15. Since the data have been normalised online using the first fifty points, and the hyperparameters of the $\mathcal{NIW}$ prior are set to $\boldsymbol{\mu_0} = \mathbf{0}, \Sigma_0 = \mathbb{I}, \nu_0 = D, \kappa_0 = 1$ as before; the increase in variance seen with the initiation of damage on the structure, and lack of separability of the clusters, leads to the creation of many new clusters (in this run 11 clusters in total). In other words, the prior encourages the process to make a mixture of unit variance Gaussian clusters, based on the normalised data. As damage progresses,

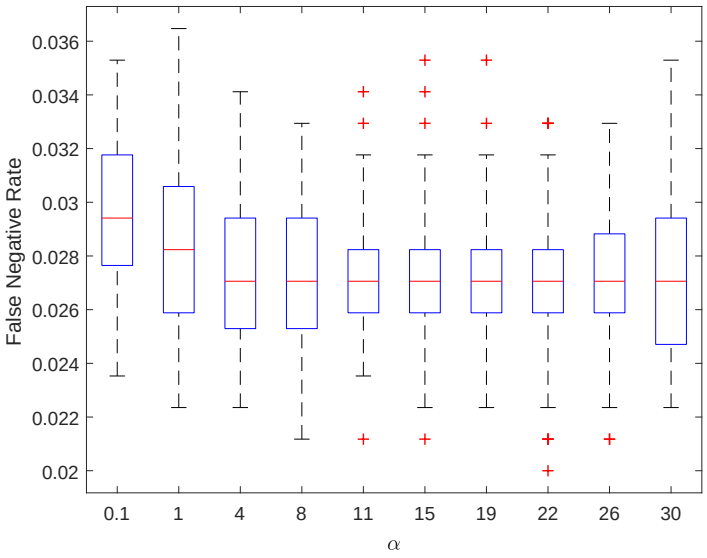


Figure 7.12: Boxplot showing distribution of False Negative rates for 100 runs at the given levels of α when using the output only features

the variance in the features increases despite the data being normalised to the lower variance portion of the signal. The process is, therefore, more likely to create a number of smaller clusters in the cloud of higher variance data instead of a single higher variance cluster. This effect is exacerbated by the higher α value, which favours the creation of more smaller clusters as the value increases.

The key question which must be asked is: in what way will this affect the operation of a system using this technique for SHM? The system remains resilient to false positives, and as discussed previously, techniques can be used to increase robustness to these. At the initiation of damage in the dataset a large number of new clusters are created which would lead to investigation, as discussed of other available environmental and operational data. The high number of alarms triggered would indicate a significant change in the structure, which in this case clearly corresponds to the damage being introduced.

7.4.2 Z24 Bridge Data

The now widely-known Z24 bridge dataset [56], has become a test-bed for many damage detection algorithms in SHM, particularly SHM of civil infrastructure. The dataset comprises of roughly one year of monitoring data from a bridge in Switzer-

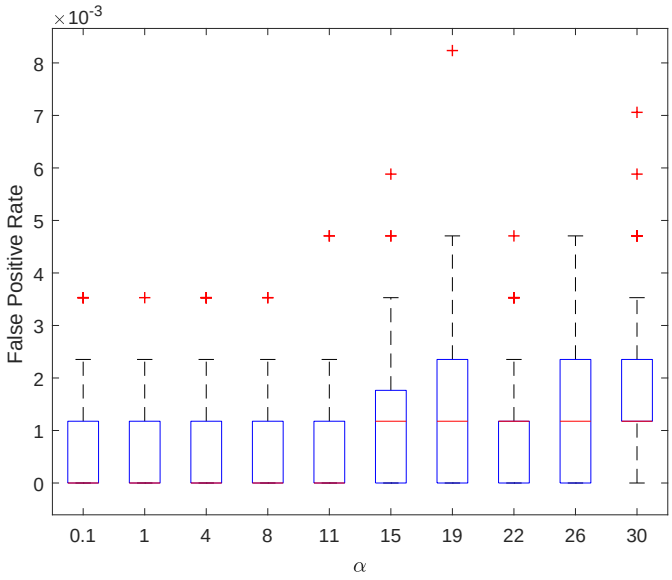


Figure 7.13: Boxplot showing distribution of False Positive rates for 100 runs at the given levels of α when using the output only features

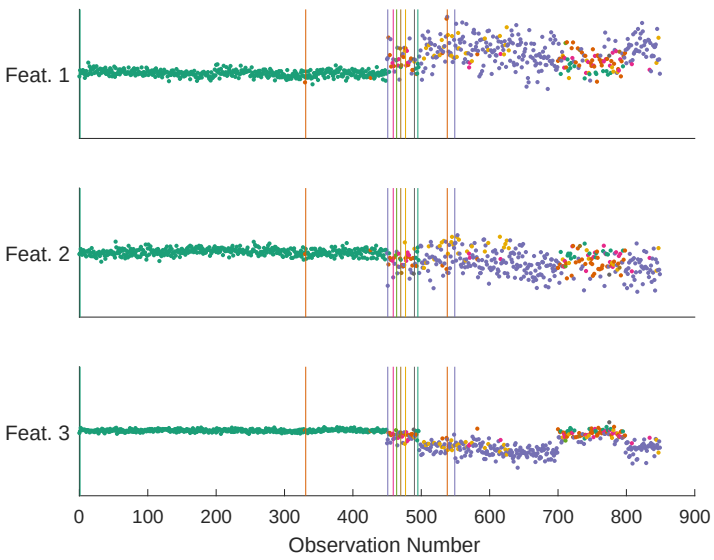


Figure 7.14: Figure showing progression of the algorithm when $\alpha = 30$, vertical lines represent the initiation of a new cluster

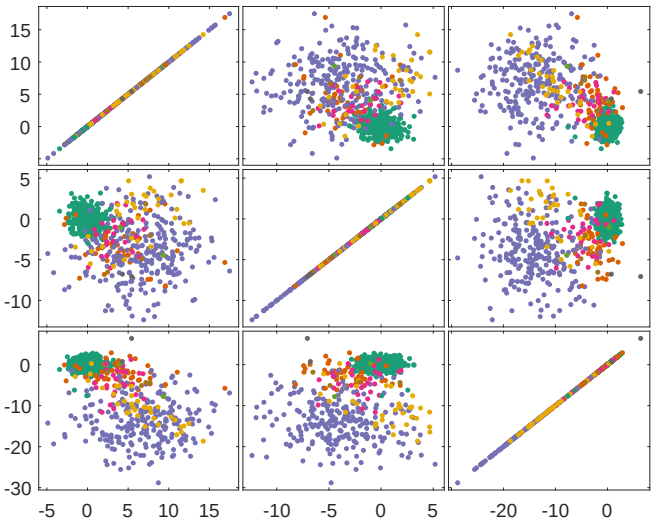


Figure 7.15: Pairwise correlation plots for the process when $\alpha = 30$, where the colours shown correspond to those in Figure 7.14

land where damage was introduced deliberately toward the end of the monitoring programme. Researchers have most commonly used the first four natural frequencies of the bridge deck as damage-sensitive features; the difficulty in the dataset arises from the changes in environmental conditions which can confound damage detection algorithms. The most significant change is when a reduction in temperature is hypothesised to have caused stiffening of the deck asphalt leading to a rise in natural frequencies.

This work makes no attempt to avoid these changes in behaviour due to environmental effects, instead it aims to demonstrate the ability of DP-based clustering to detect and subsequently classify different regimes of the structure. The data are tested with the parameters of the algorithm set as $o_{max} = 2000$ and $\alpha = 10$. Here, again, o_{max} is set on the basis of available computation time which is greater given the slower rate of arrival of the data points. α is set as before. Additionally to this, a threshold is introduced as discussed, to protect against false positives; this is required in this dataset due to the increased noise experienced in the full-scale test as opposed to the laboratory setting. The threshold was set at 50 data points; this was tuned on the basis of results from the initial section of the dataset, 500 data points. As previously mentioned, it may be possible to set a more robust trigger based on the rate of growth of the clusters, which may well constitute further work.

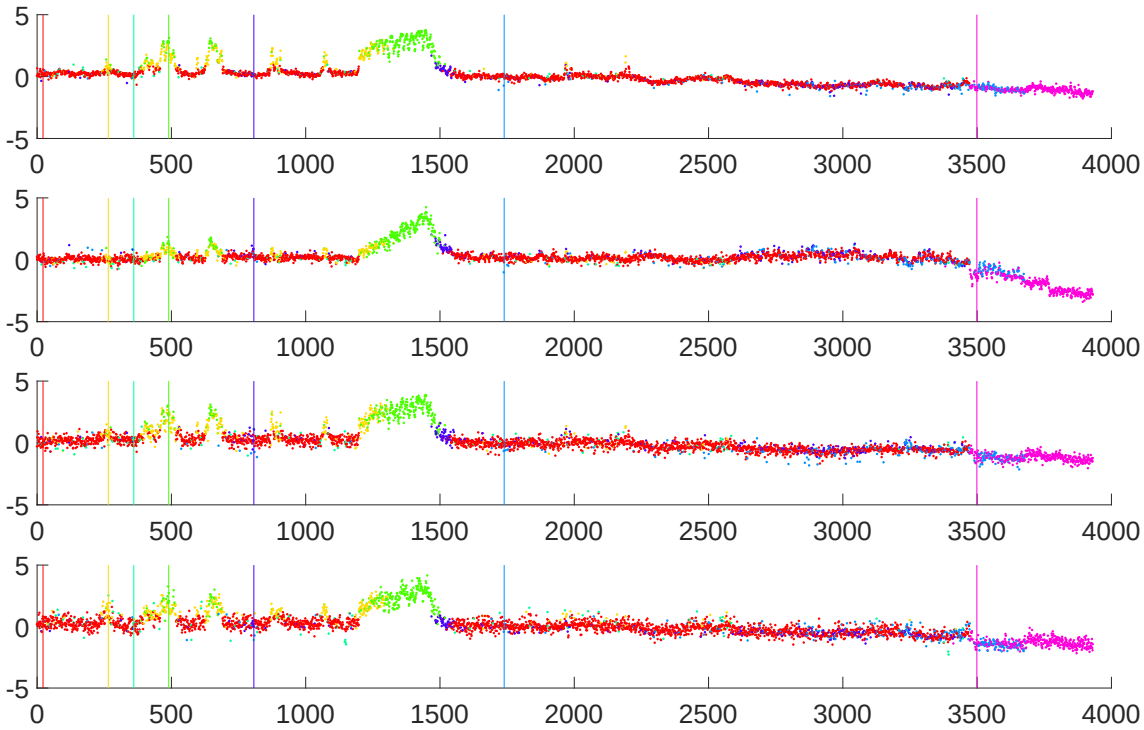


Figure 7.16: Figure showing online DP clustering applied to the Z24 bridge data using the first four natural frequencies as the features.

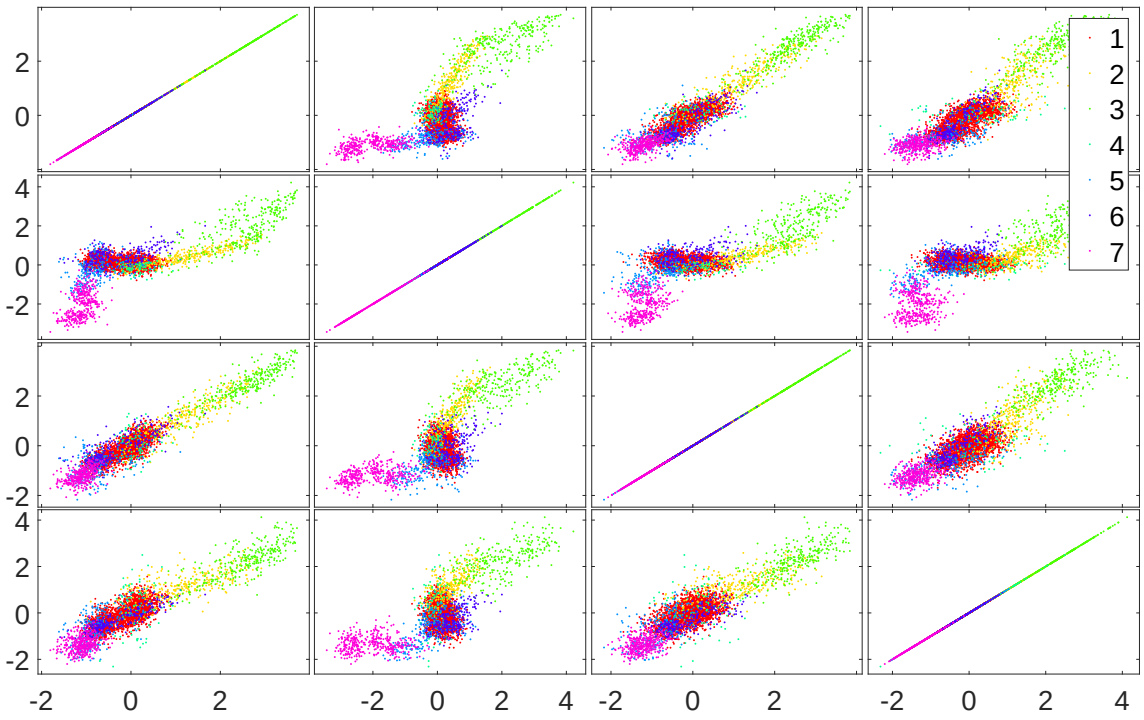


Figure 7.17: The clusters found for the Z24 bridge by the online DP clustering are shown in the feature space.

