On the Application of the Reversible Jump Markov Chain Monte Carlo Method within Structural Dynamics

A Thesis submitted to the University of Sheffield

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

by

Oana-Daniela Tiboaca

Department of Mechanical Engineering

University of Sheffield

December 2016

Table of Contents

Abstract

System Identification (SID) is an important area of structural dynamics and is concerned with constructing a functional relationship between the inputs and the outputs of a system. Furthermore, it estimates the parameters that the studied system depends upon. This aspect of structural dynamics has been studied for many years and computational methods have been developed in order to deal with the system identification of real structures, with the aim of getting a better understanding of their dynamic behaviour.

The most straightforward classification of structures is into structures that behave linearly and structures that behave nonlinearly. Even so, one needs to keep in mind that no structure is indefinitely linear. During its service, a structure can behave nonlinearly at any given point, under the right working and environmental conditions.

A key challenge in applying SID to real systems is in handling the uncertainty inherent in the process. Uncertainty arises from various sources such as modelling error and measurement error (noisy data), resulting in uncertainty in the parameter estimates. One of the ways in which one can deal with uncertainty is by adopting a probabilistic framework. In this way one admits the limitations in the process of SID through providing probability distributions over the models and parameters of interest, rather than a simple 'best estimate'.

Throughout this work a Bayesian probabilistic framework is adopted as it covers the uncertainty issue without over-fitting(it provides the simplest, least complex solutions to the issue at hand). Of great interest when working within a Bayesian framework are Markov Chain Monte Carlo(MCMC) sampling methods. Of relevance to this research are the Metropolis-Hastings(MH) algorithm and the Reversible Jump Markov Chain Monte Carlo(RJMCMC) algorithm.

Markov Chain Monte Carlo(MCMC) methods and algorithms have been extensively investigated for linear dynamical systems. One of the advantages of these methods being used in a Bayesian framework is that they handle uncertainty in a principled way. In recent years, increasing attention has been paid to the role nonlinearity plays in engineering problems. As a result, there is an increasing focus on developing computational tools that may be applied to nonlinear systems as well as linear

systems, with the objective that they should provide reliable results at reasonable computational costs. The application of MCMC methods in nonlinear system identification(NLSID) has focused on parameter estimation. However, often enough, the model form of systems is assumed known which is not the case in many contexts(such as NLSID when the nonlinearity is hard to identify and model, or Structural Health Monitoring when the damage extent or number of damage sites is unknown).

The current thesis is concerned with the development of computational tools for performing System Identification in the context of structural dynamics, for both linear and nonlinear systems.

The research presented within this work will demonstrate how the Reversible Jump Markov Chain Monte Carlo algorithm, within a Bayesian framework, can be used in the area of SID in a structural dynamics context for doing both parameter estimation and model selection. The performance of the RJMCMC algorithm will be benchmarked throughout against the MH algorithm. Several numerical case studies will be introduced to demonstrate how the RJMCMC algorithm may be applied in linear and nonlinear SID; and one numerical case study to demonstrate application to a SHM problem. These will be followed by experimental case studies to evaluate linear and nonlinear SID performance for a real structure.

Acknowledgments

Most importantly I want to thank all the people that were always there for me no matter what. My mother, Petruta was such an inspiration throughout my years. I have never met a person as strong as my mother, who dedicated her entire life to see us through. I would like to thank her and offer her my uttermost respect for being the woman that she is today. Secondly, my sister Andreea. There was no tantrum too big for her, there was no road that she was not there for me, kicking me to keep on climbing. My best friend Cristina, I don't know how I would've survived all the craziness of my PhD without you showing me that there is still so much more craziness left to enjoy! My brother, Iulian for offering me the best advice that kept me going for the length of these past years, for being the father figure I would've never hoped for. My friend Kartik that made every single day a day to laugh and gave me the advice that I always needed! The amazing Niloufar that dragged me out of the office for the Wednesday pint, when I didn't even realise I needed to take a break. Thank you all for offering me so much inspiration to get me to the point I am today. All my nephews and nieces are forever in my heart for the simple words that made me go forward and made me think that the world is a better place for the fact that they exist. For the rest of my extended family, you are simply amazing and simply the best, I don't know how I got so lucky to be surrounded by you!

I would like thank Dr Robert Barthorpe for the supervision and for all the help offered throughout the PhD period. I would like to thank Dr Peter Green, my second supervisor that always gave me a reason to question my thinking and helped me out whenever it felt like I was completely stuck. I would like to thank Prof. Keith Worden for showing me that kindness still exists and Dr Ifigeneia Antonidoiu for taking over my secondary supervision. Leslie Morton, there was no one more dedicated to getting all of my rigs up and running!

I would like to thank all of my colleagues, present and past for giving me an amazing environment and making me feel like home whenever I felt lonely.

I would like to show my respect to my late father that gave me so many reasons to keep on fighting in spite of him.

Last but not least, I would like to thank my fiance, Patrick, for joining this bumpy ride of mine when I needed him the most. Thank you so much for always steering me in the right direction!

Without all the people mentioned I would have never been to the standard I am today so once again Thank you, Thank you, Thank yoooouuuuu!!

Nomenclature

$\boldsymbol{\theta}$ Vector of parameters

$\boldsymbol{\theta}'$ Proposed vector of parameters

$\boldsymbol{M}$ Set of candidate models, $\boldsymbol{M} = \{ M^{(1)}, M^{(2)},...,M^{(l)}\}$

$\boldsymbol{\theta^{(l)}}$ Vector of parameters for model $M^{(l)}$

$\pi(\boldsymbol{\theta})$ Target PDF

$Q(\boldsymbol{\theta}'|\boldsymbol{\theta})$ Proposal PDF

$T(\boldsymbol{\theta} \rightarrow \boldsymbol{\theta}')$ Transition function

$\alpha(\boldsymbol{\theta}'|\boldsymbol{\theta})$ Acceptance probability

D Measured data

$P(\boldsymbol{\theta}|M^{(l)})$ Prior distribution

$P(D|\boldsymbol{\theta}, M^{(l)})$ Likelihood

$P(D|M^{(l)})$ Evidence term

$P(\boldsymbol{\theta}|D, M^{(l)})$ Posterior distribution

m Mass

c Damping coefficient

k Linear stiffness

$F(t)$ Forcing

D_l Damage in $M^{(l)}$

R_l Reduction in stiffness in $M^{(l)}$

$\boldsymbol{u}$ Vector of randomly generated dummy variables to allow transition between models

$\boldsymbol{u}'$ Vector of randomly generated dummy variables to allow transition between models

$g(\boldsymbol{u})$ PDF from which the random numbers $\boldsymbol{u}$ are generated

$g'(\boldsymbol{u}')$ PDF from which the random numbers $\boldsymbol{u}'$ are generated

λ dummy parameter generated from a normal distribution in order to allow dimension matching between models

μ Constant used in the RJMCMC algorithm in order to create the mapping between models

σ standard deviation

h Mapping to go forward and backward between models

b_l Probability of birth move for model M^l

d_l Probability of death move for model M^l

u_l Probability of update move for model M^l

p Constant that adjusts the update move in relation with the birth and death moves

α_m Acceptance probability of birth move

α'_m Acceptance probability of death move

f_s Sampling frequency

h_t Time step size

Chapter 1

Introduction

System identification is of great importance in the context of structural dynamics. The aim of system identification is to robustly characterise a real structure using mathematical models and experimental data, in order to predict its dynamic behaviour. This process breaks into two main areas: identification of the model structure and estimation of parameters that cannot be directly measured. Possessing knowledge about the dynamic behaviour of a structure is helpful in the engineering world as one can determine for example the excitation a structure can withstand before damage or failure occurs or the maximum displacement a structure can withstand,etc.

1.1 Motivation

The dynamics of a structure tells one about its operational requirements and limitations. Structural dynamics engineers, in research as well as industry, are interested in understanding dynamical behaviour in order to improve existing structures or make new, more efficient, reliable ones.

In order to understand the dynamic behaviour of structures, one needs to start small. The path to knowledge is done in steps, starting with an idea or concept, followed up by simulations and applications on simulated scenarios and only then experimental structures or demonstrators.

Bayesian Inference as a probabilistic framework has found particular success when applied in the field of System Identification (SID). In particular, in doing parameter estimation on both linear and nonlinear systems. Even so, it is considered to be a relatively new approach that still gives way to many discussions in the engineering community.

Uncertainty is one of the reasons why one is inclined to use a probabilistic framework in studying a systems 'definition', i.e. the modelling and estimating of parameters. Bayesian Inference is a natural choice when wanting robust models within a framework that penalises overfitting.

When dealing with a variety of models and parameters that could have complex geometrically defined probability distributions, the Bayesian approach is helped by Markov Chain Monte Carlo (MCMC) sampling algorithm. MCMC sampling methods are yet another powerful computational tool in SID.

Another issue when doing SID is choosing a model for the structure of study. There could be a number of models to choose from with varying number of parameters. Applying a Bayesian approach together with traditional MCMC algorithms is not necessarily enough when it comes to moving between models of varying dimension. The best that can be done is to apply such algorithms to each candidate model in turn, then asses the 'evidence' for adopting the respective models. This leads to a desire for a computational tool that could do model selection by 'jumping' between models of varying dimension. One such algorithm, that can jump between spaces of varying dimensions, is the Reversible Jump Markov Chain Monte Carlo (RJMCMC) algorithm [34].

1.2 Aims and Objectives

Driven by the necessity of new computational tools for doing nonlinear system identification in the context of dynamical systems, the aim of this thesis is to concentrate on one particular MCMC algorithm, RJMCMC, in order to introduce and demonstrate the capabilities of a comparatively new and untapped MCMC method which, employed within a Bayesian framework, can do simultaneous parameter estimation and model selection for both linear and nonlinear structural systems.