In the same manner as before, it is assumed that there is minimal training data available, only the first 500 data points. As the algorithm progresses, more clusters are created; this is shown in Figure 7.16, a normal condition cluster (red) is quickly established. As the temperature cools three more cluster are created (orange, cyan and green) corresponding to the progression of freezing of the deck. Two other clusters are created, the dark blue one around time point 800 and the light blue one close to time point 1700. From inspection of the pairwise plots of each variable (Figure 7.17) it appears that this light blue cluster corresponds to a shift and rotation in the normal condition. This could be caused by long term drift in the normal condition which leads the distribution of points in this state to become non-Gaussian, possibly another affect of the varying ambient temperature, precipitating the creation of a second cluster to approximate the non-Gaussian distribution. Finally, the pink cluster is created only two data points after damage is introduced to the structure showing the method's ability, given the available feature set, to detect a change in behaviour corresponding to damage. In the Z24 dataset, there are two damage states induced, however, these are both classified into the same cluster when the DPGMM is run online. There are two reasons for this behaviour. The first is due to the lack of separation between the two damage state clusters in the feature space and the choice of α as shown on the three-storey bookshelf data. The second is that the data is normalised to the initial 500 points of data. The variance observed in this phase from which the hyperparameters of the cluster shape are set is greater that then separation between the two different damage clusters. This makes it difficult for the algorithm to create a new cluster for the second damage state due to the prior belief that has been encoded in $\boldsymbol{\beta} = \{\boldsymbol{\mu}_0, \Sigma_0, \kappa_0, \nu_0\}$.

It can be seen that once the algorithm has an explicit label assigned to the implicit label from the cluster assignment, subsequent data falling in that cluster can be correctly classified. The clusters relating to the different stiffness conditions of the deck, are able to classify these events from the second occurrence onwards, avoiding the need for unnecessary interventions, as would be the case with a simple novelty detection method. Figure 7.17 shows the pairwise correlations of the first four natural frequencies of the Z24 bridge; it is in this feature space that the algorithm is operating. Here, it is clearer how the clustering algorithm is separating the feature space into a mixture of Gaussian distributions.

The results shown on the Z24 dataset, demonstrate the ability of the algorithm to deal with recurring environmental conditions while remaining sensitive to damage. It

also makes clear that this approach to a damage identification algorithm will require more interventions/inspections shortly after the installation of the SHM system but with robustness increasing over time.

7.5 Discussion

This chapter has introduced a methodology for incorporating a DP mixture model into an SHM system for online damage detection. The algorithm has been shown to perform very well on test data with multiple damaged and undamaged states. The method requires little user input and updates online with simple feedback to the user as to when intervention is required. Additionally, as clusters are assigned physically meaningful labels, additional information is available to the end user. It is believed that the method provides a promising approach for SHM when there is little or no availability of training data and inspections are possible to assign labels in a semi-supervised manner. There are a number of strengths to using this technique over a simple one-class novelty detector or a non-probabilistic method such as affinity propagation [249]. The algorithm, unlike a basic novelty detector, can be run in a semi-supervised manner to assign labels to new behaviour states online and create new clusters. This additionally allows for multi-class classification as the algorithm progresses, allowing movement up the Rytter damage hierarchy [7], as more information is uncovered, toward classification or location. The advantage of this over moving from a novelty detector to classifier online is that there is no further training phase required and the algorithm automatically incorporates both the classification and novelty detection. Unlike methods such as affinity propagation, the DP clustering algorithm has a strong Bayesian foundation. By using a Bayesian technique, not only is a probability for every cluster provided, but there is a rigorous framework for the incorporation of prior knowledge. This would allow the use of an incompletely labelled training dataset to initiate the algorithm if certain states are known at the outset. This is achieved by assigning data points to clusters to update the cluster parameters, then excluding these points from the Gibbs sampling procedure to fix those clusters as explicit priors. If new points are added to these clusters, the parameters can continue to be updated via the conjugate update steps.

A modification to the normal DP algorithm has been proposed where the Gibbs sampler is truncated to consider only the previous $o_{\max}$ points in time. This allows

the methodology to be applied online by stopping the computational complexity growing as more data are acquired. This limits the complexity at each step to be, naïvely, $\mathcal{O}\left(KD^3 o_{\max}\right)$ which is possible to compute online. However, it can be formulated such that the clusters undergo a rank one update to the covariance at each step which reduces the complexity to be $\mathcal{O}\left(KD^2 o_{\max}\right)$ on all but the first time step.

Another key advantage of the method is that, once the $\mathcal{NIW}$ hyperparameters have been set, there are only user-tunable hyperparameters, α and $o_{\max}$. If necessary, full Bayesian inference can be performed by placing a prior over the α parameter and performing inference, for example, via MCMC. The sensitivity of the process to this parameter has been discussed in terms of the affect on feature selection and normalisation. It has been shown, however, that problems which may occur from poor selection of this parameter are minimal, especially when clusters are well separated. Finally, it is possible to formulate the problem with a non-Gaussian base distribution, if the data are believed to be significantly non-Gaussian. It is worth considering whether this adds value to the inference procedure since computation time is severely increased in this case and many non-Gaussian datasets can be well represented by the Gaussian mixture model, especially when the number of mixtures does not need to be specified *a priori*.

It is noted that, although this highly flexible model has a benefit when data arrive online with an unknown number of states, there may be better tools to use in an offline state or if the problem is restricted to detection. It would be surprising if this semi-supervised method was able to compete with a fully supervised inference algorithm, since there is less information in the training phase. Although the method has been shown to work on a two-class novelty detection problem (Figure 7.8) it is expected that other methods (e.g. robust outlier detection [17]) would perform better if data from the baseline were known. However, it is encouraging that the performance shown here is comparable with many offline supervised methods, particularly for the three-storey building structure [274, 275, 277].

The introduction of this DP methodology to the SHM community opens up the opportunity for further work on the use of this algorithm. For instance, strategies for determining optimal meta-features (triggers) which can guide inspection schedules or interventions. There is also room for investigation into how prior knowledge can be best exploited in the model, particularly if partially labelled datasets can be used to improve hyperparameter selection — specifically selection of the α hyperparameter.

Finally, the model also provides flexibility to remove information from clusters, this may not seem helpful but could be the groundwork for building a system that is robust to slowly varying trends which can be found in environmental confounding influences. In summary, this work has proposed a novel approach to online detection and classification which lays the foundation for further exploitation of this flexible Bayesian model in the future.

Chapter 8

Conclusions and Future Work

The work contained in this thesis has explored the use of different Bayesian approaches to tackle a variety of problems in SHM. The offshore environment presents a challenging and fruitful area in which to apply SHM techniques due to the difficulty and cost in traditional inspection and maintenance procedures. The use of probabilistic methods more readily lends itself to risk based inspection planning and the process of making decisions under uncertainty. For this the Bayesian paradigm is presented as a natural and powerful framework in which complex models can be created, prior beliefs — or lack thereof — can be (and must be) formally incorporated, and uncertainty is not considered something to be ignored or eliminated but an integral part of understanding the world.

The problem of how to understand the state of a structure has been approached from a number of different angles. This has included the use of machine learning techniques for better understanding unknown inputs to a dynamical system, as well as online detection and classification of structural condition. Each of the techniques used have been shown to exhibit both desirable and undesirable properties and it is worth highlighting these again here.

It would be sensible to propose a *no free lunch* theorem for SHM, it is safe to conclude that there is no single algorithm that will solve all of the challenges faced. It should be noted that it has not been possible to prove this rigorously here! Hopefully, however, through continued research it will be possible to understand better the most appropriate methods for given situations. The results seen in this thesis give some

indication of situations in which certain methodologies may (or may not) perform well.

8.1 Gaussian Processes for SHM

It has been shown in the literature and the examples throughout this thesis that the application of a Gaussian Process model for SHM regression tasks can be a powerful tool — for example in virtual sensing applications. The ability to handle nonlinear regression tasks, by construction, has allowed it to be used when physical models may be difficult or even impossible to define. The lack of need for a basis function reduces the difficulty in model selection and the optimisation of the hyperparameters against the negative log marginal likelihood allows exploitation of the Bayesian Occam's Razor [100, 101] to find minimally complex models.

However, a number of steps in the formulation of the GP model — which can be easily overlooked when reading the machine learning literature — are shown in this thesis to have serious implications in an SHM application. Firstly, the choice of kernel in the Gaussian Process takes on a more tangible meaning in an SHM context where prior (and sometimes approximate) knowledge about the system can be encoded in an intuitive way; e.g. "I plotted it and it was roughly linear". The elicitation of this information and encoding it in a sensible kernel form still requires expert intervention. It is also difficult to make these leaps of intuition as the dimensionality of the input data increases due to difficulties in visualisation.

For the first time (to the author's knowledge), a population based optimisation approach to learning the GP hyperparameters has been implemented and compared to the popular gradient descent methodology. It has been shown that, not only is the population based approach a viable alternative to gradient descent, but it often outperforms a conjugate gradient approach in terms of consistency and optimal value. The drawback of this is the requirement for an increased number of function evaluations, however, when considering the need to perform multiple random restarts of the conjugate gradient method the gap is closed significantly. This was demonstrated on two synthetic datasets and also in a virtual sensing application from SHM. The prediction of port inner wing bending strain on a Tucano aircraft was shown to be sensitive to both the quality of the optimisation and the choice of kernel — which could only be made given physical insight. Here, the negative log marginal

likelihood was also seen to be a poor cost function for kernel selection as it penalised heavily the added model complexity of the added linear kernel which was essential to allowing generalisation of the model from a single flight to another randomly chosen flight. The alternative solution to this would be to increase the training set size sufficiently that the process could be fully represented by the nonlinear kernel, however, this would introduce further difficulties in implementing a Gaussian Process model with a large number of data points.

The applicability of the GP was extended by considering its ability to predict in a situation with dynamic relationships in the data by utilising it with a NARX framework. Some of the additional challenges that this represents (and which are often missing from literature using these models) were discussed. This included the difficulty in propagating uncertainty when making model predicted outputs — i.e. when there is feedback of model predictions to the input. The approach of Monte-Carlo sampling is compared to that of the exact propagation, when making a Gaussian approximation via moment matching. The problem of lag selection was also discussed from the perspective of a GP-NARX model and this was contrasted with the difficulty in polynomial NARX models where basis function terms must also be determined alongside the lag selection.

The load estimation problem for offshore, to predict the unknown forces exerted by the wave was first tackled using these Gaussian Process tools. Some of the potential benefits and pitfalls of applying the model were highlighted — especially GP-NARX. Lag selection was shown to be a difficult challenge that can greatly affect the quality of a GP-NARX model. It was also demonstrated that the use of a NMSE cost function for deciding on the most appropriate lags leads to models that fail to generalise. Instead the use of the posterior probability was shown to better highlight model quality when looking for models that would generalise for multiple step ahead predictions. It was shown for the wave loading that a static model outperformed a dynamic one if the object of interest was forecasting multiple points into the future and training was conducted using the negative log marginal likelihood. However, for the one step ahead prediction the GP-NARX model was able to far better capture the behaviour of the wave loading signal than the white-box Morison equation. In fact, even for the full model predicted output the static GP outperforms the Morison equation. The switch to a cross-validation learning approach improved the consistency of learning in the model and further improved the best performance seen from the model on unseen test data.

It may be possible, based upon the results seen in this work, to make a number of recommendations regarding the use of GPs. Firstly, it is important to consider the physical implications of the kernel choice before attempting to model the data. It is also important to establish that the training data is drawn from the same distribution as the test data, that this distribution is stationary, and that enough points have been observed to cover the possible test input space. This can often be challenging due to the high cost associated with obtaining training data for SHM. It is also seen that the most useful application of the GP-NARX may be in a situation such as model predictive control where long term forecasting is not required and the model can receive measured outputs to counteract the increase in uncertainty as time progresses. For a process like wave loading, the approach of using a GP-NARX may be difficult to implement as the uncertainty will continue to increase until the returning to the prior distribution of the GP. This imposes a horizon on how long the model can be effective for (in actuality the model will not contain much useful information long before it returns to the prior) thus limiting the life of the model.

8.2 State-Space Modelling for SHM

The state-space model was introduced as an alternative framework in which to perform system identification for parametric models such as the Duffing oscillator as well as semi-parametric models in the Latent Force Model. This could be extended to nonparametric models if, for example, a Gaussian Process State-Space Model was adopted although this isn't shown in this work.

The first task attempted was to make use of this state-space framework to perform identification of a Duffing oscillator. Here the displacement of the oscillator had been measured with a very high measurement noise (50% RMS) which would make usual methods of parameter identification difficult and which obscured the other states of the model — the velocity and acceleration. A particle Gibbs framework was used to allow the smoothing distributions over the hidden states of the model to be recovered whilst estimating the parameter distributions of the model in a fully Bayesian manner. A number of challenges in implementing this kind of model became apparent.

It was necessary to employ ancestor sampling and particle rejuvenation to help mixing in the Markov Chain. This allowed the use of far fewer particles in the

method which in turn led to increased mixing for lower computation time. It is usual that structural dynamics systems will have a low amount of process noise (in some exceptional cases zero) so it was necessary to make use of particle rejuvenation. The author believes this is the first use of a particle Gibbs scheme in structural dynamic system identification, not withstanding the use of more modern techniques such as ancestor sampling and particle rejuvenation which were also introduced in this work. Problems still exist in this methodology, however, it is noted that the sampling frequency will affect the quality of the results due to the first order discretisation used in the Gibbs sampler and that the overestimation of process noise can lead to misidentification of the damping parameter. This, along with other extensions, will be discussed when considering future work to be completed.

Returning to linear systems and combining the state-space modelling framework with the flexibility of Gaussian process regression, this work introduces the use of the Gaussian Process Latent Force Model for operational modal analysis. It is shown that work on representation of the Gaussian Process as a linear state-space model can be combined with the Latent Force idea to produce a model ideally suited to OMA. Here the unknown inputs can be converted into latent states in the model whose form can have a prior distribution placed over it. Rather than using a random walk which may give rise to poor estimations of the loading time history, a Gaussian Process prior can be used to encode beliefs about smoothness or periodicity. As an aside, it can be shown that the random walk assumption is in fact a special case of a Gaussian Process with an exponential kernel described by a Matérn kernel whose roughness parameter $\nu = 1/2$.

By employing this model, it was shown that an output only identification is possible even if the forcing is non-Gaussian and close to the resonance of the structure. This is a very powerful tool for the offshore industry where it can be difficult to ensure that resonant frequencies of the structure are not close to the wave loading frequency. The identification is also conducted within a Markov Chain Monte Carlo approach such that the parameter distributions are learnt in a Bayesian manner. The state-space approach has been shown to be a more flexible extension to the NARX model in which inference can be made about both parameters in a model and about latent states. This is a natural framework in which to consider dynamic problems that inherently contain these hidden states.

8.3 Dirichlet Processes for SHM

Finally, an alternative look at the SHM challenge was presented in Chapter 7 where a Dirichlet Process model was used in combination with an online Gibbs sampler to allow damage detection and classification without the need to pre-collect a large training dataset or to specify the expected damage mechanisms. This approach of non-parametric Bayesian clustering allowed information to be incorporated online and for prior information (e.g. known labels) to be handled effectively. It was shown to be a suitable methodology for output only damage identification and classification on both a laboratory structure and a full scale structure.

The investigation of methods where there is either incomplete or no training data is a key future direction for SHM. The large costs associated with testing and collecting labelled datasets from structures is too high for fully supervised methods to be the only option if an end user wishes to perform classification. The key implication of being able to operate online from day zero is increased benefit and operation time for the same cost of installation of a system on a structure. It is usually difficult to anticipate all of the possible conditions a structure might experience, the benefit of a method which can generate new classes/clusters as needed is the removal of this need to learn how many states there are from a large training set.

8.4 Future Work

The research carried out for this project has raised a number of interesting avenues for further investigation which will motivate future work in this area. The Gaussian Process model is a promising technology for SHM and its application not only in this work but in literature would encourage continued research into its application to SHM. Although the work shown here has demonstrated that the training of this kind of model can be improved with the use of a population based optimisation scheme; it would be valuable to compare this to Bayesian approaches to estimating the hyperparameters (e.g. [88]) or to alternative optimisation methods such as a Bayesian optimisation [287]. It is worth bearing in mind the *no free lunch* theorem of Wolpert and Macready [121], however, it does appear that there is sustained advantage in ensuring the choice of an appropriate optimisation scheme for this particular problem.

A far harder area of research that remains open is that of automating the task of model selection inside for GP models, especially when the input data exist in high dimensional spaces. It may be found that it is helpful to reduce the dimensionality of this data in order to aid this, the use of dimensionality reduction for sparsity has already been explored [288] and leaves it as a promising area of investigation. An alternative approach would be to consider a methodology like Bayesian Model Averaging [289] where multiple model forms could be combined to avoid the selection problem. It is worth considering as well if the Gaussian assumption in the Gaussian Process is valid or if it is more valuable employing a robust regression method, e.g. [290].

As the results in Chapter 4 show, the modelling of dynamic data with a GP remains a significant challenge. It is worth considering how improvements can be made in this area. A promising avenue of investigation is the incorporation of physical knowledge into the system by means of a *grey-box* philosophy, in fact it is readily argued that the approach of the GP LFM does this! If pursuing the use of a GP-NARX model then the same kernel selection questions from the static case must be addressed. In addition to this it will be necessary to develop a more rigorous methodology for lag selection, possibly borrowing techniques from older polynomial models. Alternatively this could be another application of Bayesian Model Averaging if several models with randomly selected lags are considered. It can also be possible to train the model for better MPO behaviour by optimising against the posterior probability of a cross validation dataset for learning both the hyperparameters and the lags — the affect of this choice should be investigated in more detail in the future.

As the size of available datasets continues to increase it will also become necessary to benchmark the behaviour of different approximations of the GP designed to reduce the computational load. This may include the use of sparse Gaussian Processes [125–127, 291] or reduced rank approximations [128]. The work referenced in application of the GP LFM where the GP can be represented as a linear SSM can also lead to computational advantage [228]. It is also worthwhile considering how this type of model could be implemented in an online setting, for example as in [292].

As discussed, the framework of state-space models is a more general model form in which the NARX model family sits. Continued investigation of this type of model is also a area of research that deserves future attention. In terms of nonlinear system identification, the next short-term step is to prove the effectiveness of the methodology shown in Chapter 5 on measured data from a laboratory test and also to

prove the usefulness of the model when there is an additional hidden state, as in the Bouc-Wen hysteresis model [293]. The model also currently requires the dynamics of the system to be known *a priori*, this limitation would motivate investigating either model selection within this framework or a move to a semi/non-parametric model for the nonlinearity. In many engineering systems, this is a realistic goal since the nonlinearities in a system can remain static, e.g. the cubic term in a Duffing oscillator is not time dependent. There is also further work to be done into how to handle the lack of process noise in many engineering systems and the effect of the sample rate on biasing the parameter estimates. It should additionally be investigated whether the linear discretisation in the Particle Gibbs sampler can be handled in a different manner to reduce the effect of the sampling frequency.

The LFM requires proving on full scale structures, it is likely in this scenario that the system will need to be represented via a reduced order model. The effect of this on the efficacy of the method remains to be seen. The possibility is also open to combine the Particle Gibbs methodology used on the Duffing oscillator with the LFM model to perform output only identification under a non-Gaussian excitation. The LFM may also see improved behaviour from better kernel selection and faces many of the same challenges as are seen in kernel selection for the standard GP.

The effectiveness of the Dirichlet Process for online identification and classification of damage has been shown here. However, continued investigation into building a decision system around the process is a key area of research. Incorporating this type of model into a method which can optimise inspection intervals is a key objective. It may also be possible to consider modifications to the algorithm where long term environmental trends can be eliminated increasing the robustness of the method. It is also worth considering if the method can be extended to the case where the base distribution is non-Gaussian, as many datasets in SHM are in fact mixtures of non-Gaussian distributions.

The final area of future work is around how to interpret the outputs of these models and how the outputs can be incorporated into robust automated decision frameworks. This may be the key driver in adoption of SHM as a whole. This is motivated by the desire to eliminate expert intervention and allow complete SHM systems (sensor networks, signal processing methods, learning algorithms, and decision engines) to make cost-optimal decisions in an automated manner. This is, however, maybe best discussed in the next and final section.

8.5 The Outlook for Offshore

I would like to round off this thesis by offering up a few opinions and observations on the possible outlook for the offshore sector with relation to SHM. Here it might be useful to discuss some of my perceived obstacles to widespread adoption of an automated SHM driven inspection and maintenance strategy in the offshore industry — although many of these comments could be equally applied to other industries. For this adoption happen across offshore, I see three key barriers:

1. *It needs to be proven that SHM does not compromise safety compared to conservative regular inspections*
2. *The cost benefit to the end user of SHM must be shown*
3. *SHM solutions need to be presented as complete systems for incorporation in business operations*

In the community, we are very good at coming up with more impressive components of SHM systems. I believe that it will be necessary to begin to present the overall system which I see as a combination of four sub-components[a]:

1. *The sensor network*
2. *The data management and signal processing*
3. *The learning algorithms*
4. *The decision engine*

Compare this with the outlines set out by Farrar and Worden [98] where it is stated that "The process of implementing a damage identification strategy for aerospace, civil and mechanical engineering infrastructure is referred to as structural health monitoring (SHM)". It is my feeling that the challenge of SHM goes beyond simply moving up Rytter's hierarchy [7, 8] but in the coming years major developments must be made in how a system can be specified and proven

to have tangible business benefit (an issue that is raised in [98]) and how systems can be designed with autonomous or semi-autonomous decision making in mind.

The use of machine learning has revolutionised the way in which damage detection is done (for data-driven SHM) and ever more sophisticated statistical methods are being employed. However, maybe the weakest area of development within SHM is that of automated decision engines.

One perceived issue may be that each of these steps are normally discussed in isolation. If attending an SHM conference this problem is most stark with each of these different disciplines separating themselves into streams. This is of course a generalisation and in fact more general streams do exist. It will be noted though that it is rare to see a presentation that covers the SHM process from deployment to decision — or outlining how this type of process may take place. I think the reason for this is twofold; first this is a massive undertaking to consider this type of end-to-end solution (certainly it is nearly impossible to cover in a twenty minute presentation). Secondly though, I feel that there is a general lack of consideration of how, for instance, a new damage localisation algorithm, may fit into the overall operation of a business concerned with ensuring integrity of its assets.

I will attempt to, for offshore, begin this conversation now of one potential roadmap to implementing a full SHM solution. In doing I will highlight at each step some of the challenges. The solutions to some of these are currently available in the literature but for some they are not. The purpose here is not to say this is the "correct way to do SHM" but instead to help formulate some of the questions an end user should be asking.

The starting point should be an analysis of the business case for SHM, this will be guided by a number of things. The expected cost of a conventional inspection and maintenance strategy, the asset value, the availability of capital for installing a system, the expected cost of the system, whether there are case studies on similar structures to prove effectiveness, to name a few. It is my understanding that this discussion remains open in both the literature and in industry and quantification of real cost-benefit is very challenging for SHM. However, assuming for now that a business considers it beneficial to install as system or if some form of monitoring is mandated by the regulator, the next stage should be specification

of a sensor system.

In offshore, one of the key challenges is in specifying and installing effective sensor systems. First of all it is worth looking ahead at how the sensor system will be used. For example, if a user is only interested in the behaviour of a few key components it may be beneficial to concentrate the sensors around these components. In the future it is also hopeful that there will be continued development into optimal sensor placement, e.g. Flynn and Todd [294]. Developments that will aid the installation of systems on offshore structure will include the cost of installation in different areas of the structure, and mixed sensor types. In order to guide this optimal sensor placement it is also necessary for there to be understanding of the expected failure in the structures, consideration of population level data may help in this regard [295]. This is expected to become especially true in wind farms where there are a number of nominally identical structures.

In the most part, the appropriate signal processing steps will be guided by the sensor types and expected damage. However, the removal of confounding influences will continue to be a challenge in offshore due to the highly changeable environment and difficulty in measuring all the variables of interest. Into this, questions should be asked about data management and integrity. It is important to ensure that data are stored efficiently and in a manner which is easily accessed by other components of the SHM system, e.g. the damage detection algorithm. As the systems are expected to operate for extended periods of time, users should be especially careful regarding the installation of multiple disconnected sensor networks and ensuring their synchronisation. Long term monitoring of large scale structures should also motivate the development of algorithms for SHM that can effectively process large and growing databases efficiently — this includes the development of methods that can operate online and in parallel. The development of asynchronous online learning algorithms would be a valuable pursuit for SHM.

In terms of learning algorithms for offshore, I believe the development of algorithms which can incorporate data online will significantly improve SHM system performance over its lifetime. This will reduce cost and increase confidence in deploying monitoring systems. However, the choice of learning algorithm whether it is a novelty detector, virtual sensing method, classifier, or any other

methodology should be probabilistic in an environment with as much uncertainty as offshore. Methods used should be able to self-quantify the confidence that they make in their outputs. The reason for this is that this is what allows integration into risk based decision frameworks. Of course in the future we will see methods that have lower false alarms, higher detection rates and lower errors but philosophically I do not believe that the perfect method will (or indeed could) ever be found. Any assumption that an assessment method is infallible is a dangerous one.