A set of objectives were decided upon, in order to meet the desired aim:

- Introduce the RJMCMC method to be used in the context of structural dynamics in order to do system identification of linear and nonlinear structures in a probabilistic framework, by simultaneously providing estimates of the parameters of a system and selecting a robust model to best fit the available data set;
- Adapt and apply the RJMCMC algorithm on a series of numerical case studies of SDOF(single degree of freedom) and MDOF(multi-degree of freedom) scenarios in order to evaluate its capabilities for performing system identification of linear and nonlinear structures;
- Adapt and apply the RJMCMC algorithm on experimental case studies in order to emphasize its capabilities of simultaneously do parameter estimation and model selection for both linear and nonlinear systems;
- Discuss the RJMCMC method in the context of SID of dynamical systems by underlining its advantages and considering its drawbacks;

1.3 SID, Bayesian Inference and MCMC methods

There are many ways in which one can define the problem of identifying a system. Most of them are closely related to how one classifies the issue at hand. It is generally accepted that system identification is split into two main problems: parameter estimation and model selection. However, at a closer look, identifying a system is an even more complex issue. Starting from the system of interest, one needs to specify if this is a theoretical system(is it based on simulations) or an experimental one [1]. In the case of a theoretical system, the structure is always known while in an experimental system the structure might be unknown as well. The theoretical system is based on a description of the structure that obeys the laws of physics and that will be expressed in a mathematical format, depending on a set of parameters that usually have physical meaning. On the other hand, when it comes to an experimental system, one is dealing with a real structure, surrounded by unknowns and uncertainties. The modelling problem is dependant on what type of structure it is; whether the model form is known or unknown; and whether it is parametric or non-parametric identifiable(i.e. the model represented by parameters with physical

meaning, or is it defined by purely statistical parameters with no physical interpretation). The experimental modelling is based on having some sort of information or data from a structure, i.e. input and output data and modelling what happens in between them [1].

If one chooses to go even further down the ladder of classifications, one will find that the identification problem can be sorted according to what kind of modelling one chooses and that in itself it depends on the chosen approach between the theoretical modelling and the experimental modelling. There are three main classes:

- white box models;
- grey box models;
- black box models;

True white box modelling can only belong to the theoretical modelling class as it assumes that the physical laws that surround the system are known and also are the parameters. In the case where one knows the physical laws but not the parameters or the model structure of the system and deals with measurable input-output signals, the problem can be placed in the grey box model class. The final class, black box modelling, is based entirely on measured data. In this case, data from the real system is employed in identifying the model. In this case, it is assumed that only the input-output signals are known because they are measurable,with no further assumptions made on the form of the model. Such models have limited predictive capabilities [1].

An important aspect that needs to be understood is that there is no such thing as a full and complete identification of a system. The structures and systems are identified only to a certain extent, taking into consideration only the instant in time they were analysed at and the excitation they were put under in that instant in time. A structure can be put under a range of excitations differing from what the user might think it is going to experience in use. In real life those ranges can be exceeded and most times that is when one deals with nonlinear effects.

When conducting system identification, one's aim is to use measured data in order to compute or refine the mathematical model of a real system. SID can be conducted in three main ways[2]:

- modal parameter identification;

- structural-model parameter identification;
- control-model identification.

For the purpose of this research, interest is focused on the second method of conducting SID, that of structural-model parameter identification. In order for the identification process to be conducted, one makes use of data that includes information on input and output, where input refers to the excitation introduced into the system/structure and output refers to the measured response of the system/structure after the excitation was applied(examples: displacement, velocity, acceleration). Arguably the greatest challenge faced when applying SID is uncertainty. Uncertainty could arise from either unknown,ignored or/and compressed information. This information can be either from the studied system itself, conducted experiments or processing errors. Examples of situations where one can encounter uncertainty are when modelling the system, due to the differences between the model and reality, during the estimation process due to the fact that one or more parameters are unknown or during the measuring of the data due to noise. Uncertainty is usually categorised under either aleatoric uncertainty or epistemic uncertainty [4]. Epistemic uncertainty or otherwise called systematic uncertainty can arise for example, from applying modelling approximations in order to conserve simplicity [5], [6]. Systematic errors in measurement procedures fall under the same category of epistemic uncertainty. Variation of model parameters may in itself lead to epistemic uncertainty. When modelling through computational methods, one could assume fixed values for parameters when in reality, during the physical experiment the parameter values are most likely varying under changing vibration ranges. Engineers tend to concentrate on reducing epistemic uncertainty. The second category of uncertainty, aleatoric uncertainty, is also known as statistical uncertainty. This category contains all uncertainties that cannot be reduced. Aleatoric uncertainties could be present due to limitations in used machinery or data acquisition systems. An example of aleatoric uncertainty would be noise. When the noise level cannot be controlled any further, it falls under aleatoric uncertainty. Engineers tend to accept the aleatoric uncertainty as it is due to lack of possibilities of reducing it [7]. The sources of uncertainty are many and differ from one case to another. Some of the main sources of uncertainty are:

- Parameter uncertainty - looking at the issue of parameter estimation from a Bayesian point of view, leads to a high dependency between the resulting estimates and the data used. The uncertainty arises from having problems with

the collected data, such as the data containing not enough necessary information to describe the dynamic behaviour of the structure or even something as basic as insufficient data points [4];

- Model uncertainty - models are used to mathematically approximate relationships between inputs and outputs. They are decided by a user and as such, uncertainties occur. Models are put together by using parameters that the user believes the functional is dependent on. Parameters bring their own uncertainty into the model. Moreover, the user might not always know the exact dependency of responses of a system on parameters, for which extra research would need to be conducted (i.e. sensitivity analysis);

- Experimental uncertainty - the gathered data cannot be known precisely. There are sources of noise in an experimental setting, either from cables, rattling from bolts, etc. There are also instrumental uncertainties from the equipment used, measurement error, all of which need to be taken into account.

Whatever the provenance, uncertainty can affect the results of system identification, sometimes invalidating them [3]. For this reason, rather than attempting to know exactly what the parameters of a system are or how the system is mathematically modeled, one is trying to provide a probabilistic estimation of the parameters of interest and an approximation of the real system through a mathematical model.

1.3.1 System Identification Applications in Structural Dynamics: Linear system, Nonlinear system and Damage Identification

A general linear system that is also linear in parameters is commonly represented as:

$$M\ddot{y}(t) + C\dot{y}(t) + Ky(t) = F(t) \tag{1.1}$$

where M, C and K are the mass, damping coefficient and stiffness matrices respectively, $\ddot{y}(t)$ is acceleration, $\dot{y}(t)$ is velocity and $y(t)$ displacement and $F(t)$ is the introduced excitation or forcing.

Systems are commonly considered to be either linear or nonlinear in nature, and the way a system is classified will dictate the SID approach that may be applied to it. System identification of linear systems has been well dealt with in the past by using the principle of superposition, reciprocity and homogeneity [8]. When it comes to nonlinear systems, these principles and rules fail. For example, in a nonlinear system, there is no guarantee that by interchanging the positions of measuring input and output will result in the same frequency response function(FRF). For superposition, it cannot be assumed that the sum of inputs into a nonlinear system is the sum of their individual outputs. This makes nonlinearity hard to deal with. Typically, in the engineering industry, one considers structures to be linear or behaving linearly because there are well developed tools that can deal with linear problems. Most real structures though have a degree of nonlinearity in their behaviour. Some common examples of nonlinearity are friction, contacts and stiffening/softening effects. Nonlinearity is increasingly being considered as part of the reason why structures fail. For this reason, researchers try to find more ways in which one can understand nonlinearity. Not taking into account the nonlinearity in a structure can lead to errors in the prediction of its dynamic behaviour. Using those results in the modelling procedure can have catastrophic outcomes. There are many examples of failures in engineering because the dynamic behaviour of real structures was wrongly predicted due to the use of a linear model for nonlinear behaviour. A famous example would be the Tacoma Narrows bridge which failed due to aeroelastic flutter, a nonlinear phenomena that was not accounted for during design [9]. Unexpected vibration regime was at the core of NASA Helios Prototype's failure [10]. The structural design of the Helios UAV(Unmanned Aerial Vehicle) was not able to withstand prolonged turbulence which lead to the solar powered electric aircraft's disintegration. Yet another example of unexpected dynamic behaviour is the excessive environmental induced vibration of the Dongting Lake bridge in China [11]. The structural design of the Dongting Lake bridge relies on cables which, in practice, are subjected to high vibrational regimes during periods of intense wind and rain. These inputs were not accounted for during design. In order to suppress these unwanted effects, a retro-fitted MR(Magnetorheological) damping system was necessary. From the industrial point of view, understanding nonlinearity has beneficial potential through improved design, reduction of unexpected failures or reduction in need for inspection.

A general nonlinear system will look like:

$$M\ddot{y}(t) + C\dot{y}(t) + Ky(t) + \beta f(y, \dot{y}) = F(t) \tag{1.2}$$

where the new introduced term $\beta f(y, \dot{y})$ represents the nonlinear functional that is dependent on displacement and velocity. The identification process now involves the nonlinearity which means that the first step would be to characterise the nonlinearity, followed by finding its functional form and determining its parameters, β. Some examples of characterised nonlinearities are either related to the displacement, such as bilinear stiffness or cubic stiffness or related to the velocity, eg. quadratic damping, Coulomb friction or even a combination of the two [8]. Because the nonlinearities have many different reasons for appearing, often one can get stuck at the first step of characterising the nonlinearity, i.e. identifying what type of nonlinearity is present and where it is located. Even if one manages to characterise it, modelling the nonlinearity is also very difficult. There are developed methods to identify nonlinearity but the issue is that they do not apply to all types of nonlinearity. There is no developed general method valid for any type of nonlinearity present in a structure. Because of its complexity and due to the fact that linear rules, as the principle of superposition, do not apply to nonlinear system, one is restricted to study only a small number of degrees of freedom which is quite irrelevant in real engineering problems where structures are made out of a high number of degrees of freedom.

The approach followed in the present thesis is off-line identification, which in simple terms means that the data was gathered from the structure under certain excitation conditions and then processed on a computer. The opposite to this approach is online identification which means that the identification process is run as the gathering of the data is happening.

Probably as important as system identification is the detection, characterization and location of damage in dynamical structures. While knowing a structures dynamic behaviour helps in determining its usage limitations, detecting damage in a structure can prevent those limits being reached or structure failure. A great deal of interest has been paid to treating the damage identification task as one of model updating with both deterministic and non-deterministic updating methods having been extensively investigated.

1.3.2 Bayesian Inference

Unfortunately, for all areas of structural dynamics the quantification of uncertainty is a constantly occurring problem. If one thinks about the experimental modelling, one could get uncertainty from both input and output signals. One of the most attractive ways of dealing with uncertainty is by treating the issue from a probabilistic point of view. In this sense the end user is presented with an area of possible solutions. This is naturally presented through probability distributions. Even more so, there is particular interest in the methods that come with Bayesian inference. There has been substantial research focus on using a Bayesian framework for probabilistic analysis due to the fact that it prevents overfitting [12]. This means that in the process of model selection within SID, a Bayesian analysis will provide a robust and the least complex model as a solution. Bayesian inference has its roots in Bayes' theorem which will be seen again and fully discussed in following chapters.

In a general sense, Bayes' theorem may be expressed as:

$$P(A|B,C) = \frac{P(B|A,C)P(A|C)}{P(B)} \tag{1.3}$$

The Bayes' theorem makes sense as:

- the posterior distribution, $P(A|B,C)$, of the variables of interest(A);
- the conditional probability distribution, $P(A|C)$, of A given the event C happened;
- the likelihood, $P(B|A,C)$, of the event B happening given A exists and event C happened;
- the probability, $P(B)$, of the event B happening or existing.

In a system identification scenario, the events in Bayes' theorem would be expressed as $A = \boldsymbol{\theta}$, where $\boldsymbol{\theta}$ is the vector of parameters, $B = \boldsymbol{M}$, where $\boldsymbol{M}$ is the set of possible models and $C = D$, where D is the available data.

Of particular interest, especially for the purpose of this thesis, is how one gets the probability distributions of interest. Firstly, it is needed to know that the interest lays in getting the posterior probability distribution of parameters and models,

depending on the task at hand. Now, probability distributions come in various forms and complexities and for the most complex geometries, one cannot just simply evaluate their respective integrals due to them becoming mathematically intractable. When that is the case, a possible solution is using the Markov Chain Monte Carlo(MCMC) sampling algorithms. These methods gained popularity in being used within a Bayesian framework due to the fact that they allow sampling from complex probability distributions.

These mathematical computational tools are used for both parameter estimation and model selection issues. Many of them work only in special cases, others can only be employed for the parameter estimation issue,etc. There are only a few that are either built or can be modified for dealing with both parameter estimation and model selection.

One such algorithm is the Reversible Jump Markov Chain Monte Carlo(RJMCMC). This algorithm gained attention for its capabilities of doing simultaneously parameter estimation and model selection on linear as well as nonlinear systems.

The RJMCMC algorithm is based on sampling from the joint probability of both parameters and models, as expressed in equation (1.4). In this case, one needs to know the prior probability distribution of models as well (let one consider that $\mathbf{M}$ is the set of possible models). For clarity, there are others MCMC sampling algorithms that can perform model selection but it is applied as an 'add on' to doing parameter estimation. Basically, a by-product of identifying the estimates of parameters is an evaluation of the evidence for the model. This provides one with an indication of the preferred model to best fit the data, once parameter estimation has been conducted on each individual model first.

$$P(A, M|C) = \frac{P(C|A, M)P(A|C)P(M|C)}{P(C)} \tag{1.4}$$

The present thesis aims to show how the algorithm presents itself as a novel tool for nonlinear system identification as well as other areas of structural dynamics, such as structural health monitoring.

1.4 Scope and Outline of thesis

This section provides short summaries for the chapters of this thesis. The thesis is structured so that the reader gets a flavour of both the system identification and structural health monitoring research done simultaneously, research that gets united through the Reversible Jump Markov Chain Monte Carlo algorithm.

Chapter 1 provides an introduction to structural dynamics in the areas of System Identification together with an overview of what methods were chosen to deal with the issue of SID. The motivation for this research is given, followed by the outline of the thesis.

Chapter 2 provides the background and a brief critical analysis of work that has been done in System Identification by employing a Bayesian framework so far, work that motivated the present thesis and also includes the literature review of the most relevant previous studies.

Chapter 3 is concerned with covering the theory of the Bayesian probabilistic framework and the Markov Chain Monte Carlo(MCMC) sampling methods. It provides a description of MCMC algorithms and how they fit into the Bayesian framework as well as a more in depth description of Bayesian Inference.

Chapter 4 provides a detailed description of all principles of the Reversible Jump Markov Chain Monte Carlo(RJMCMC) algorithm and how it may be applied in a structural dynamics context.

Chapter 5 contains a series of numerical case studies. This section of the thesis introduces and provides the results of the numerical case studies for applying RJMCMC in SID and SHM.

Chapter 6 provides the results of applying the RJMCMC algorithm on experimental data.

Chapter 7 summarises the thesis and provides conclusions and discussions on the topics of the thesis. This is followed by the Appendix section where the relevant derivations are presented.

Chapter 2

Background to System Identification - Literature Review

Chapter 2 covers the relevant background in system identification, specifically targeted on doing SID within a Bayesian framework by using the Markov Chain Monte Carlo sampling algorithms. There is extensive literature available for structural dynamics and for the existing methods of doing SID in the dynamical context. Some of the relevant ones will be reviewed in the present chapter. The literature review is conducted with the aim of presenting the work done so far in Bayesian SID for parameter estimation and model selection, with the aim of emphasising the need for the RJMCMC computational method. Some of the relevant work done in structural dynamics within the context of system identification using Bayesian inference and MCMC sampling methods is summarised next.

2.1 Structural Dynamics and MCMC Sampling Methods

MCMC sampling methods have been widely used in most research areas. Their wide application makes them even more attractive to researchers. Until 1990, MCMC al-

gorithms were employed mainly in the areas of chemistry and physics. After 1990, the algorithms were introduced into statistical mathematics which provided an opportunity for them to be used in other research areas such as signal processing, structural dynamics, etc. [13].

Bayesian inference became popular as a probabilistic framework due to its capability of making use of prior knowledge but mainly due to the fact that it allows a study of uncertainty. Its most recent use in the area of structural dynamics had a boom when the MCMC sampling algorithms surfaced. The sampling necessary in getting the posterior probability distribution within a Bayesian framework became achievable with the MCMC methods.

As the present work is concerned with using MCMC methods in the context of structural dynamics, a review of work done in this particular area will be detailed further. Bayesian inference has been employed with MCMC sampling methods in many relevant works. One particular work of high importance is conducted by Beck et al. in [14]. In fact, the contributions of Beck and co-workers to the field of Bayesian SID cannot be overstated due to their high implication in using MCMC algorithms with Bayesian Inference; however, it is not the intention to present a detailed overview of their work here. Beck et al. [14] proposed using MCMC methods in order to get "regions of concentration" of the posterior probability distribution as explained by Bayes theorem. In order to do so, they applied the MH algorithm to a locally identifiable model (multiple optimum parameter vectors) and to a locally unidentifiable model (continuum of optimum parameter vectors). In order to improve convergence of the algorithm, a version of Simulated Annealing was applied. As expected, paper [14] provides a discussion on uncertainty and reliability and possible ways of overcoming it. The algorithm introduced in [14] offered good results in SID when used on two models, but without a proper concentration of resources on the problem of model selection.

MCMC sampling algorithms within a Bayesian framework have been used for doing parameter estimation for nonlinear systems in the past, but not directly model selection. One example of such work is by K. Worden et al. [15]. The authors discussed the use of the MH algorithm on two models, a 'Bouc-Wen' hysteresis model and a nonlinear model of Duffing type. The algorithm was used for the issue of parameter estimation and the Deviance Information Criterion was employed in order to conduct model selection. This can prove time consuming and computationally

expensive. A discussion on the correlation of the parameters was conducted. Even though extremely relevant in doing nonlinear system identification, the work done in [15] does not cover doing model selection through MCMC methods.

In [16] the author provides a straightforward overview of the importance of Bayesian probabilistic framework in structural dynamics. The MCMC sampling algorithms are promoted for doing both parameter estimation and model selection in a robust way. Another added benefit of using the Bayesian framework is as explained in [16], the fact that it can be used not only with MCMC methods but also Laplace's method. The paper does not include a particular application of Bayesian inference but nonetheless is a powerful work in promoting the use of the Bayesian framework and MCMC sampling algorithms.

A more thorough work is presented in [17]. The contribution explains doing model selection and updating between classes of hysteretic models using a fairly new MCMC algorithm, the Transitional Markov Chain Monte Carlo method, which was firstly introduced by Chen et al. in [18]. A good description is provided regarding globally identifiable, locally identifiable and unidentifiable models. The TMCMC method is illustrated using simulated seismic data. The advantage of using the TMCMC algorithm is that one can do model selection by comparing the evidence of the models considered. A drawback to the approach is that the computational cost increases considerably once one has to consider more than two models. Also, it requires post-processing of the data.

In [19], the authors propose using the Gibbs sampler as a starting point for doing model selection. The models considered are of varying dimension, multiple changepoint models and non-nested models. The authors make use of such called pseudopriors, which are considered problematic in literature due to the drawback of being associated with slow convergence of the Markov chains. Conditional probability distributions need to be defined due to the fact that the authors chose to use the Gibbs sampler.

A thorough explanation of the Metropolis-Hastings algorithm in particular is given in [20]. The authors' purpose in their paper is to provide a tutorial on one of the most popular MCMC sampling algorithms. A discussion on its drawbacks, the irreversibility of the Markov chains and the choice of proposal is conducted. Proposals are crucial in MCMC sampling algorithms. They decide where the chain

goes next. A bad proposal can lead to an inappropriate acceptance rate (too high or too low). In [21], Grazian et al. are treating the issue of slow convergence and low acceptance ratio in the Metropolis-Hastings algorithm. The proposed method is making use of the concept of delayed acceptance of the proposed samples and it is promoted as an improved MH algorithm that can deal with "Big Data". It follows that in [21] the concentration is on estimating parameters and not selection between model classes.