The final building block of an effective SHM strategy for offshore will be the introduction of decision engines. These decision engines will need to be able to aggregate all the accumulated uncertainty in a process and present a series of decisions ranked in order of risk to the end user. My impression is that this is one of the great open questions in SHM. Can we develop this decision engine that can handle multiple uncertain models, uncertain data sources, and limited data; then present in a simple manner a series of possible actions with associated risk? Consider a simple example where already this problem clearly requires a probabilistic approach.

A structure has a sensor system installed, but it is known to have varying sensitivity across the structure leading to uncertain areas and in some places due to limitations in sensor placement unknowns, e.g. in the foundations. A virtual sensing approach is taken to reduce this uncertainty but this method has its own associated uncertainty. There are then three different systems employed to understand the structural condition: the first a Bayesian method akin to the one presented in Chapter 7 for anomaly detection and classification, the other a data-driven fatigue model, the last a finite element updating based approach. The decision engine must be able to aggregate the information from the uncertain sensing system, the virtual sensor information and the damage detection systems. The task of objectively combining this information is not possible for human operators which motivates the need for further investigation of decision engines.

In conclusion, to see widespread adoption of an SHM driven approach to inspection and maintenance in offshore, there is not just a requirement to continue developments of components but instead the presentation of full systems should become more prevalent. In addition to this there is currently a lack of automated

risk based decision engines which are easily interpretable. Finally, the understanding and handling of uncertainty will only become more important as more data are collected — it appears to me that the only framework in which this can be done with the complexity and rigour required is the Bayesian one. Therefore, I hope that the research conducted for this thesis can contribute to the ongoing development of Bayesian approaches to the SHM problem as a whole, but more importantly it would be satisfying to see these methods or derivatives thereof implemented on live structures.

[a] These components may seem more readily applicable to a data-driven framework, however, it can be argued that the learning algorithm stage would also cover methods such as model updating and sensitivity analysis.

Linear Gaussian State-Space Models

The LG-SSM is the simplest useful form of Bayesian SSM that is analysed and has closed forms for both the filtering densities [192] and the smoothing densities [229]. The LG-SSM is a special case of the general Bayesian state space model shown in Equation (5.3) where both the transition and observation models are linear functions and there is Gaussian noise on both the transition and the observation. The discrete form of the LG-SSM is shown here[1],

$$\mathbf{x}_t = A\,\mathbf{x}_{t-1} + B\,\mathbf{u}_{t-1} + w_{t-1} \qquad w_{t-1} \sim \mathcal{N}(0\,,\ Q) \tag{A.1a}$$

$$\mathbf{y}_t = C\,\mathbf{x}_t + D\,\mathbf{u}_t + v_t \qquad v_t \sim \mathcal{N}(0\,,\ R) \tag{A.1b}$$

Which when written in its probabilistic representation appears as,

$$p_\theta(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}, \mathbf{u}_{t-1}) = \mathcal{N}(A\,\mathbf{x}_{t-1} + B\,\mathbf{u}_{t-1}\,,\ Q) \tag{A.2a}$$

$$p_\theta(\mathbf{y}_t \,|\, \mathbf{x}_t, \mathbf{u}_t) = \mathcal{N}(C\,\mathbf{x}_t + D\,\mathbf{u}_t\,,\ R) \tag{A.2b}$$

[1]It is possible to extend this model such that the matrices A, B, C, D, Q, R are also dependent on time. This is not shown here to avoid notational clutter.

The first distribution to be calculated to solve the filtering and smoothing problems is the predictive distribution $p_\theta\left(\mathbf{x}_t \,|\, \mathbf{y}_{1:t-1}, \mathbf{u}_{t-1}\right)$. The convention is adopted that at any given time t the states are distributed $\mathbf{x}_t \sim \mathcal{N}\left(\mathbf{m}_t\,,\ P_t\right)$. A key part of the filter is the recursive nature of the calculations, it is assumed, therefore, that some initial mean $\mathbf{m}_0$ and covariance P_0 is known or can be estimated.

$$
\begin{aligned}
p_\theta\left(\mathbf{x}_t \,|\, \mathbf{y}_{1:t-1}, \mathbf{u}_{1:t-1}\right) &= \int p_\theta\left(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}, \mathbf{u}_{t-1}\right) p_\theta\left(\mathbf{x}_{t-1} \,|\, \mathbf{y}_{1:t-1}, \mathbf{u}_{1:t-1}\right) \mathrm{d}\mathbf{x}_{t-1} \\
&= \int p_\theta\left(\mathbf{x}_{t-1}, \mathbf{x}_t \,|\, \mathbf{y}_{1:t-1}, \mathbf{u}_{1:t-1}\right) \mathrm{d}\mathbf{x}_{t-1} \\
&= \int \mathcal{N}\left(\hat{\mathbf{m}}\,,\ \hat{P}\right) \mathrm{d}\mathbf{x}_{t-1} \\
\hat{\mathbf{m}} &= \begin{bmatrix} \mathbf{m}_{t-1} \\ A\,\mathbf{m}_{t-1} + B\,\mathbf{u}_{t-1} \end{bmatrix} \\
\hat{P} &= \begin{bmatrix} P_{t-1} & P_{t-1}A^{\mathsf{T}} \\ AP_{t-1} & AP_{t-1}A^{\mathsf{T}} + Q \end{bmatrix} \\
&= \mathcal{N}\left(\underbrace{A\,\mathbf{m}_{t-1} + B\,\mathbf{m}_{t-1}}_{\bar{m}_t}\,,\ \underbrace{AP_{t-1}A^{\mathsf{T}} + Q}_{\bar{P}_t}\right)
\end{aligned}
\tag{A.3}
$$

It is useful to define intermediate variables relating to the prediction step of the filter which is the distribution $p_\theta\left(\mathbf{x}_t \,|\, \mathbf{y}_{1:t-1}, \mathbf{u}_{1:t-1}\right)$, such that the prediction is written $p_\theta\left(\mathbf{x}_t \,|\, \mathbf{y}_{1:t-1}, \mathbf{u}_{1:t-1}\right) = \mathcal{N}\left(\bar{\mathbf{m}}_t\,,\ \bar{P}_t\right)$. To arrive at the likelihood $p_\theta\left(\mathbf{y}_t \,|\, \mathbf{y}_{1:t-1}, \mathbf{u}_{1:t-1}\right)$ and the filtering distribution $p_\theta\left(\mathbf{x}_t \,|\, \mathbf{y}_{1:t}, \mathbf{u}_{1:t}\right)$ it is first necessary to write down the joint distribution over $\mathbf{x}_t$ and $\mathbf{y}_t$ — $p_\theta\left(\mathbf{x}_t, \mathbf{y}_t \,|\, \mathbf{y}_{1:t-1}, \mathbf{u}_{1:t}\right)$,

$$
p_\theta\left(\mathbf{x}_t, \mathbf{y}_t \,|\, \mathbf{y}_{1:t-1}, \mathbf{u}_{1:t}\right) = \mathcal{N}\left(\begin{bmatrix} \bar{\mathbf{m}}_t \\ C\bar{\mathbf{m}}_t + D\mathbf{u}_t \end{bmatrix}\,,\ \begin{bmatrix} \bar{P}_t & \bar{P}_t C^{\mathsf{T}} \\ C\bar{P}_t & C\bar{P}_t C^{\mathsf{T}} + R \end{bmatrix}\right) \tag{A.4}
$$

The observation likelihood is simply the marginalisation with respect to $\mathbf{x}_t$ over this conditional such that

$$
p_\theta\left(\mathbf{y}_t \,|\, \mathbf{y}_{1:t-1}, \mathbf{u_t}\right) = \mathcal{N}\left(C\bar{\mathbf{m}}_t + D\mathbf{u}_t\,,\ C\bar{P}_t C^{\mathsf{T}} + R\right) \tag{A.5}
$$

Again for clarity it is useful to define two intermediary variables, the sensitivity

matrix $S = C\bar{P}_t C + R$ and the Kalman gain $K = \bar{P}_t C^\mathsf{T} S^{-1}$. Applying the identity for conditional Gaussian distributions the filtering distribution can be found.

$$\begin{aligned} p_\theta\left(\mathbf{x}_t \,|\, \mathbf{y}_{1:t}, \mathbf{u}_{1:t}\right) &= p_\theta\left(\mathbf{x}_t \,|\, \mathbf{y}_t, \mathbf{y}_{1:t-1}, \mathbf{u}_{1:t}\right) \\ &= \mathcal{N}\left(\mathbf{m}_t\,,\ P_t\right) \\ \mathbf{m}_t &= \bar{\mathbf{m}}_t + K\left[\mathbf{y}_t - C\bar{\mathbf{m}}_t - D\mathbf{u}_t\right] \\ P_t &= \left(\mathbb{I} - KC\right)\bar{P}_t \end{aligned} \tag{A.6}$$

In this way the filtering equations can be written down in closed for for an LG-SSM. Following a similar line of reasoning it is also possible to establish the Rauch-Tung-Striebel smoothing equations [229] in the LG-SSM although these are merely stated here since their derivation is very similar to that of the filtering equations.

$$\begin{aligned} p_\theta\left(\mathbf{x}_t \,|\, \mathbf{y}_{1:T}, , \mathbf{u}_{1:T}\right) &= \mathcal{N}\left(\mathbf{m}_t^s\,,\ P_t^s\right) \\ G_t &= P_t A^\mathsf{T} \bar{P}_{t+1}^{-1} \\ \mathbf{m}_t^s &= \mathbf{m}_t + G_t\left[\mathbf{m}_{t+1}^s - \bar{\mathbf{m}}_{t+1}\right] \\ P_t^s &= P_t + G_t\left[P_{t+1}^s - \bar{P}_{t+1}\right] G_t \end{aligned} \tag{A.7}$$

Details of 5TH Order Runge-Kutta Scheme

The numerical integration scheme is a 5^{th} order Runge-Kutta method [213], whose details are shown here for completeness.

$$x_{t+1} = x_t + \frac{\Delta}{90}\left(7a_1 + 32a_3 + 12a_4 + 32a_5 + 7a_6\right) \tag{B.1a}$$

$$a_1 = \mathbf{f}\left(t, x_t\right) \tag{B.1b}$$

$$a_2 = \mathbf{f}\left(t + 0.25\Delta, x_t + 0.25\Delta a_1\right) \tag{B.1c}$$

$$a_3 = \mathbf{f}\left(t + 0.125\Delta, x_t + 0.125\Delta a_1 + 0.125\Delta a_2\right) \tag{B.1d}$$

$$a_4 = \mathbf{f}\left(t + 0.5\Delta, x_t - 0.5\Delta a_2 + \Delta a_3\right) \tag{B.1e}$$

$$a_5 = \mathbf{f}\left(t + 0.75\Delta, x_t + 0.1875\Delta a_1 + 0.5625\Delta a_4\right) \tag{B.1f}$$

$$a_6 = \mathbf{f}\left(t + \frac{6}{7}\Delta, x_t - \frac{3}{7}\Delta a_1 + \frac{2}{7}\Delta a_2 + \frac{12}{7}\Delta a_3 - \frac{12}{7}\Delta a_4 + \frac{8}{7}\Delta a_5\right) \tag{B.1g}$$

Where Δ is the time step and $\mathbf{f}\left(\cdot\right)$ is the vector Markov form of the ODE which for the Duffing oscillator Equation (5.14) is given by,

$$\mathbf{f}\left(t, x_t\right) = \begin{bmatrix} x_{t,2} \\ \frac{F(t)}{m} - \frac{c}{m} x_{t,1} - \frac{k}{m} x_{t,1} - \frac{k_3}{m} x_{t,1}^3 \end{bmatrix} \tag{B.2}$$

The notation $x_{t,1}$ denotes the value of the first state in x_t this corresponds to the displacement of the oscillator. Equivalently, $x_{t,2}$ corresponds to the velocity value.

Bibliography

[1] R. Kitchin. *The data revolution: Big data, open data, data infrastructures and their consequences*. Sage, 2014.

[2] K. Schwab. *The fourth industrial revolution*. Crown Business, 2017.

[3] J. E. Mottershead and M. Friswell. Model updating in structural dynamics: a survey. *Journal of sound and vibration*, 167(2):347–375, 1993.

[4] M. Friswell and J. E. Mottershead. *Finite element model updating in structural dynamics*, volume 38. Springer Science & Business Media, 2013.

[5] K. Worden and G. Manson. The application of machine learning to structural health monitoring. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 365(1851):515–537, 2007.

[6] C. R. Farrar and K. Worden. *Structural Health Monitoring: A Machine Learning Perspective*. John Wiley & Sons, 2012.