A few examples of using MCMC sampling methods and Bayesian inference for nonlinear system identification can be found in [22–27].

In [22] the author applied two different MCMC sampling algorithms, the Simulated Annealing method and the Hybrid Monte Carlo on a SDOF nonlinear system, with the nonlinearity coming in the form of Coulomb damping. The paper is presented as a tutorial on a simulated case, with the concluding remarks that the Simulated Annealing method is more appropriate in that particular scenario. On the other hand, the Simulated Annealing method on its own is incapable of exploring large parameter spaces and it is a clear drawback to this method.

In [23], the authors of this particular work used real data in order to conduct SID on nonlinear systems with nonlinearities of friction and stiffness.

Simulated annealing and data annealing are two often encountered algorithms in the MCMC literature, some examples being [24] and [25]. The purpose of Simulated Annealing is to make sure that a Markov Chain does not get stuck in a local maximum region. Green [24] uses the Shannon entropy to gradually introduce data into the algorithm in order to provide a smooth transition from the prior probability distribution of parameters to the posterior distribution of parameters. The method is successfully applied in doing system identification on nonlinear dynamical systems and it is illustrated using simulated data. In [26],it is proposed to do system identification of nonlinear dynamical systems using what is known as 'highly informative training data'. The Shannon entropy is used to decide the amount of information the data contains. The method is employed on a simulated 3-DOF mass-spring-damper system. The paper outlines one of the issues of this method, the assumption that there exists only one set of parameters that is suitable. Both studies discussed above have applied the approaches proposed on experimental data. In [27], Green et al. conduct an overview of employing MCMC sampling algorithm within a Bayesian framework for doing nonlinear system identification. The work presented in [27] is

a tutorial and concentrates on the advantages of Bayesian inference and MCMC methods.

The most recent work in the area of nonlinear system identification of dynamical structures, at the moment of writing, is by Noël and Kerschen, [28]. The paper is a review of the progress done in the last decade in nonlinear system identification, in the context of structural dynamics, aiming only at parameter estimation issues. Clearly a drawback is the lack of a review on model selection methods as well. The methods of restoring force surface (RFS), nonlinear normal modes(NNM), bifurcation theory are applied on experimental data gathered from a SmallSat spacecraft that displays nonlinear behaviour. The clearance and stiffness coefficients are considered for parameter estimation and the results for each method are presented accordingly.

The following section is concerned with the RJMCMC algorithm which was introduced in 1995 by Peter J. Green in paper [34]. The author introduced the RJMCMC algorithm as an extension to the MH algorithm with the added benefit of being able to do parameter estimation and model selection at the same time. In order to do so, Green proved that the Markov chain created by his algorithm obeys the principle of detailed balance while jumping between different parameter spaces. Another work by the same author [35], explains the RJMCMC algorithm in further detail.

2.2 General Research and RJMCMC

Having [34] as a starting point, many researchers adapted the RJMCMC algorithm. An example of such work is [37] where the authors chose to use the Gibbs sampler rather than the MH algorithm at the core of the RJMCMC algorithm (the method was used on logistic and simulated regression problems).

The RJMCMC algorithm found its use on NARMAX(Nonlinear Autoregressive Moving Average with eXogenous input models) models as well [36]. The authors of [36] used the algorithm in order to present a comparison between the forward regression method and the RJMCMC, with the latter having several benefits.

Nuclear physicists have taken the opportunity to employ the RJMCMC algorithm as an alternative to linear and nonlinear least squares techniques [31]. The in-

terest of the authors of [31] was to use MCMC methods in order to quantify the isotopic content of radioactive material. The methods proved successful in nuclear spectroscopy but the authors brought to attention the fact that when it comes to MCMC algorithms, the processing time can become an issue.

The RJMCMC algorithm was employed heavily in the biology field. Issues such as recombination rates in biological processes are relevant for the biological community and the RJMCMC method proved effective in treating these sort of challenges [32].

In the area of signal processing, Andrieu et al analysed the RJMCMC's capabilities in doing model selection [39] and estimation of sinusoids corrupted by noise [38].

As expected, the RJMCMC algorithm has been used in computer science problems too. In [40], the authors use the RJMCMC method on gathered data of bicycles positioned in a rack over several days. This is conducted as a problem of image processing for recognizing a series of events, such as the bicycles being put on the rack or removed from it. The authors of [40] define a series of events in order to show how they are connected. Simulated annealing is employed in order to give the Maximum a Posteriori over a series of explanations.

In experimental work one always encounters noisy data. With this in mind, the authors of [41] developed an approach to getting sinusoids out of noisy data by employing the RJMCMC algorithm, with the natural frequencies as parameters to be estimated. Their aim was to separate frequencies that were too close to each other, with concluding remarks that the algorithm demonstrated convergence issues.

Remarkable work is presented by Yeh et al. in [42]. The authors of [42] introduce a new algorithm called Locally Annealed RJMCMC(LARJ-MCMC) in the area of graphical layout models. This is done with the scope of exploring the probable space more efficiently. The new algorithm is applied on two examples, the placement of chairs and tables in a coffee shop and golf courses arrangements on basis of difficulty levels and boundaries. The drawback of the work presented in [42] is that the authors considered only forward modelling and not inverse modelling as well.

A particularly relevant literature contribution is found in [43]. In the article the author conducts a critical overview of the progress done with the RJMCMC algorithm, 10 years after it was firstly introduced. The conversation is kept mainly on two issues, that of efficient chains and automation of the method. The author also discusses a number of available softwares to implement the RJMCMC algorithm.

However, the article was published in 2005 and at the present moment the advances in RJMCMC surpass 21 years since its development. Even so, there exist no automation yet of the method, while the efficiency in chain progression was slightly approached by combining the RJMCMC algorithm with other MCMC methods, such as it was done in [42].

In Chapter 6 of [44], the developer of the RJMCMC algorithm addresses some of the issues that arise when using the RJMCMC method. The first important aspect to remember from [44] is that the automation of the RJMCMC algorithm is so far unattainable due to the computational costs required and the amount of programming necessary. As in previous contributions from the same author, some examples of using the RJMCMC method are given, on a small regression problem and change point analysis cases. The author makes some reference to other contributions that treated in fairness the proposal choice issue. Even though this particular contribution does not bring any new or innovative improvements to the RJMCMC algorithm, nevertheless is a valuable overview of the method's capabilities and drawbacks.

The issue of Markov chains convergence is discussed by Brooks et al. in [45]. The authors of [45] firstly explain the RJMCMC algorithm and give an example of its usage on a graphical gaussian problem. Even more relevant is the fact that they consider a previous method that attempted to explain the convergence of Markov chains, method developed by Gelman and Rubin dated back to 1992 [46]. The novelty of their approach is that they study the Markov chains convergence by looking at the variance within chains and within models, while Gelman and Rubin only looked at it from the perspective of variance between chains. This is done by parallel simulations, basically starting off chains at different positions. As Peter J. Green suspected, the problem of convergence is much more complicated to delve into; fact that appears in the conclusions of [45].

A fair conclusion from all the previous literature review is that there is still much more space for research in structural dynamics when it comes to MCMC sampling algorithms and the RJMCMC method in particular.

There is interest sparking in other MCMC sampling algorithms that could be promising for doing both parameter estimation and model selection. One such algorithm is the Approximate Bayesian Computation(ABC) method. The ABC algorithm is thoroughly explained in [47]. The authors of the paper present a tutorial on the ABC

method that fully comprises of the explanation of the algorithm, its capabilities and drawbacks by using it on simulated examples in all its variations, ABC - particle filtering, ABC - partial rejection control, ABC - population Monte Carlo sampling , ABC - sequential Monte Carlo sampling and ABC - MH sampling. One thing all these methods have in common is the fact that through the ABC algorithm, one does not need to evaluate the likelihood function. The likelihood function is seen in broad terms as the probability of the data. And this is where the important drawback of the method comes into place - the ABC algorithm works best on simulated scenarios, those where real data is not available. Now, when doing SID one normally finds that the data is available through testing and that the identification process makes use of that data in order to approximate a model and its parameters. For this reason, the examples provided in [47] are applications on a binomial function and an exponential one.
However, in [48], one can find yet another variation of the ABC method, ABC by Subset Simulation(ABC-SubSim). The Subset Simulation approach was first developed in [49] and it works with nested regions of the probability of iterest. In the [49] contribution, the authors used the SubSim method together with a modified Metropolis algorithm(MMA) in order to study failure probabilities that are too small to be studied by normal probability sampling algorithms. This is done by the use of conditional probability distributions. The MMA sampling algorithm gives higher acceptance rates for the samples, which is a positive point but creates time dependance between samples, which is a clear drawback. The idea behind MMA is that for a vector of a high number of parameters to be sampled(i.e. more than 10^3 parameters), the sampling procedure happens in turns for each parameter of the vector rather than for the vector of parameters as a block. Going back to the contributions of [48], the SubSim method is put together with the ABC algorithm to do parameter estimation and discuss doing model selection, as a by-product of the parameter estimation. The newly introduced method, ABC-SubSim is a combination of ABC with MMA doing sampling on small probabilities distributions. The algorithm is applied on simulated scenarios of a moving average model and a linear oscillator. The concluding remarks are that ABC-SubSim brings improvements to other variations of the ABC algorithm but the model selection issue is still unaddressed. Also, the algorithm is once again, applied to simulated cases as once noisy data is considered (i.e. experimental data), the algorithm's efficiency drops.

The usage of the MCMC sampling algorithms and respectively, the RJMCMC algorithm spread into other areas of structural dynamics as well. Of particular im-

portance, due to the work conducted by the author of the present thesis, is the applications of the RJMCMC algorithm encountered in structural health monitoring(SHM). The next section addresses the literature review of past work conducted in applying MCMC methods in SHM.

2.3 SHM Literature Review

Damage detection is an important aspect in today's industrial world as efficient and reliable ways of using mechanical, civil, aerospace structures are sought. By detecting damage in a structure one has as an end purpose increasing the life span of otherwise expensive to replace parts. There are several contexts in which damage detection is conducted:

- SHM
- CM(Condition monitoring)
- HUMS(Health and usage monitoring system)
- DP(Damage Prognosis)
- SPC(Statistical Process Control)
- NDT(Nondestructive testing)

[66]

The focus of this thesis for Structural Health Monitoring is on combining Bayesian inference with SHM in order to conduct damage detection by using the RJMCMC algorithm. This is considered in order to show the vast applications of the RJMCMC algorithm in structural dynamics. The aim is to provide a method that can identify and also quantify damage in a structure.

Damage detection is a widely considered area in research and the methods to deal with identification and characterisation of damages or defects are numerous. Probably the simplest case of damage and the most popular in damage detection is crack in a beam. This type of damage is popular in industrial applications. Methods of assessing damage vary. Nondestructive evaluation techniques are mostly desired. Users tend to model cracks in beams through massless rotational springs [50]. Modal

analysis(i.e. natural frequencies and mode shapes) is used as well to determine cracks position, location and size [51], [52]. Finite element analysis can be quite useful in modelling cracks in structures and the extent of damage to very fine tuning. This is mostly desirable if the user wants to construct a continuous model. Functions can be used in order to define the crack [53]. Damages due to fatigue are prevalent in engineering structures. One way of analysis of maximum damage a structure can take is by using surrogate models, such as Kriging surrogate models [54]. In [54] the authors identify the cracks location and size using stochastic particle swarm optimization and use them as input for the surrogate models that have as output modal frequencies. Evolutionary algorithms, such as Bees algorithm have been used in the past to do multiple crack detection [55]. Experimental investigations are important as they give one a view of what is happening to damaged structures under a range of vibration regimes [56]. A highly important effect of some cracks in structures is induced nonlinerity and studies exist on these effects with applications in SHM such as [57].

2.3.1 MCMC Sampling Methods

There are applications of the Bayesian framework in SHM, in contributions such as [68]. The authors of [68] conducted a study in order to identify damage in a component by using Bayesian inference. The damage was detected by using the probabilities of model parameters, which were obtained through several trails of data gathering and probability theory. An alarm function was defined such as if the probabilities of certain parameters(such as stiffness) were under the alarm function value, that indicated damage in the structure. The drawback to this approach is that it does not quantify the damage(i.e. location, type etc.)and undamaged case data is required. In [69], the focus was on damage growth in an aircraft structure, using the Paris law model. The authors provided a comparison between Bayesian inference as a tool for identifying damage growth parameters and nonlinear regression method with the same purpose. They concluded that even though Bayesian inference is attractive due to its capabilities of quantifying uncertainties, once the number of parameters increases, the nonlinear regression method becomes desirable due to lower computational cost. The method proved successful in identifying parameters of damage growth models.

The authors of [70] focused on doing damage detection for a thin plate with a single

crack. The plate was subjected to free vibration and the experimental gathered data was used to identify the single crack present in the plate. The model of the plate was produced using Finite Element(FE) analysis. Bayesian inference together with the population-based MCMC method were applied in order to produce estimates of the crack location, position and orientation. The method proved successful and it emphasized the advantage of not needing undamaged data from the plate. The drawback to the procedure introduced in [70] is that one needs to be confident that the FE model was defined correctly(i.e. geometry, boundary conditions, material properties,etc.).

Multiple degrees of freedom analysis exist in areas such as civil engineering where progressive damage is of more interest. In [58], by using a Bayesian framework together with the adaptive MH algorithm, the authors managed to quantify the uncertainty inherent in the progress of damage in a 7-story full-scale building. An FE model of the building is also introduced. Hierarchical Bayesian Model Updating and RJMCMC have also been used as of recently in civil engineering applications [59],[60].

2.3.2 RJMCMC

The RJMCMC has been used for damage detection scenarios previously. In [71] two cases of component degradation were considered. The first scenario included degradation due to reparations or aging of the component. The parameters considered in the case study were failure rate and the time at which the changes in failure rate happened. The second scenario included degradation due to fatigue and it followed the Paris-Erdogan law for crack growth. The parameters taken into consideration for the second case study were mean and standard deviation of multiplicative noise model and time at which changes in the mean and standard deviation happened. In both cases the RJMCMC method proved successful in identifying the studied parameters, considering the varying number of change points. Though relevant, the work done in [71] fails to characterise the damage inherent in the component, either due to machining, fatigue or life span reduction.

In [73] the focus is only on detecting damage due to fatigue of components. The study was carried on a rectangular component with a middle crack. There were two models for crack growth considered, the Paris model and the small time scale

model. The RJMCMC algorithm was applied to do model selection between the two models of fatigue crack growth and to identify the parameters characterising each crack growth model, rather than parameters characterising the crack or damage itself. The authors of [73] dealt with model updating and averaging as well, which proved to be straightforward. The main advantage of the method applied was that only one simulation with the RJMCMC was necessary in order to get the parameters of interest. No characterisation of the damage was provided as it was considered part of prior knowledge. One of the most recent publications on the usage of the RJMCMC method in damage detection is [74]. The authors of [74] used headphones as tools for identifying possible dents in submerged structures with the end purpose of efficiently predicting the loads from buckling. The RJMCMC method was used due to the fact that the dents number is unknown. In this study the focus was on characterizing the dents through the identified parameters as well: dent amplitude, dent vertical and horizontal location.

2.4 Summary

The current chapter was aimed at presenting the past work done in structural dynamics by employing Bayesian system identification. The work done previously tends to concentrate on the problem of parameter estimation, with some MCMC algorithms being more effective or computationally faster than others. For the issue of model selection, there is still a need for algorithms that can move between spaces of varying dimensions. In the considered previous work, the model selection was approached by using criterions such as Bayesian information criterion or Akaike information criterion. The lack of a computational tool that is able to do both parameter estimation and model selection has driven the need to introduce the RJMCMC algorithm in the area of SID of dynamical structures. The following chapter will deal with the theory of Bayesian inference and MCMC sampling algorithms in order to set up the scene for the RJMCMC algorithm.

Chapter 3

Bayesian Inference

Chapter 3 covers the theory behind Bayesian inference, why is it being used, its advantages in dealing with uncertainty as well as its important connection to Markov Chain Monte Carlo sampling algorithms. This leads to the other purpose of the present chapter, introducing Markov Chain Monte Carlo algorithms. Some relevant or most used algorithms are explained briefly, while the main attention is kept on the Metropolis-Hastings method, which is at the core of the Reversible Jump Markov Chain Monte Carlo algorithm. Each MCMC sampling algorithm has its advantages and disadvantages, which will be discussed below. The idea of using MCMC sampling algorithms in doing system identification for dynamical systems is not a new one. Various MCMC methods have been used in structural dynamics, within a Bayesian framework, to do parameter estimation, just a few examples being [24], [25], [48]. It is important to have a basic knowledge of what the other MCMC methods can do so that their limitations could be exposed and the need for the RJMCMC algorithm in structural dynamics emphasised.

3.1 Introduction

The need for using Bayesian inference and probabilistic frameworks in general, came from the existence of uncertainty. Uncertainty could arise from either unknown, ignored and/or compressed information. This information can be either from the studied system itself, conducted experiments or processing errors. Examples of situ-

ations where one can encounter uncertainty are during the process of mathematically modelling the system, due to the differences between the approximated model and reality, during the estimation process due to the fact that one or more parameters are unknown and there could be a range of possible values. A very common source of uncertainty is encountered during data recording due to noise present in the equipment, cables, sensors, etc. Whatever the provenance, uncertainty can affect the results of system identification, sometimes invalidating them [3]. For this reason, rather than attempting to know exactly what parameters describe a system or precisely know how the system is mathematically modelled, one is trying to provide a probabilistic estimation of the parameters of interest and an approximation of the real system through a robust (and as simplified as possible) mathematical model. This is translated into a probabilistic approach,where parameter estimates are expressed as probability distributions instead of crisp 'best estimates'[77]. There are two main views when one addresses probability theory [12]:

- Frequentist view;
- Bayesian view.

As the name implies, the frequentist approach is based on the frequency of outcomes. It does not take into account anything that happened before the event of study. Frequentists tend to concentrate on the variability of the data. Their main concern is the probability of the data given the considered event happening. In this scenario the data is not fixed but rather the parameters are considered constant. Another probabilistic approach is Bayesian Inference, which is the approach used in the present work. Unlike the frequentist view, the Bayesian approach is concerned with the probability of the studied event happening given the set of data that is considered fixed. Bayesian Inference is based on degrees of belief, which means that when constructing the probability of an event happening, one takes into account any prior information about that event. The dispute between the two probabilistic approaches is still without a conclusion. As the present thesis is focused on using a Bayesian approach, the rest of this chapter will concentrate on the advantages of using a Bayesian framework.