[7] A. Rytter. *Vibrational based inspection of civil engineering structures*. PhD thesis, Dept. of Building Technology and Structural Engineering, Aalborg University, 1993.

[8] K. Worden and J. M. Dulieu-Barton. An overview of intelligent fault detection in systems and structures. *Structural Health Monitoring*, 3(1):85–98, 2004.

[9] K. Worden, C. R. Farrar, G. Manson, and G. Park. The fundamental axioms of structural health monitoring. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 463(2082):1639–1664, 2007.

[10] M. M. Moya and D. R. Hush. Network constraints and multi-objective optimization for one-class classification. *Neural Networks*, 9(3):463–474, 1996.

[11] K. Worden. Structural fault detection using a novelty measure. *Journal of Sound and vibration*, 201(1):85–101, 1997.

[12] K. Worden, G. Manson, and N. R. J. Fieller. Damage detection using outlier analysis. *Journal of Sound and Vibration*, (3):647–667, 2000.

[13] K. Worden, G. Manson, and D. Allman. Experimental validation of a structural health monitoring methodology: Part I. novelty detection on a laboratory structure. *Journal of Sound and Vibration*, 259(2):323–343, 2003.

[14] G. Park, A. C. Rutherford, H. Sohn, and C. R. Farrar. An outlier analysis framework for impedance-based structural health monitoring. *Journal of Sound and Vibration*, 286(1-2):229–250, 2005.

[15] J. Toivola, M. A. Prada, and J. Hollmén. Novelty detection in projected spaces for structural health monitoring. In *International Symposium on Intelligent Data Analysis*, pages 208–219. Springer, 2010.

[16] N. Dervilis, E. Cross, R. Barthorpe, and K. Worden. Robust methods of inclusive outlier analysis for structural health monitoring. *Journal of Sound and Vibration*, 333(20):5181–5195, 2014.

[17] N. Dervilis, K. Worden, and E. J. Cross. On robust regression analysis as a means of exploring environmental and operational conditions for SHM data. *Journal of Sound and Vibration*, 347:279–296, 2015.

[18] R. Brincker, P. H. Kirkegaard, P. Andersen, and M. Martinez. Damage detection in an offshore structure. In *Proceedings-Spie The International Society for Optical Engineering*, pages 661–661. SPIE INTERNATIONAL SOCIETY FOR OPTICAL, 1995.

[19] J. Gao and P.-N. Tan. Converting output scores from outlier detection algorithms into probability estimates. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 212–221. IEEE, 2006.

[20] C. M. Bishop. Pattern recognition. *Machine Learning*, 128, 2006.

[21] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, NY, USA:, 2001.

[22] A. Farhidzadeh, S. Salamone, and P. Singla. A probabilistic approach for damage identification and crack mode classification in reinforced concrete structures. *Journal of Intelligent Material Systems and Structures*, 24(14): 1722–1735, 2013.

[23] N. Mechbal, J. S. Uribe, and M. Rébillat. A probabilistic multi-class classifier for structural health monitoring. *Mechanical Systems and Signal Processing*, 60:106–123, 2015.

[24] F. Kopsaftopoulos and S. Fassois. A functional model based statistical time series method for vibration based damage detection, localization, and magnitude estimation. *Mechanical Systems and Signal Processing*, 39(1-2):143–161, 2013.

[25] S. Sankararaman and S. Mahadevan. Bayesian methodology for diagnosis uncertainty quantification and health monitoring. *Structural Control and Health Monitoring*, 20(1):88–106, 2013.

[26] X.-S. Si, W. Wang, C.-H. Hu, and D.-H. Zhou. Remaining useful life estimation–a review on the statistical data driven approaches. *European journal of operational research*, 213(1):1–14, 2011.

[27] J. Sikorska, M. Hodkiewicz, and L. Ma. Prognostic modelling options for remaining useful life estimation by industry. *Mechanical Systems and Signal Processing*, 25(5):1803–1836, 2011.

[28] E. Zio and G. Peloni. Particle filtering prognostic estimation of the remaining useful life of nonlinear components. *Reliability Engineering & System Safety*, 96(3):403–409, 2011.

[29] C. Sbarufatti, M. Corbetta, M. Giglio, and F. Cadini. Adaptive prognosis of lithium-ion batteries based on the combination of particle filters and radial basis function neural networks. *Journal of Power Sources*, 344:128–140, 2017.

[30] A. O'Hagan and J. E. Oakley. Probability is perfect, but we can't elicit it perfectly. *Reliability Engineering & System Safety*, 85(1-3):239–248, 2004.

[31] I. Albert, S. Donnet, C. Guihenneuc-Jouyaux, S. Low-Choy, K. Mengersen, J. Rousseau, *et al.* Combining expert opinions in prior elicitation. *Bayesian Analysis*, 7(3):503–532, 2012.

[32] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng. *Handbook of Markov Chain Monte Carlo*. CRC Press, 2011.

[33] A. Gelman, J. B. Carlin, D. B. Rubin, A. Vehtari, D. B. Dunson, and H. S. Stern. *Bayesian data analysis, third edition*. CRC Press, 2013.

[34] J. O. Berger. *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media, 2013.

[35] A. Gelman and C. R. Shalizi. Philosophy and the practice of Bayesian statistics. *British Journal of Mathematical and Statistical Psychology*, 66(1):8–38, 2013.

[36] R. Coppolino, S. Rubin, *et al.* Detectability of structural failures in offshore platforms by ambient vibration monitoring. In *Offshore Technology Conference*. Offshore Technology Conference, 1980.

[37] R. M. Kenley, C. J. Dodds, *et al.* West sole WE platform: Detection of damage by structural response measurements. In *Offshore technology conference*. Offshore Technology Conference, 1980.

[38] B. Peeters and G. De Roeck. Stochastic system identification for operational modal analysis: a review. *Journal of Dynamic Systems, Measurement, and Control*, 123(4):659–667, 2001.

[39] T. Moan. Reliability-based management of inspection, maintenance and repair of offshore structures. *Structure and Infrastructure Engineering*, 1(1):33–62, 2005.

[40] D. M. Frangopol and M. Liu. Maintenance and management of civil infrastructure based on condition, safety, optimization, and life-cycle cost. *Structure and infrastructure engineering*, 3(1):29–41, 2007.

[41] J. D. Sørensen. Framework for risk-based planning of operation and maintenance for offshore wind turbines. *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, 12(5): 493–506, 2009.

[42] J. J. Nielsen and J. D. Sørensen. On risk-based operation and maintenance of offshore wind turbine components. *Reliability Engineering & System Safety*, 96(1):218–229, 2011.

[43] P. Weber, G. Medina-Oliva, C. Simon, and B. Iung. Overview on Bayesian networks applications for dependability, risk analysis and maintenance areas. *Engineering Applications of Artificial Intelligence*, 25(4):671–682, 2012.

[44] R. Perveen, N. Kishor, and S. R. Mohanty. Off-shore wind farm development: Present status and challenges. *Renewable and Sustainable Energy Reviews*, 29: 780–792, 2014.

[45] D. McMillan and G. W. Ault. Quantification of condition monitoring benefit for offshore wind turbines. *Wind Engineering*, 31(4):267–285, 2007.

[46] Z. Hameed, Y. Hong, Y. Cho, S. Ahn, and C. Song. Condition monitoring and fault detection of wind turbines and related algorithms: A review. *Renewable and Sustainable energy reviews*, 13(1):1–39, 2009.

[47] Y. Feng, Y. Qiu, C. J. Crabtree, H. Long, and P. J. Tavner. Monitoring wind turbine gearboxes. *Wind Energy*, 16(5):728–740, 2013.

[48] I. Antoniadou, N. Dervilis, E. Papatheou, A. Maguire, and K. Worden. Aspects of structural health and condition monitoring of offshore wind turbines. *Phil. Trans. R. Soc. A*, 373(2035):20140075, 2015.

[49] R. Fuentes, T. Howard, M. B. Marshall, E. Cros, R. Dwyer-Joyce, T. Huntley, and R. H. Hestmo. Detecting damage on wind turbine bearings using acoustic emissions and Gaussian process latent variable models. In *Structural Health Monitoring 2015: System Reliability for Verification and Implementation*, volume 2, pages 2302–2309. DEStech Publications, Inc., 2015.

[50] R. Fuentes. *On Bayesian Networks for Structural Health and Condition Monitoring*. PhD thesis, University of Sheffield, 2017.

[51] H. 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.

[52] E. J. Cross, K. Worden, and Q. Chen. Cointegration: a novel approach for the removal of environmental trends in structural health monitoring data. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 467(2133):2712–2732, 2011.

[53] E. Cross. *On Structural Health Monitoring in Changing Environmental and Operational Conditions*. PhD thesis, University of Sheffield, 2012.

[54] J. L. Beck and L. S. Katafygiotis. Updating models and their uncertainties. I: Bayesian statistical framework. *Journal of Engineering Mechanics*, 124(4): 455–461, 1998.

[55] G. Steenackers and P. Guillaume. Finite element model updating taking into account the uncertainty on the modal parameters estimates. *Journal of Sound and Vibration*, 296(4-5):919–934, 2006.

[56] B. Peeters and G. De Roeck. One-year monitoring of the Z24-Bridge: environmental effects versus damage events. *Earthquake Engineering & Structural Dynamics*, 30(2):149–171, 2001.

[57] A. Iliopoulos, W. Weijtjens, D. Van Hemelrijck, and C. Devriendt. Fatigue assessment of offshore wind turbines on monopile foundations using multi-band modal expansion. *Wind Energy*, 20(8):1463–1479, 2017.

[58] A. Skafte, U. T. Tygesen, and R. Brincker. Expansion of mode shapes and responses on the offshore platform valdemar. In *Dynamics of Civil Structures, Volume 4*, pages 35–41. Springer, 2014.

[59] N. Perišić and U. T. Tygesen. Cost-effective load monitoring methods for fatigue life estimation of offshore platform. In *ASME 2014 33rd International Conference on Ocean, Offshore and Arctic Engineering*. American Society of Mechanical Engineers, 2014.

[60] N. Perišić, P. H. Kirkegaard, and U. T. Tygesen. Load identification of offshore platform for fatigue life estimation. In *Structural Health Monitoring, Volume 5*, pages 99–109. Springer, 2014.

[61] N. Noppe, A. Iliopoulos, W. Weijtjens, and C. Devriendt. Full load estimation of an offshore wind turbine based on scada and accelerometer data. In *Journal of Physics: Conference Series*, volume 753, page 072025. IOP Publishing, 2016.

[62] L. Ziegler, S. Voormeeren, S. Schafhirt, and M. Muskulus. Design clustering of offshore wind turbines using probabilistic fatigue load estimation. *Renewable Energy*, 91:425–433, 2016.

[63] K. Maes, A. Iliopoulos, W. Weijtjens, C. Devriendt, and G. Lombaert. Dynamic strain estimation for fatigue assessment of an offshore monopile wind turbine using filtering and modal expansion algorithms. *Mechanical Systems and Signal Processing*, 76:592–611, 2016.

[64] J. F. Wilson. *Dynamics of Offshore Structures*. John Wiley & Sons, 2003.

[65] L. M. Bryant, H. M. Matlock, *et al.* Three-dimensional analysis of framed structures with nonlinear pile foundations. In *Offshore Technology Conference*. Offshore Technology Conference, 1977.

[66] T. Nogami, J. Otani, K. Konagai, and H.-L. Chen. Nonlinear soil-pile interaction model for dynamic lateral motion. *Journal of Geotechnical Engineering*, 118(1):89–106, 1992.

[67] L. V. Andersen, M. Vahdatirad, M. T. Sichani, and J. D. Sørensen. Natural frequencies of wind turbines on monopile foundations in clayey soilsa probabilistic approach. *Computers and geotechnics*, 43:1–11, 2012.

[68] S. Bisoi and S. Haldar. Dynamic analysis of offshore wind turbine in clay considering soil–monopile–tower interaction. *Soil Dynamics and Earthquake Engineering*, 63:19–35, 2014.

[69] L. Liu, S. M. Kuo, and M. Zhou. Virtual sensing techniques and their applications. In *Networking, Sensing and Control, 2009. ICNSC'09. International Conference on*, pages 31–36. IEEE, 2009.

[70] G. Holmes, P. Sartor, S. Reed, P. Southern, K. Worden, and E. Cross. Prediction of landing gear loads using machine learning techniques. *Structural Health Monitoring*, 15(5):568–582, 2016.

[71] R. Temam. *Navier-Stokes equations: theory and numerical analysis*, volume 343. American Mathematical Soc., 2001.

[72] V. Girault and P.-A. Raviart. *Finite Element Methods for Navier-Stokes Equations: Theory and Algorithms*, volume 5. Springer Science & Business Media, 2012.

[73] A. O'Hagan and J. Kingman. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–42, 1978.

[74] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

[75] D. Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[76] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Dunson. *Bayesian data analysis*, volume 3. CRC press, 2014.