3.2 Main Bayesian Concepts

The first document related to Bayesian inference are dated back to the year 1763 when the publication 'An Essay Towards Solving a Problem in the Doctrine of Chances'[61] surfaced, after the death of its author, the Reverend Thomas Bayes. The term 'Bayesian' came into its correct usage sometime in the 1900s when statisticians started to employ it within the context of probability approaches that make use of degrees of belief. In order to understand how Bayes theory works, let one assume that we have two events, event A and event B. It is considered that the reader is familiar with the basics of probability theory. For more in detail information the reader is referred to [63]. In the below equations $P(A|B)$ and $P(B|A)$ are referred to as marginal probabilities while $P(A, B)$ and $P(B, A)$ are referred to as conditional probabilities. As it is known that:

$$P(A|B) = \frac{P(A, B)}{P(B)} \tag{3.1}$$

and that

$$P(B|A) = \frac{P(B, A)}{P(A)} \tag{3.2}$$

given that

$$P(A, B) = P(B, A) \tag{3.3}$$

one can easily conclude that

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} \tag{3.4}$$

Equation 3.4 is know as Bayes' theorem, even though the current form of this equation was introduced by Laplace (in discrete form) and not by Bayes himself [62]. In simpler terms, Bayes' theorem tells one that the posterior probability distribution of event B happening (given event A) is equal to the likelihood of event A (given event B) multiplied by the probability of event B (as it is prior known), divided by

the probability of event A. Bayes' theorem is valid for more than two events as well. Let one look at a slightly different scenario now. One has now three events, event A, event B and event C and the interest is in finding the probability $P(B, A|C)$,i.e. the joint probability of events B and A given that event C happened. In the same way as before,

$$P(A, B|C) = \frac{P(A, B, C)}{P(C)} \tag{3.5}$$

and that

$$P(C|A, B) = \frac{P(C, A, B)}{P(A, B)} \tag{3.6}$$

as

$$P(A, B, C) = P(C, A, B) \tag{3.7}$$

one can conclude that

$$P(A, B|C) = \frac{P(C|A, B)P(A, B)}{P(C)} \tag{3.8}$$

Using equation 3.1 one can replace $P(A, B)$ in equation 3.8 such that it now becomes

$$P(A, B|C) = \frac{P(C|A, B)P(A|B)P(B)}{P(C)} \tag{3.9}$$

One of the most important aspects when using a Bayesian framework rather than other probabilistic approaches is the use of a prior distribution [67]. Priors, defined in equations (3.4) and (3.9) as $P(B)$, represent the knowledge one has about the event one is studying[78]. This knowledge could be expressed as a Uniform distribution in some cases (imposing limits on the values a certain event can take) or maybe as other type of probability distributions (a commonly encountered used probability for priors is the Normal distribution which will be discussed in more detail further on). It is considered that the more a posterior probability distribution (expressed in equations (3.4) and (3.9) as $P(B|A)$ and $P(A, B|C)$) of interest is dependent on the prior, the

less robust it becomes. Priors though, work hand in hand with the likelihood (defined in equations (3.4) and (3.9) as $P(A|B)$ and $P(C|A,B)P(A|B)$). What controls the likelihood is the amount of data (i.e. events A and C in equations (3.4) and (3.9)) one has and how 'informative' that data is. In simpler words, the more 'informative' data one has, the more weight the likelihood will have in Bayesian inference and so it follows that the less control the prior has on the end result [67]. Another aspect of Bayesian inference is the so called evidence term (represented in equations (3.4) and (3.9) as $P(A)$ and $P(C)$). Due to the fact that one is following a probabilistic framework, one needs to ensure that the posterior distributions integrate to 1 [63]. In the case of discrete events, that would mean a summation of the numerator of equations (3.4) and (3.9), while in the case of continuous events it would translate to the integral of the numerator. The integrals are often enough found to be intractable which led to the usage of estimating algorithms that are further explained in this chapter.

Now that Bayes' theorem was defined, let us concentrate on the advantages of using a Bayesian framework:

- Bayesian inference makes use of prior knowledge in order to help speed up the process of getting the posterior probability of interest;
- Within a Bayesian framework the data is considered fixed while the parameters are considered not to be constant. This provides estimations over the parameters through the form of probability distributions rather than exact values;
- Though sometimes governed by intractable integrals and posterior distributions with complex geometries, the Bayesian framework can use Markov Chain Monte Carlo sampling algorithms to deal with the issue;
- Bayesian inference is also capable to provide confidence intervals and deal with uncertainty;
- In engineering one wants efficient but simple solutions - Bayesian inference prevents overfitting.
- Bayesian inference follows the Occam's principle - it favors mathematical simplicity.

For the past decade or so,Bayesian inference started coming into its own rights in

the field of research, proving itself to be more and more the chosen approach in scenarios where uncertainty accountability is crucial. The author of this thesis aims to demonstrate that a Bayesian framework is a reliable and efficient choice when dealing with SID of dynamical structures.

One of the reasons why Bayesian inference became so popular is due to the fact that MCMC sampling methods came into light and the mathematical difficulties that occurred until that point(such as intractable integrals), finally found a solution. This is not to say that other estimation methods did not exist until the discovery of MCMC sampling algorithms. Some examples of estimation methods used in Bayesian inference are least-squares and Newton-Raphson; these two methods in particular, concentrate on finding a maximum estimate locally,such that they are more concerned with the likelihood [67].

For the purpose of the present work, the author found MCMC sampling methods effective and as such, the next part of this chapter will cover background on MCMC sampling methods.

3.3 Bayesian Inference and Markov Chain Monte Carlo

As mentioned in the previous section of the current chapter, in order to get the posterior probability distribution of interest using Bayes' theorem, one can make use of a number of estimating methods. Once the likelihood, prior distribution and evidence term are estimated, one can get the posterior distribution of interest.

Let now one concentrate on the evidence term in Bayes' theorem, that being the denominator in equations (3.4) and (3.9). The evidence term, also known as the normalising constant is relevant when one is dealing with more than one event. It is being used to provide evidence on the events of study so that one can easily choose the best event to represent the known data-this is known as the Bayes factor which is the ratio of evidences of two different events. The normalising constant can sometimes be hard to calculate, especially when it comes to continuous events - in that case, the evidence term is often enough an intractable integral.

It is found in literature that one of the used methods in estimating the evidence term in Bayes' theorem is Laplace's method [12], [72].

The Laplace method provides an approximation for the evidence term. In order to do that, it uses Taylor-expansion and mathematical concepts, to get to the final form [12]:

$$Z_p = P^*(x_0)\sqrt{\frac{(2\pi)^k}{det(T)}} \tag{3.10}$$

where Z_p is the evidence term, P^* is the unnormalised probability density of interest, x_0 is the maximum value of the probability distribution, k is the dimension of space x of all values under the probability distribution P^* and T is the matrix of 2^{nd} order derivative of $-ln(P^*)$ at the maximum value x_0 (for the full derivation of the Laplace method, see Appendix). The issues that arise with the Laplace method are mostly concerned with the fact that through all of the approximations, the Z_p term becomes basis-dependent(Z_p becomes dependent on any transformation of x) when dealing with real events it should be basis independent. Of course this means that one needs to be careful when choosing the basis. Another relevant problem when using Laplace method is that it approximates the probability of interest to be a Gaussian shaped probability distribution which is often enough not true.

Even though good in the process of estimating, due to its drawbacks which can be found in [72], the Laplace method can be replaced by MCMC sampling algorithms. The MCMC methods can express the evidence as a basis-independent term. First of all, let one define what Markov Chain Monte Carlo sampling algorithms are; MCMC sampling algorithms are computational tools that use random variables and Markov chains in order to provide samples from a probability distribution of interest. In order to create a Markov chain one needs a starting point (or more precisely a probability distribution at the starting point) and a function that helps transition from the current point's probability distribution to the next one. This chain starts randomly moving within the considered space and it depends only on the previous point. Another way of calling this type of chain is 'memoryless'.

There are two problems that MCMC methods aim to solve [12]:

- Provide samples from probability distributions of interest

- Estimate expectations of functions under the studied probability distribution

Each of the MCMC sampling algorithm can either deal with one of the aspects of system identification or both. Examples of known MCMC sampling algorithms are : Importance Sampling, Gibbs Sampling, Rejection Sampling, Slice Sampling, Metropolis-Hastings algorithm and many other variations of these. Let one learn more about each of these sampling algorithms, its advantages and disadvantages to better understand the author's choice of using the Metropolis-Hastings algorithm in the present work. The Beta distribution, represented in **Figure 3.1**, will be used as a demonstrator for explaining how the MCMC sampling algorithms work.

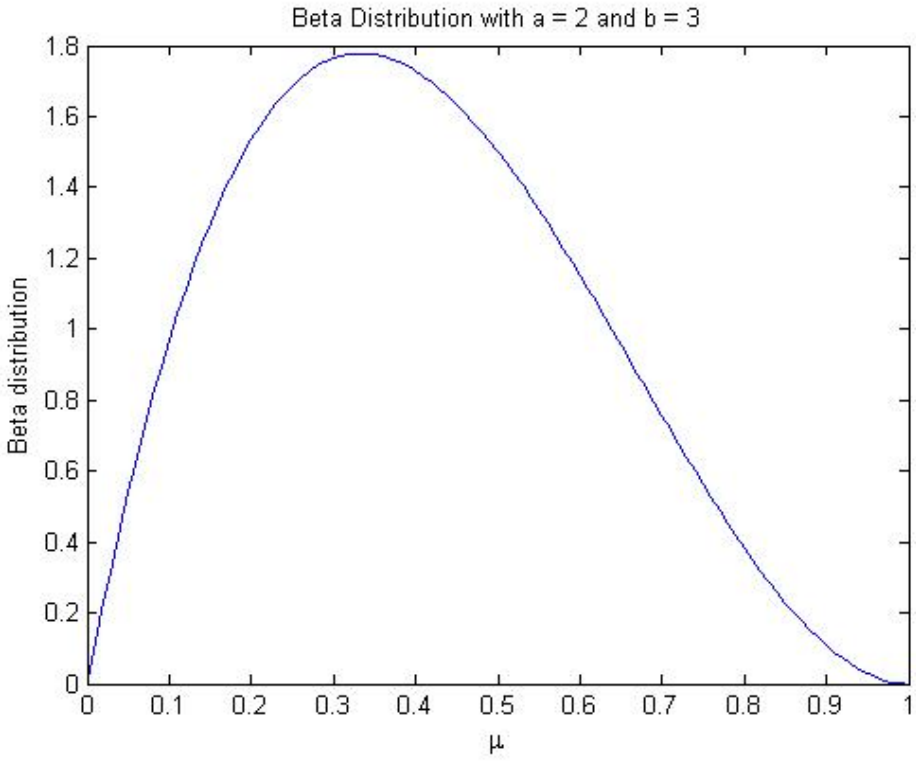

Figure 3.1: Beta Distribution with a = 2 and b = 3

The beta distribution is a known probability distribution defined by shape parameters a and b [77]. Let one concentrate on some MCMC sampling methods.

3.3.1 Importance Sampling

Importance Sampling is an MCMC algorithm that tries to asses the importance of each sample. The algorithm assigns different weights to each sample in the Markov chain and by comparing those weights it decides where to go next. Importance Sampling deals with expectations of functions under a certain probability distribution - for the purpose of exemplifying the method, let one say the probability distribution is the beta distribution defined earlier. This means that it covers only the second issue that MCMC sampling algorithms, in general, are trying to assess [12]. This can already be seen as a disadvantage.

The name of the method comes from the fact that each point in the estimation process is checked first for importance by using a weighting system.

Let one assume the following scenario:

There is a Beta Distribution(see **Figure 3.1**) defined by the following equation:

$$\beta(\mu \mid a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\mu^{(a-1)}(1-\mu)^{(b-1)} \tag{3.11}$$

where $a = 2$ and $b = 3$ are Beta distribution shape parameters arbitrarily chosen and $\Gamma(a+b)$, $\Gamma(a)$ and $\Gamma(b)$ are gamma functions of the shape parameters and are defined as:

$$\Gamma(a+b) = (a+b-1)! \tag{3.12}$$

$$\Gamma(a) = (a-1)! \tag{3.13}$$

$$\Gamma(b) = (b-1)! \tag{3.14}$$

for a and b integers.

The variable of interest is μ.

Let one assume that the Beta distribution is the target distribution of interest to

a proportionality factor due to the fact that the target probability distribution of interest has a complex geometry. Now one defines a polynomial function(represented in **Figure 3.2** in black), for which one is trying to get the expectation of, under the Beta distribution defined previously.

$$\mu^3 + \mu^2 + 4\mu + 0.2 = y \tag{3.15}$$

By using the Importance Sampling algorithm approach, one will need a new extra distribution, let one say a Gaussian distribution(represented in **Figure 3.2** in red) of mean $m = 0.4$ and standard deviation $\sigma = 0.2$ (as defined in the equation below), such that the Beta distribution(represented in **Figure 3.2** in blue) of interest is enveloped by the new defined distribution.

$$N(\mu|m, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(\mu-m)^2}{2\sigma^2}} \tag{3.16}$$

The weight for each sample is calculated by doing the ratio of the value of Beta density for that sample and the value of Gaussian density for that sample. With the help of this ratio one calculates the expectation of the function under the Beta distribution for that sample. The process keeps repeating for whatever number of iterations is decided on by the user.

As mentioned before, the Importance sampling algorithm only covers the expectation of functions and not the sampling. One issue with this sampling method is that the new proposed distribution(in the example above a Gaussian distribution) needs to completely envelope the target distribution(i.e. the Beta distribution in the example above). As one can observe in **Figure 3.2**, this is one example when the Gaussian distribution was defined incorrectly in the sense that it does not envelope the Beta distribution.

Another issue that comes across is the fact that this sampling method will not work in many dimensions as it can be expensive time wise and the weight factors can vary too much,making it unreliable.

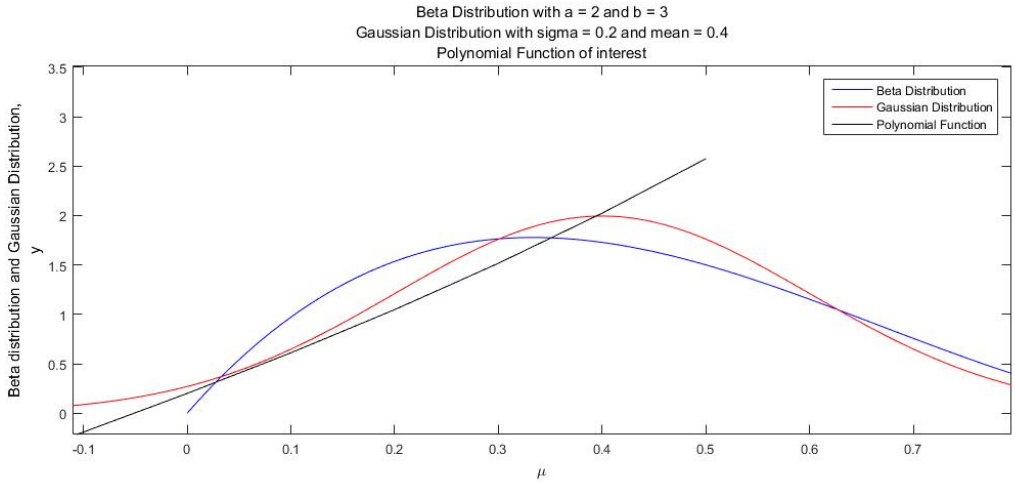

Figure 3.2: Beta Distribution, Gaussian Distribution and Polynomial Function

3.3.2 Gibbs Sampling

Gibbs sampling is another MCMC method commonly found in literature. Gibbs algorithm is concerned with providing samples from joint distributions. In order for the Gibbs sampling methods to be applied, one needs to have a minimum of two probability distributions of interest and to construct the proposal density as the conditional probability distribution of the joint distributions. Again, it is useful to know the probability distributions of interest to a certain extent of proportionality, due to the fact that it prevents the Markov chain getting 'stuck' in low probability regions.

Let one illustrate the Gibbs sampling procedure by employing the use of a multivariate Beta distribution. In the context of two or higher dimensional space, the Beta distribution can be approximated as a Dirichlet distribution with the probability density function:

$$f(\mu \mid \alpha) = \frac{\prod_{i=1}^{K} \mu_i^{(\alpha_i - 1)}}{B(\alpha)} \tag{3.17}$$

where μ is the vector of variables of interest, α is the distribution's control vector parameter, K is the number of variables and $B(\alpha)$ is a function defined by:

$$B(\alpha) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{K} \alpha_i)} \tag{3.18}$$

with the Γ function defined as previously.

The Dirichlet distribution can be a little mathematically heavy but the only purpose of it in this chapter is to be used as a demonstrator so no intense knowledge about it is required.

So, as mentioned before, the Gibbs sampling algorithm can only be employed when the vector of variables μ contains at least two variables, in other words, when $K \geqq 2$.

Let one assume that in the present scenario $K = 2$ so that $\mu = \{\mu^{(1)}, \mu^{(2)}\}$ is a vector of two variables. In this case, the proposals for the new variables will be taken as follows, $\mu^{(1)*}$ from the conditional probability $P(\mu^{(1)} \mid \mu^{(2)})$ and $\mu^{(2)*}$ from the conditional probability $P(\mu^{(2)} \mid \mu^{(1)})$.

One of the disadvantages of the Gibbs method is the possibility of a slow convergence of the Markov chain to the posterior distribution of interest due to the fact that the moves proposed are based on a random walk. This on the other hand can be addressed by using methods of 'controlling' random walks, methods that can be found in [12] but are not further discussed in this thesis. A drawback could be considered the fact that it only works for high dimensional spaces but at the same time there are other methods (such as Metropolis-Hastings) that can be employed in singular dimensional space effectively.

3.3.3 Rejection Sampling

Moving forward through the MCMC sampling methods, one has the Rejection sampling algorithm as yet another possibility for Bayesian Inference.

The Rejection sampling method is quite straightforward and mathematically approachable in the same sense as the Importance sampling method discussed earlier. As with the Importance sampling, the Rejection sampling algorithm works with a proposal that needs to be known to be quite close to the targeted distribution of interest. Unfortunately, this is hard to achieve when one has no idea about the posterior distribution.

For a better understanding, let one assume that one is dealing with the same Beta distribution described in the Importance sampling section so that the probability distribution of interest is as in **Figure 3.1**, to a proportionality extent. In the context of Rejection sampling, the user has to find a proposal distribution such that by multiplying the density function of that distribution by a known factor, let one call it j, the obtained density function completely envelopes the target probability distribution function, i.e. the Beta distribution - the process can be visualised in **Figure 3.3**.

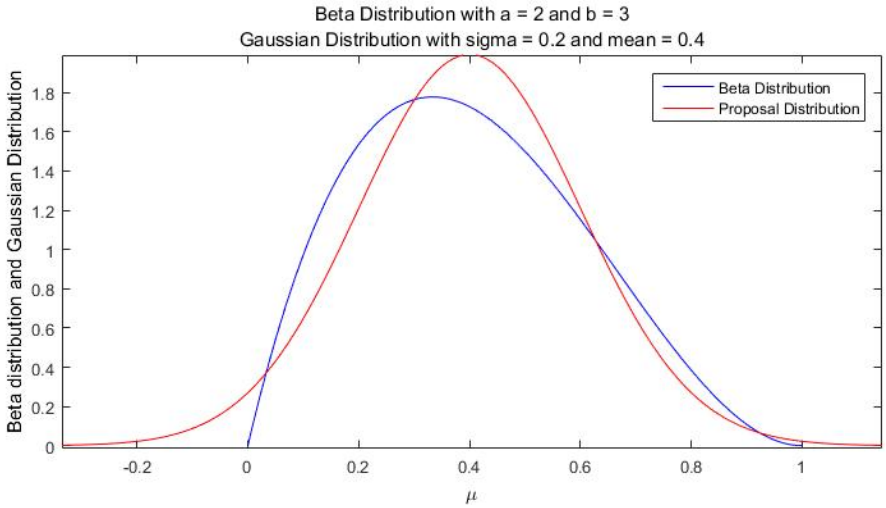

Figure 3.3: Beta Distribution and Proposal Distribution in red

Now, this is clearly a drawback as it is difficult to define an appropriate multiplication factor j. Also, when one goes into higher dimensions, the algorithm falls short in convergence as the rejection rate of the samples would be very high. This makes the algorithm inefficient. For the stated reasons, the Rejection sampling algorithm will be consider no more in this thesis.

3.3.4 Metropolis-Hastings Algorithm

The author would like at this point to emphasis the importance of this following sampling algorithm, the Metropolis-Hastings(MH) sampling method, for the work presented in this thesis.