[77] R. Fuentes, E. Cross, A. Halfpenny, K. Worden, and R. J. Barthorpe. Aircraft parametric structural load monitoring using gaussian process regression. In *EWSHM-7th European workshop on structural health monitoring*, 2014.

[78] J. Hensman, R. Mills, S. Pierce, K. Worden, and M. Eaton. Locating acoustic emission sources in complex structures using Gaussian processes. *Mechanical Systems and Signal Processing*, 24(1):211–223, 2010.

[79] L. D. Avendaño-Valencia, E. N. Chatzi, K. Y. Koo, and J. M. Brownjohn. Gaussian process time-series models for structures under operational variability. *Frontiers in Built Environment*, 3:69, 2017.

[80] T. E. Fricker, J. E. Oakley, N. D. Sims, and K. Worden. Probabilistic uncertainty analysis of an frf of a structure using a Gaussian process emulator. *Mechanical Systems and Signal Processing*, 25(8):2962–2975, 2011.

[81] F. DiazDelaO and S. Adhikari. Structural dynamic analysis using Gaussian process emulators. *Engineering Computations*, 27(5):580–605, 2010.

[82] C. A. Micchelli, Y. Xu, and H. Zhang. Universal kernels. *Journal of Machine Learning Research*, 7(Dec):2651–2667, 2006.

[83] P. C. Mahalanobis. On the generalized distance in statistics. In *National Institute of Science, India*. National Institute of Science of India, 1936.

[84] A. Abdessalem, N. Dervilis, D. Wagg, and K. Worden. ABC-NS: a new computational inference method applied to parameter estimation and model selection in structural dynamics. In *23 Congrès Français de Mécanique*, 2017.

[85] A. B. Abdessalem, N. Dervilis, D. Wagg, and K. Worden. Model selection and parameter estimation in structural dynamics using approximate Bayesian computation. *Mechanical Systems and Signal Processing*, 99:306–325, 2018.

[86] D. Duvenaud. *Automatic model construction with Gaussian processes*. PhD thesis, University of Cambridge, 2014.

[87] I. Andrianakis and P. G. Challenor. The effect of the nugget on Gaussian process emulators of computer models. *Computational Statistics & Data Analysis*, 56 (12):4215–4228, 2012.

[88] A. Svensson, J. Dahlin, and T. B. Schön. Marginalizing Gaussian process hyperparameters using sequential monte carlo. In *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2015 IEEE 6th International Workshop on*, pages 477–480. IEEE, 2015.

[89] D. Petelin, M. Gasperin, and V. Smídl. Adaptive importance sampling for Bayesian inference in Gaussian process models. *IFAC Proceedings Volumes*, 47 (3):5011–5016, 2014.

[90] R. B. Gramacy and N. G. Polson. Particle learning of Gaussian process models for sequential design and optimization. *Journal of Computational and Graphical Statistics*, 20(1):102–118, 2011.

[91] N. Cressie. The origins of kriging. *Mathematical geology*, 22(3):239–252, 1990.

[92] M. L. Stein. *Interpolation of Spatial Data: Some Theory for Kriging*. Springer, 1999.

[93] A. E. Raftery. Bayesian model selection in social research. *Sociological methodology*, pages 111–163, 1995.

[94] R. E. Kass and A. E. Raftery. Bayes factors. *Journal of the american statistical association*, 90(430):773–795, 1995.

[95] A. Gelman and D. B. Rubin. Avoiding model selection in Bayesian social research. *Sociological methodology*, 25:165–173, 1995.

[96] K. Worden, T. Rogers, and E. Cross. Identification of nonlinear wave forces using Gaussian process narx models. In *Nonlinear Dynamics, Volume 1*, pages 203–221. Springer, 2017.

[97] L. S. Bastos and A. O'Hagan. Diagnostics for Gaussian process emulators. *Technometrics*, 51(4):425–438, 2009.

[98] C. R. Farrar and K. Worden. An introduction to structural health monitoring. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 365(1851):303–315, 2007.

[99] S. Park, D. J. Inman, and C.-B. Yun. An outlier analysis of MFC-based impedance sensing data for wireless structural health monitoring of railroad tracks. *Engineering Structures*, 30(10):2792–2799, 2008.

[100] D. J. MacKay. *Bayesian methods for adaptive models*. PhD thesis, California Institute of Technology, 1992.

[101] C. E. Rasmussen and Z. Ghahramani. Occam's razor. In *Advances in neural information processing systems*, pages 294–300, 2001.

[102] I. Murray and Z. Ghahramani. A note on the evidence and Bayesian Occam's razor. Technical report, Gatsby Compuational Neuroscience Unit, Imperial College London, 2005.

[103] A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3, 2009.

[104] E. Polak and G. Ribiere. Note sur la convergence de méthodes de directions conjuguées. *Revue française d'informatique et de recherche opérationnelle, série rouge*, 3(1):35–43, 1969.

[105] Y.-H. Dai and Y. Yuan. A nonlinear conjugate gradient method with a strong global convergence property. *SIAM Journal on optimization*, 10(1):177–182, 1999.

[106] R. Storn and K. Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

[107] A. K. Qin and P. N. Suganthan. Self-adaptive differential evolution algorithm for numerical optimization. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 2, pages 1785–1791. IEEE, 2005.

[108] J. Zhang and A. C. Sanderson. Jade: adaptive differential evolution with optional external archive. *IEEE Transactions on evolutionary computation*, 13 (5):945–958, 2009.

[109] R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*, pages 39–43. IEEE, 1995.

[110] M. Dorigo and G. Di Caro. Ant colony optimization: a new meta-heuristic. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 2, pages 1470–1477. IEEE, 1999.

[111] A. H. Gandomi and A. H. Alavi. Krill herd: a new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation*, 17(12):4831–4845, 2012.

[112] D. Karaboga and B. Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of global optimization*, 39(3):459–471, 2007.

[113] S. Mirjalili, S. M. Mirjalili, and A. Lewis. Grey wolf optimizer. *Advances in Engineering Software*, 69:46–61, 2014.

[114] R. P. Brent. *Algorithms for Minimization Without Derivatives*. Courier Corporation, 2013.

[115] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou. Multi-verse optimizer: a nature-inspired algorithm for global optimization. *Neural Computing and Applications*, 27(2):495–513, 2016.

[116] J. Sun, B. Feng, and W. Xu. Particle swarm optimization with particles having quantum behavior. In *Evolutionary Computation, 2004. CEC2004. Congress on*, volume 1, pages 325–331. IEEE, 2004.

[117] J. Sun, W. Xu, and B. Feng. A global search strategy of quantum-behaved particle swarm optimization. In *Cybernetics and Intelligent Systems, 2004 IEEE Conference on*, volume 1, pages 111–116. IEEE, 2004.

[118] L. dos Santos Coelho. A quantum particle swarm optimizer with chaotic mutation operator. *Chaos, Solitons & Fractals*, 37(5):1409–1418, 2008.

[119] J. Liu, W. Xu, and J. Sun. Quantum-behaved particle swarm optimization with mutation operator. In *Tools with Artificial Intelligence, 2005. ICTAI 05. 17th IEEE International Conference on*, pages 4–pp. IEEE, 2005.

[120] J. Sun, W. Fang, V. Palade, X. Wu, and W. Xu. Quantum-behaved particle swarm optimization with Gaussian distributed local attractor point. *Applied Mathematics and Computation*, 218(7):3763–3775, 2011.

[121] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82, 1997.

[122] K. Worden, R. Barthorpe, E. Cross, N. Dervilis, G. Holmes, G. Manson, and T. Rogers. On evolutionary system identification with applications to nonlinear benchmarks. *Mechanical Systems and Signal Processing*, 112:194–232, 2018.

[123] S. Reed. Development of a parametric-based indirect aircraft structural usage monitoring system using artificial neural networks. *The Aeronautical Journal*, 111(1118):209–230, 2007.

[124] S. C. Reed. Indirect aircraft structural monitoring using artificial neural networks. *The Aeronautical Journal*, 112(1131):251–265, 2008.

[125] J. Hensman, N. Durrande, and A. Solin. Variational fourier features for Gaussian processes. *arXiv preprint arXiv:1611.06740*, 2016.

[126] T. D. Bui, J. Yan, and R. E. Turner. A unifying framework for sparse Gaussian process approximation using power expectation propagation. *arXiv preprint arXiv:1605.07066*, 2016.

[127] J. Quiñonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6 (Dec):1939–1959, 2005.

[128] A. Solin and S. Särkkä. Hilbert space methods for reduced-rank Gaussian process regression. *arXiv preprint arXiv:1401.5508*, 2014.

[129] R. Murray-Smith and B. A. Pearlmutter. Transformations of Gaussian process priors. In *Deterministic and Statistical Methods in Machine Learning*, pages 110–123. Springer, 2005.

[130] J. Wang, A. Hertzmann, and D. J. Fleet. Gaussian process dynamical models. In *Advances in neural information processing systems*, pages 1441–1448, 2006.

[131] N. D. Lawrence and A. J. Moore. Hierarchical Gaussian process latent variable models. In *Proceedings of the 24th international conference on Machine learning*, pages 481–488. ACM, 2007.

[132] R. Frigola, F. Lindsten, T. B. Schön, and C. E. Rasmussen. Bayesian inference and learning in Gaussian process state-space models with particle mcmc. In *Advances in Neural Information Processing Systems*, pages 3156–3164, 2013.

[133] R. Frigola, Y. Chen, and C. E. Rasmussen. Variational Gaussian process state-space models. In *Advances in Neural Information Processing Systems*, pages 3680–3688, 2014.

[134] A. Svensson, A. Solin, S. Särkkä, and T. Schön. Computationally efficient Bayesian learning of Gaussian process state space models. In *Artificial Intelligence and Statistics*, pages 213–221, 2016.

[135] R. Frigola-Alcade. Bayesian time series learning with Gaussian processes. *University of Cambridge*, 2015.

[136] K. Worden, G. Manson, and E. J. Cross. On Gaussian process narx models and their higher-order frequency response functions. In *Solving Computationally Expensive Engineering Problems*, pages 315–335. Springer, 2014.

[137] K. Worden, W. Becker, T. Rogers, and E. Cross. On the confidence bounds of Gaussian process narx models and their higher-order frequency response functions. *Mechanical Systems and Signal Processing*, 104:188–223, 2018.

[138] J. Kocijan. *Modelling and Control of Dynamic Systems Using Gaussian Process Models*. Springer, 2016.

[139] S. A. Billings. *Nonlinear System Identification: NARMAX Methods In The Time, Frequency, and Spatio-Temporal Domains*. John Wiley & Sons, 2013.

[140] B. Korte and J. Vygen. *Combinatorial Optimization: Theory and Algorithms*. 2008. ISBN 3540718435.

[141] A. Girard, C. E. Rasmussen, J. Q. Candela, and R. Murray-Smith. Gaussian process priors with uncertain inputs application to multiple-step ahead time series forecasting. In *Advances in neural information processing systems*, pages 545–552, 2003.

[142] J. Q. Candela, A. Girard, J. Larsen, and C. E. Rasmussen. Propagation of uncertainty in bayesian kernel models-application to multiple-step ahead forecasting. Technical report, University of Cambridge, 2003.

[143] J. Quiñonero-Candela, A. Girard, and C. E. Rasmussen. Prediction at an Uncertain Input for Gaussian Processes and Relevance Vector Machines Application to Multiple-Step Ahead Time-Series Forecasting. Technical report, University of Cambridge, 2003.

[144] A. Girard, C. E. Rasmussen, J. Q. Candela, and R. Murray-Smith. Gaussian process priors with uncertain inputs-application to multiple-step ahead time series forecasting. *Advances in neural information processing systems*, 2003. ISSN 1049-5258.

[145] A. Girard, C. E. Rasmussen, and R. Murray-Smith. Gaussian process priors with uncertain inputs: Multiple-step ahead prediction. *Univ. Glasgow, Glasgow, Technical Report TR-2002-119*, 2002.

[146] M. KORENBERG, S. Billings, Y. Liu, and P. McIlroy. Orthogonal parameter estimation algorithm for non-linear stochastic systems. *International Journal of Control*, 48(1):193–210, 1988.

[147] S. Chen, S. A. Billings, and W. Luo. Orthogonal least squares methods and their application to non-linear system identification. *International Journal of control*, 50(5):1873–1896, 1989.

[148] H.-L. Wei, S. A. Billings, and J. Liu. Term and variable selection for non-linear system identification. *International Journal of Control*, 77(1):86–110, 2004.

[149] E. I. George and R. E. McCulloch. Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993.

[150] E. I. George and R. E. McCulloch. Approaches for Bayesian variable selection. *Statistica sinica*, pages 339–373, 1997.

[151] T. Baldacchino, S. R. Anderson, and V. Kadirkamanathan. Computational system identification for Bayesian NARMAX modelling. *Automatica*, 49(9): 2641–2651, 2013.

[152] S. L. Kukreja, H. L. Galiana, and R. E. Kearney. A bootstrap method for structure detection of NARMAX models. *International Journal of Control*, 77 (2):132–143, 2004.

[153] T. Baldacchino, S. R. Anderson, and V. Kadirkamanathan. Structure detection and parameter estimation for NARX models in a unified em framework. *Automatica*, 48(5):857–865, 2012.

[154] W. R. Jacobs, T. Baldacchino, and S. R. Anderson. Sparse bayesian identification of polynomial narx models. *IFAC-PapersOnLine*, 48(28):172–177, 2015.

[155] W. R. Jacobs, T. Baldacchino, T. J. Dodd, and S. R. Anderson. Sparse Bayesian nonlinear system identification using variational inference. *IEEE Transactions on Automatic Control*, 2018.

[156] A. Iliopoulos, R. Shirzadeh, W. Weijtjens, P. Guillaume, D. Van Hemelrijck, and C. Devriendt. A modal decomposition and expansion approach for prediction of dynamic responses on a monopile offshore wind turbine using a limited number of vibration sensors. *Mechanical Systems and Signal Processing*, 68: 84–104, 2016.

[157] H. P. Hjelm, R. Brincker, J. Graugaard-Jensen, and K. Munch. Determination of stress histories in structures by natural input modal analysis. In *Proceedings of 23rd Conference and Exposition on Structural Dynamics (IMACXXIII)*, 2005.

[158] U. Tygesen, K. Worden, T. Rogers, G. Manson, and E. Cross. State-of-the-art and future directions for predictive modelling of offshore structure dynamics using machine learning. In *Dynamics of Civil Structures, Volume 2*, pages 223–233. Springer, 2019.

[159] U. T. Tygesen, M. S. Jepsen, J. Vestermark, N. Dollerup, and A. Pedersen. The true digital twin concept for fatigue re-assessment of marine structures. In *ASME 2018 37th International Conference on Ocean, Offshore and Arctic Engineering*, pages V001T01A021–V001T01A021. American Society of Mechanical Engineers, 2018.

[160] E. Reynders. System identification methods for (operational) modal analysis: review and comparison. *Archives of Computational Methods in Engineering*, 19(1):51–124, 2012.

[161] M. C. Kennedy and A. O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63 (3):425–464, 2001.

[162] P. D. Arendt, D. W. Apley, and W. Chen. Quantification of model uncertainty: Calibration, model discrepancy, and identifiability. *Journal of Mechanical Design*, 134(10):100908, 2012.

[163] D. Higdon, C. Nakhleh, J. Gattiker, and B. Williams. A Bayesian calibration approach to the thermal problem. *Computer Methods in Applied Mechanics and Engineering*, 197(29-32):2431–2441, 2008.

[164] B. Andersson, R. Andersson, L. Håkansson, M. Mortensen, R. Sudiyo, and B. Van Wachem. *Computational fluid dynamics for engineers*. Cambridge University Press, 2011.

[165] J. Morrison, M. O'Brien, J. Johnson, and S. Schaaf. The force exerted by surface waves on piles. *Petroleum Transaction*, 189:149–157, 1950.

[166] T. Sarpkaya. *Wave forces on offshore structures*. Cambridge university press, 2010.

[167] J. Bishop. A description of the Christchurch Bay wave force project. *Rep. Natl. Marit. Inst.,(*, (33), 1978.

[168] J. Bishop and R. Ashford. Christchurch bay tower]. *In: Force measurements on structures at sea, 1979,*, page 13, 1980.

[169] J. R. Bishop, R. G. Tickell, K. A. Gallagher, *et al.* The UK Christchurch Bay project; a review of results. In *Offshore Technology Conference*. Offshore Technology Conference, 1980.

[170] J. Bishop *et al.* Wave force data from the second christchurch bay tower. In *Offshore Technology Conference*. Offshore Technology Conference, 1985.

[171] R. Tickell and J. Bishop. analysis of waves forces at the christchurch bay tower. In *Proceedings, 4th International Offshore Mechanics and Arctic Eng Symposium (OMAE)*, pages 142–150, 1985.

[172] G. Najafian, R. Tickell, R. Burrows, and J. Bishop. The UK Christchurch Bay compliant cylinder project: analysis and interpretation of Morison wave force and response data. *Applied Ocean Research*, 22(3):129–153, 2000.

[173] K. P. Murphy. *Machine learning: a probabilistic perspective*. 2012.

[174] A. Etemad-Shahidi, R. Yasa, and M. Kazeminezhad. Prediction of wave-induced scour depth under submarine pipelines using machine learning approach. *Applied Ocean Research*, 33(1):54–59, 2011.

[175] I. Malekmohamadi, M. R. Bazargan-Lari, R. Kerachian, M. R. Nikoo, and M. Fallahnia. Evaluating the efficacy of SVMs, BNs, ANNs and ANFIS in wave height prediction. *Ocean Engineering*, 38(2-3):487–497, 2011.

[176] J. C. Fernández, S. Salcedo-Sanz, P. A. Gutiérrez, E. Alexandre, and C. Hervás-Martínez. Significant wave height and energy flux range forecast with machine learning classifiers. *Engineering Applications of Artificial Intelligence*, 43:44–53, 2015.

[177] K. Worden, P. Stansby, G. Tomlinson, and S. Billings. Identification of nonlinear wave forces. *Journal of fluids and structures*, 8(1):19–71, 1994.

[178] A. Swain, S. Billings, P. Stansby, and M. Baker. Accurate prediction of non-linear wave forces: Part i (fixed cylinder). *Mechanical systems and signal processing*, 12(3):449–485, 1998.

[179] C. C. Aggarwal, A. Hinneburg, and D. A. Keim. On the surprising behavior of distance metrics in high dimensional space. In *International Conference on Database Theory*, pages 420–434. Springer, 2001.

[180] H. Sohn, C. R. Farrar, N. F. Hunter, and K. Worden. Structural health monitoring using statistical pattern recognition techniques. *Journal of dynamic systems, measurement, and control*, 123(4):706–711, 2001.

[181] L. Ljung. *System identification*. Springer, 1998.

[182] S. Särkkä. *Bayesian filtering and smoothing*, volume 3. Cambridge University Press, 2013.

[183] A. H. Jazwinski. *Stochastic processes and filtering theory*. Courier Corporation, 2007.

[184] M. S. Grewal. Kalman filtering. In *International Encyclopedia of Statistical Science*, pages 705–708. Springer, 2011.

[185] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *American Control Conference, Proceedings of the 1995*, volume 3, pages 1628–1632. IEEE, 1995.

[186] S. J. Julier and J. K. Uhlmann. New extension of the Kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition VI*, volume 3068, pages 182–194. International Society for Optics and Photonics, 1997.

[187] E. A. Wan and R. Van Der Merwe. The unscented Kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 153–158. Ieee, 2000.

[188] J. J. LaViola. A comparison of unscented and extended Kalman filtering for estimating quaternion motion. In *American Control Conference, 2003. Proceedings of the 2003*, volume 3, pages 2435–2440. IEEE, 2003.

[189] M. St-Pierre and D. Gingras. Comparison between the unscented Kalman filter and the extended Kalman filter for the position estimation module of an integrated navigation information system. In *IEEE Intelligent Vehicles Symposium*, pages 831–835. Citeseer, 2004.

[190] F. Gustafsson and G. Hendeby. Some relations between extended and unscented Kalman filters. *IEEE Transactions on Signal Processing*, 60(2):545–555, 2012.

[191] N. J. Gordon, D. J. Salmond, and A. F. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140(2):107–113, 1993.

[192] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[193] R. Douc and O. Cappé. Comparison of resampling schemes for particle filtering. In *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, pages 64–69. IEEE, 2005.

[194] M. K. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American statistical association*, 94(446):590–599, 1999.

[195] M. K. Pitt and N. Shephard. Auxiliary variable based particle filters. *Sequential Monte Carlo methods in practice*, pages 273–293, 2001.

[196] C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov Chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.

[197] F. Lindsten, M. I. Jordan, and T. B. Schön. Particle Gibbs with ancestor sampling. *The Journal of Machine Learning Research*, 15(1):2145–2184, 2014.

[198] M. Marcos, F. M. Calafat, Á. Berihuete, and S. Dangendorf. Long-term variations in global sea level extremes. *Journal of Geophysical Research: Oceans*, 120(12):8115–8134, 2015.

[199] F. Lindsten, P. Bunch, S. S. Singh, and T. B. Schön. Particle ancestor sampling for near-degenerate or intractable state transition models. *arXiv preprint arXiv:1505.06356*, 2015.

[200] L. Virgin. *Introduction to experimental nonlinear dynamics*. IOP Publishing, 2001.

[201] F. Moon and P. J. Holmes. A magnetoelastic strange attractor. *Journal of Sound and Vibration*, 65(2):275–296, 1979.

[202] R. Pintelon and J. Schoukens. *System identification: a frequency domain approach*. John Wiley & Sons, 2012.

[203] T. Wigren and J. Schoukens. Three free data sets for development and benchmarking in nonlinear system identification. In *Control Conference (ECC), 2013 European*, pages 2933–2938. IEEE, 2013.

[204] L. Ljung, Q. Zhang, P. Lindskog, and A. Juditsky. Modeling a non-linear electric circuit with black box and grey box models. In *2004 IFAC Symposium on Nonlinear Control Systems, Stuttgart, Germany, September, 2004*, 2004.

[205] V. Verdult. Identification of local linear state-space models: the silver-box case study. *IFAC Proceedings Volumes*, 37(13):393–398, 2004.

[206] M. Espinoza, K. Pelckmans, L. Hoegaerts, J. A. Suykens, and B. De Moor. A comparative study of LS-SVMs applied to the silverbox identification problem. In *Proc. of the 6th IFAC Symposium on Nonlinear Control Systems (NOLCOS)*, 2004.

[207] A. Wills and B. Ninness. Estimation of generalised Hammerstein-Wiener systems. In *Proceedings of the 15th IFAC Symposium on System Identification (SYSID 2009)*, pages 1104–1109. Citeseer, 2009.

[208] A. Marconato, J. Sjöberg, J. Suykens, and J. Schoukens. Identification of the silverbox benchmark using nonlinear state-space models. In *IFAC Proceedings. 16th IFAC Symposium on System Identification*, volume 16, pages 632–637, 2012.

[209] I. Kovacic and M. J. Brennan. *The Duffing equation: nonlinear oscillators and their behaviour*. John Wiley & Sons, 2011.

[210] G. Kerschen, K. Worden, A. F. Vakakis, and J.-C. Golinval. Past, present and future of nonlinear system identification in structural dynamics. *Mechanical systems and signal processing*, 20(3):505–592, 2006.

[211] K. Worden and J. Hensman. Parameter estimation and model selection for a class of hysteretic systems using Bayesian inference. *Mechanical Systems and Signal Processing*, 32:153–169, 2012.

[212] G. Quaranta, G. Monti, and G. C. Marano. Parameters identification of Van der Pol–Duffing oscillators via particle swarm optimization and differential evolution. *Mechanical Systems and Signal Processing*, 24(7):2076–2095, 2010.

[213] J. Butcher. On fifth order Runge-Kutta methods. *BIT Numerical Mathematics*, 35(2):202–209, 1995.

[214] F. Lindsten, T. B. Schön, and M. I. Jordan. Bayesian semiparametric Wiener system identification. *Automatica*, 49(7):2053–2063, 2013.

[215] E. Lourens, C. Papadimitriou, S. Gillijns, E. Reynders, G. De Roeck, and G. Lombaert. Joint input-response estimation for structural systems based on reduced-order models and vibration data from a limited number of sensors. *Mechanical Systems and Signal Processing*, 29:310–327, 2012.

[216] C.-K. Ma, J.-M. Chang, and D.-C. Lin. Input forces estimation of beam structures by an inverse method. *Journal of sound and vibration*, 259(2): 387–407, 2003.

[217] S. E. Azam, E. Chatzi, and C. Papadimitriou. A dual Kalman filter approach for state estimation via output-only acceleration measurements. *Mechanical Systems and Signal Processing*, 60:866–886, 2015.

[218] S. E. Azam, E. Chatzi, C. Papadimitriou, and A. Smyth. Experimental validation of the Kalman-type filters for online and real-time state and input estimation. *Journal of Vibration and Control*, 23(15):2494–2519, 2017.

[219] J. Ching, J. L. Beck, and K. A. Porter. Bayesian state and parameter estimation of uncertain dynamical systems. *Probabilistic engineering mechanics*, 21(1): 81–96, 2006.

[220] S. Gillijns and B. De Moor. Unbiased minimum-variance input and state estimation for linear discrete-time systems with direct feedthrough. *Automatica*, 43(5):934–937, 2007.

[221] S. E. Azam, V. Dertimanis, E. Chatzi, and P. C. Output-only schemes for joint input-state-parameter estimation of linear systems. In *Proceedings of the 1st ECCOMAS Thematic Conference on Uncertainty Quantification in Computational Sciences and Engineering*, pages 497–510, 2015.

[222] M. Alvarez, D. Luengo, and N. Lawrence. Latent force models. In *Artificial Intelligence and Statistics*, pages 9–16, 2009.

[223] M. A. Alvarez, D. Luengo, and N. D. Lawrence. Linear latent force models using Gaussian processes. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2693–2705, 2013.

[224] M. A. Álvarez and N. D. Lawrence. Computationally efficient convolved multiple output Gaussian processes. *Journal of Machine Learning Research*, 12(May):1459–1500, 2011.

[225] M. Hoshiya and E. Saito. Structural identification by extended Kalman filter. *Journal of engineering mechanics*, 110(12):1757–1770, 1984.

[226] L. Ljung. Asymptotic behavior of the extended Kalman filter as a parameter estimator for linear systems. *IEEE Transactions on Automatic Control*, 24(1): 36–50, 1979.

[227] J. Hartikainen and S. Sarkka. Sequential inference for latent force models. *arXiv preprint arXiv:1202.3730*, 2012.

[228] J. Hartikainen and S. Särkkä. Kalman filtering and smoothing solutions to temporal Gaussian process regression models. In *Machine Learning for Signal Processing (MLSP), 2010 IEEE International Workshop on*, pages 379–384. IEEE, 2010.

[229] H. E. Rauch, C. Striebel, and F. Tung. Maximum likelihood estimates of linear dynamic systems. *AIAA journal*, 3(8):1445–1450, 1965.

[230] M. Alvarez and N. D. Lawrence. Sparse convolved Gaussian processes for multi-output regression. In *Advances in neural information processing systems*, pages 57–64, 2009.

[231] M. S. Handcock and M. L. Stein. A Bayesian analysis of kriging. *Technometrics*, 35(4):403–410, 1993.

[232] Z. Ghahramani and G. E. Hinton. Parameter estimation for linear dynamical systems. Technical report, Technical Report CRG-TR-96-2, University of Totronto, Dept. of Computer Science, 1996.

[233] D. D. Klerk, D. J. Rixen, and S. N. Voormeeren. General Framework for Dynamic Substructuring: History, Review and Classification of Techniques. *AIAA Journal*, 2008. ISSN 0001-1452. doi: 10.2514/1.33274.

[234] X. Zhu. Semi-supervised learning. In *Encyclopedia of machine learning*, pages 892–897. Springer, 2011.

[235] G. Manson, K. Worden, and D. Allman. Experimental validation of a structural health monitoring methodology: Part II. novelty detection on a gnat aircraft. *Journal of Sound and Vibration*, 259(2):345–363, 2003.

[236] K. K. Nair and A. S. Kiremidjian. Time series based structural damage detection algorithm using Gaussian mixtures modeling. *Journal of dynamic systems, measurement, and control*, 129(3):285–293, 2007.

[237] E. Figueiredo and E. Cross. Linear approaches to modeling nonlinearities in long-term monitoring of bridges. *Journal of Civil Structural Health Monitoring*, 3(3):187–194, 2013.

[238] J. Kullaa. Structural health monitoring under nonlinear environmental or operational influences. *Shock and Vibration*, 2014, 2014.

[239] E. Figueiredo, G. Park, C. R. Farrar, K. Worden, and J. Figueiras. Machine learning algorithms for damage detection under operational and environmental variability. *Structural Health Monitoring*, 10(6):559–572, 2011.

[240] L. Yu, J.-H. Zhu, and L.-L. Yu. Structural Damage Detection in a Truss Bridge Model Using Fuzzy Clustering and Measured FRF Data Reduced by Principal Component Projection. *Advances in Structural Engineering*, 16(1):207–217, jan 2013.

[241] A. Diez, N. L. D. Khoa, M. Makki Alamdari, Y. Wang, F. Chen, and P. Runcie. A clustering approach for structural health monitoring on bridges. *Journal of Civil Structural Health Monitoring*, 6(3):429–445, jul 2016.

[242] M. M. Alamdari, T. Rakotoarivelo, and N. L. D. Khoa. A spectral-based clustering for structural health monitoring of the Sydney Harbour Bridge. *Mechanical Systems and Signal Processing*, 87:384–400, mar 2017.

[243] D.-A. Tibaduiza, M.-A. Torres-Arredondo, L. Mujica, J. Rodellar, and C.-P. Fritzen. A study of two unsupervised data driven statistical methodologies for detecting and classifying damages in structural health monitoring. *Mechanical Systems and Signal Processing*, 41(1-2):467–484, dec 2013.

[244] R. Langone, E. Reynders, S. Mehrkanoon, and J. A. K. Suykens. Automated structural health monitoring based on adaptive kernel spectral clustering. *Mechanical Systems and Signal Processing*, 90:64–78, 2017.

[245] S. Chen, F. Cerda, P. Rizzo, J. Bielak, J. H. Garrett, and J. Kovacevic. Semi-supervised multiresolution classification using adaptive graph filtering with application to indirect bridge structural health monitoring. *IEEE Transactions on Signal Processing*, 62:2879–2893, 2014.

[246] K. Krishnan Nair and A. S. Kiremidjian. Time series based structural damage detection algorithm using Gaussian mixtures modelling. *Journal of Dynamic Systems, Measurement, and Control*, 129(3):285, 2007.

[247] L. Qiu, S. Yuan, F.-K. Chang, Q. Bao, and H. Mei. On-line updating Gaussian mixture model for aircraft wing spar damage evaluation under time-varying boundary condition. *Smart Materials and Structures*, 23(12), 2014.

[248] L. Qiu, S. Yuan, H. Mei, and F. Fang. An improved Gaussian mixture model for damage propagation monitoring of an aircraft wing spar under changing structural boundary conditions. *Sensors (Basel, Switzerland)*, 16(3):291, 2016.

[249] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007.

[250] A. Vlachos, Z. Ghahramani, and A. Korhonen. Dirichlet process mixture models for verb clustering. *Proceedings of the ICML workshop on Prior Knowledge for Text and Language*, 2008.

[251] D. M. Blei. Probabilistic topic models. *Commun. ACM*, 55(4):77–84, 2012.

[252] C. Wang, J. Paisley, and D. Blei. Online variational inference for the hierarchical Dirichlet process. *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 15:752–760, 2011.

[253] O. Yakhnenko and V. Honavar. Annotating images and image objects using a hierarchical Dirichlet process model. *Proceedings of the 9th International Workshop on Multimedia Data Mining: held in conjunction with the ACM SIGKDD 2008*, pages 1–7, 2008.

[254] B. Thirion, A. Tucholka, M. Keller, P. Pinel, A. Roche, J.-F. Mangin, and J.-B. Poline. High level group analysis of FMRI data based on Dirichlet process mixture models. *Biennial International Conference on Information Processing in Medical Imaging*, pages 482–494, 2007.

[255] A. R. F. da Silva. A Dirichlet process mixture model for brain MRI tissue classification. *Medical Image Analysis*, 11(2):169–182, 2007.

[256] N. Lartillot and H. Philippe. A Bayesian mixture model for across-site heterogeneities in the amino-acid replacement process. *Molecular Biology and Evolution*, 21(6):1095–1109, 2004.

[257] N. Lartillot, N. Rodrigue, D. Stubbs, and J. Richer. PhyloBayes MPI: phylogenetic reconstruction with infinite mixtures of profiles in a parallel environment. *Systematic Biology*, 62(4):611–615, 2013.

[258] F. Wood and M. J. Black. A nonparametric Bayesian alternative to spike sorting. *Journal of Neuroscience Methods*, 173(1):1–12, 2008.

[259] D. Chakraborty, N. Kovvali, A. Papandreou-Suppappola, and A. Chattopadhyay. An adaptive learning damage estimation method for structural health monitoring. *Journal of Intelligent Material Systems and Structures*, 26(2): 125–143, 2015.

[260] W. B. Johnson and J. Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary Mathematics*, 26(189-206):1, 1984.

[261] G. Schwarz *et al.* Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

[262] H. Akaike. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723, 1974.

[263] E. Figueiredo, L. Radu, K. Worden, and C. R. Farrar. A Bayesian approach based on a Markov-chain Monte-Carlo method for damage detection under unknown sources of variability. *Engineering Structures*, 80:1–10, 2014.

[264] J. Diebolt and C. P. Robert. Estimation of finite mixture distributions through Bayesian sampling. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 363–375, 1994.

[265] C. E. Rasmussen. The infinite Gaussian mixture model. *Advances in neural information processing systems*, pages 554–560, 2000.

[266] R. M. Neal. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, 2000.

[267] D. M. Blei and M. I. Jordan. Variational methods for the Dirichlet process. *Proceedings of the twenty-first international conference on Machine learning*, page 12, 2004.

[268] D. M. Blei and M. I. Jordan. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–143, 2006.

[269] M. Seeger. Low rank updates for the Cholesky decomposition. Technical report, University of California Berkley, 2004.

[270] P. E. Gill, G. H. Golub, W. Murray, and M. A. Saunders. Methods for modifying matrix factorizations. *Mathematics of Computation*, 28(126):505–535, 1974.

[271] B. Efron. Bootstrap methods: another look at the jackknife. In J. N. Kotz S., editor, *Breakthroughs in Statistics*, pages 569–593. Springer, 1992.

[272] K. Worden and E. J. Cross. On correlation and causality in structural dynamics. *Proceedings of the 6th European Conference on Structural Control, Sheffield*, 2016.

[273] C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, pages 424–438, 1969.

[274] E. Figueiredo, G. Park, J. Figueiras, C. Farrar, and K. Worden. Structural health monitoring algorithm comparisons using standard data sets. Technical report, Los Alamos National Laboratory (LANL), Los Alamos, NM (United States), 2009.

[275] Y.-L. Zhou, E. Figueiredo, N. Maia, and R. Perera. Damage detection and quantification using transmissibility coherence analysis. *Shock and Vibration*, 2015, 2015.

[276] E. Figueiredo, J. Figueiras, G. Park, C. R. Farrar, and K. Worden. Influence of the autoregressive model order on damage detection. *Computer-Aided Civil and Infrastructure Engineering*, 26(3):225–238, 2011.

[277] R. P. Bandara, T. H. Chan, and D. P. Thambiratnam. Structural damage detection method using frequency response functions. *Structural Health Monitoring*, 13(4):418–429, 2014.

[278] W. Fan and P. Qiao. Vibration-based damage identification methods: a review and comparative study. *Structural health monitoring*, 10(1):83–111, 2011.

[279] C. R. Farrar, S. W. Doebling, and D. A. Nix. Vibration–based structural damage identification. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 359(1778):131–149, 2001.

[280] P. Welch. The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. *IEEE Transactions on audio and electroacoustics*, 15(2):70–73, 1967.

[281] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is nearest neighbor meaningful? In *International conference on database theory*, pages 217–235. Springer, 1999.

[282] O. Cappé. Online expectation maximisation. *Mixtures: Estimation and Applications*, pages 31–53, 2011.

[283] Y. C. Eldar and G. Kutyniok. *Compressed Sensing: Theory and Applications*. Cambridge University Press, 2012.

[284] S. Dasgupta and A. Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003.

[285] E. J. Candes and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE transactions on Information Theory*, 52 (12):5406–5425, 2006.

[286] K. Worden, C. R. Farrar, J. Haywood, and M. Todd. A review of nonlinear dynamics applications to structural health monitoring. *Structural Control and Health Monitoring*, 15(4):540–567, 2008.

[287] J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[288] E. Snelson and Z. Ghahramani. Variable noise and dimensionality reduction for sparse Gaussian processes. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, pages 461–468. AUAI Press, 2006.

[289] J. A. Hoeting, D. Madigan, A. E. Raftery, and C. T. Volinsky. Bayesian model averaging. In *Proceedings of the AAAI Workshop on Integrating Multiple Learned Models*, volume 335, pages 77–83. Citeseer, 1998.

[290] P. Jylänki, J. Vanhatalo, and A. Vehtari. Robust Gaussian process regression with a student-t likelihood. *Journal of Machine Learning Research*, 12(Nov): 3227–3257, 2011.

[291] E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. *Advances in neural information processing systems*, 18:1257, 2006.

[292] T. D. Bui, C. Nguyen, and R. E. Turner. Streaming sparse Gaussian process approximations. In *Advances in Neural Information Processing Systems*, pages 3299–3307, 2017.

[293] M. Ismail, F. Ikhouane, and J. Rodellar. The hysteresis Bouc-Wen model, a survey. *Archives of Computational Methods in Engineering*, 16(2):161–188, 2009.

[294] E. B. Flynn and M. D. Todd. A Bayesian approach to optimal sensor placement for structural health monitoring with application to active sensing. *Mechanical Systems and Signal Processing*, 24(4):891–903, 2010.

[295] E. Papatheou, R. J. Barthorpe, and K. Worden. An experimental investigation of feature availability in nominally identical structures for population-based shm. In *Structural Health Monitoring and Damage Detection, Volume 7*, pages 185–191. Springer, 2015.