Similar to the Gibbs sampling, the MH method relies on creating a Markov chain of samples through a random walk, with the end purpose of providing the user with

samples from the posterior probability distribution of interest.

One of the advantages of using the MH algorithm in comparison with other MCMC methods is the fact that the proposal density can be anything the user chooses, in the sense that it does not have to be similar to the target distribution of interest as was described before.

The MH algorithm can be used in both low and high dimensional spaces. The following is the pseudo-code for the MH implementation in the work done by the author:

Algorithm 1 MH Algorithm

1: **for** $n = 1 : N$ **do**
2: *Get* u *from* $U[0,1]$ $-$*generates a random number from an uniform distribution*;
3: **if** $u \leq \alpha_{MH}$ **then**
4: *Update to proposed value of* μ;
5: **else**
6: *Reiterate current value of* μ;
7: **end if**
8: **end for**

The random value u decides where the new proposed state will be, causing a random walk. The acceptance probability, α_{MH} is a statistically defined measure of whether the new proposed state should be accepted into the chain of samples or not. The acceptance probability depends on the target and proposed probability distributions. It is common for the chain of samples or Markov chain to be allowed a burn-in period. This means a period for the chain to settle to the desired mean. To this extent, a number of first samples will be discarded usually from the probability distribution. It comes down to the educated decision of the user what amount of samples will be discarded. Thinning is another procedure chosen by the user for reasons such as to save computational space [75]. This basically means that only every $n^{n^{th}}$ sample will be saved to create the chain of samples to build the posterior distribution. Generally this is done in order to avoid autocorrelation in the Markov chain. Again, it comes down to the user to decide if thinning of the chain is a good approach.

Figure 3.4 illustrates the use of the MH algorithm by sampling from a Gaussian distribution defined by parameters $m = 0$ and $\sigma = 1$ (samples of which are represented in blue), using a Gaussian distribution of mean $m = 0$ and standard deviation $\sigma = 1$ as a proposal(i.e. Gaussian distribution represented in red).

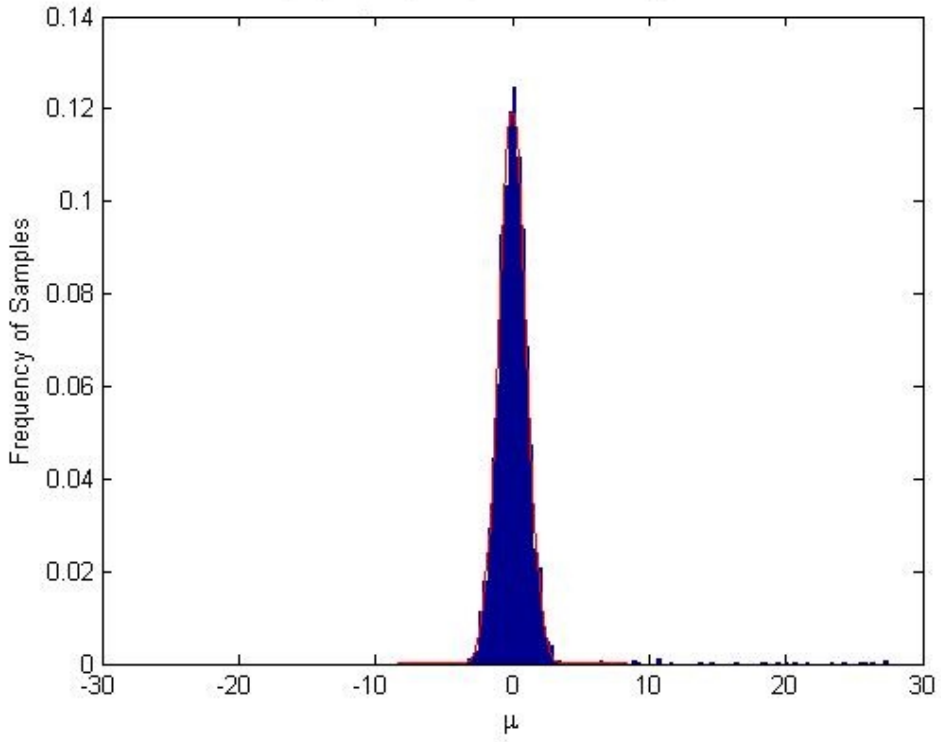

Figure 3.4: MH algorithm applied on a Gaussian Distribution - samples from target distribution and theoretical distribution in red

It is only fair to talk a bit about its disadvantages as well. The convergence of the algorithm to the distribution of interest can be slow due to the fact that, as with the Gibbs sampling, it employs a random walk with samples dependent on each other. This means that the algorithm can be run for some time in order to reach independent samples, which is what one is looking for.

The author's particular interest in the MH sampling algorithm comes from the fact that its usage is popular in problems of system identification in the context of structural dynamics. The popularity of the MH algorithm is due to the fact that the evidence term can be ignored in the sampling process and also due to the fact that the proposal PDF can have any form the user chooses. Within system identification, the MH algorithm is used in doing parameter estimation.

In his paper [34], Green made an observation about the MH algorithm. When confronted with the issue of model selection, the Markov chain loses detailed balance (explained in the following chapter) as it is incapable of moving between spaces of varying dimensions.

This sparked the author's interest in the RJMCMC algorithm, as an alternative to other methods of conducting model selection as the RJMCMC algorithm proved to be capable of doing parameter estimation and model selection simultaneously while obeying the principle of detailed balance as required in order for the Markov chain to be able to provide samples. The RJMCMC algorithm will be explained in full detail in **Chapter 4**.

3.3.5 Slice Sampling

Yet another MCMC sampling algorithm is the Slice sampling method. Rather than an algorithm in its own rights, the Slice sampling method improves the random walk present in the MH algorithm or the Gibbs sampling algorithm by employing a self-adjusting step size during the walk. The step size is considered a parameter to be constantly updated.

The Slice sampling algorithm concentrates more on the Markov chain in itself in order to reduce randomness in steps. In the case of a MH algorithm, an interval is decided upon, between the current sample,μ and the proposed sample,μ'. A sample is drawn from the uniform distribution of the chosen interval. The probability of the new drawn sample is compared against a random variable generated from a uniform distribution between 0 and the proportional value of $P(\mu)$. If the condition is satisfied, then the proposed value is accepted, otherwise, a new interval is chosen.

There are much more comprehensive reviews of the MCMC sampling algorithms in literature such as [76] but for the purpose of this present thesis what was presented so far will suffice.

3.4 Summary

In this chapter the basic theory behind Bayesian Inference was presented. Bayesian Inference is a probabilistic framework used in dealing with uncertainty. One of its main advantages is that it penalises complexity. In the context of doing SID in structural dynamics, one is interested in robust models with appropriately approximated parameters. System identification can be considered successful if the results obtained in processing are in accordance with the data used and also if the mod-

elling can be applied in a generalised scenario. It was also covered in this chapter the importance of MCMC sampling algorithms in doing Bayesian Inference. Some of the most important methods were briefly introduced to the reader, while the main attention was kept on the MH algorithm, due to the fact that it is at the core of the RJMCMC method which is what this thesis is concentrated with.

The following chapter, **Chapter 4**, provides the introduction and in depth description of the RJMCMC algorithm, as employed in system identification of dynamical structures.

Chapter 4

Reversible Jump Markov Chain Monte Carlo

Chapter 4 is concerned with presenting the background and theory for the Reversible Jump Markov Chain Monte Carlo algorithm, as used in the context of SID in structural dynamics. The RJMCMC method in itself was developed in 1995 [34]. At that moment in time, it was seen as a general computational tool to be used in mathematical statistics. The contribution of this present chapter is to update and apply the original ideas of the RJMCMC algorithm for dynamical systems. In order to do that, the algorithm had to be reconsidered mathematically and adapted to the SID framework. Concepts such as detailed balance, dimension matching and mappings, are defined and derived to fit with the specific problems of SID of dynamical systems.

4.1 Transdimensional Markov Chain Monte Carlo methods

The RJMCMC algorithm can be considered as part of a bigger class of transdimensional algorithms. This simply means that these particular algorithms are capable of making transitions between spaces of different dimensions. It cannot be stressed enough that there is a gap, in methods and algorithms in structural dynamics, that are capable of doing both parameter estimation and model selection in a probabilis-

tic manner. It is known that there exist criterions to decide on which model, from a group of possible models, best approximates a structure such as BIC(Bayesian Information Criterion) or AIC(Akaike Information Criterion). These criterions though, do not provide a probability distribution over the group of models. Even more so, the transdimensional methods are designed to move between spaces of varying dimensions in order to do parameter estimation with the added benefit of providing estimates over the model space. These methods aim to tackle issues such as SID in more effective ways.

The TMCMC method [44] does parameter estimation for a model with an extra outcome, that of the evidence term. If one remembers from **Chapter 3**, the evidence term is useful in that it provides a means for doing selection between models, in the case one is dealing with competing models. In simpler terms, if one has two models to choose from for the same system, one could conduct a TMCMC methods analysis on each model, get the evidence term individually, compare the evidence terms and that would tell the user which model to choose for the studied system. One other algorithm as such is the Transdimensional Transformation Markov Chain Monte Carlo(TTMCMC)[64]. The TTMCMC is an extension to the Transdimensional Markov Chain Monte Carlo(TMCMC) algorithm. It is reasonable to conclude that this approach can get computationally expensive, especially if one deals with more that two models. On the other hand, the TTMCMC algorithm is trying to do the model selection process live. That is, as the parameters for each model are assessed, the evidence term for each individual model is updated and compared to the other models.

There are a few algorithms in literature that can at least provide the means to do model selection such as the ABC method [48]. In the same sense as the TMCMC method, the algorithms provide as an extra outcome the evidence term.

4.2 Reversible Jump Markov Chain Monte Carlo

Going back to the RJMCMC method, this particular algorithm has the necessary computational tools to do parameter estimation and model selection 'in one go', in the sense that it considers the model indicator as an added parameter to be identified. It this sense, the probability distribution over the models indicators will indicate which model is more appropriate to choose for the studied system.

Now, there are on-going conversations in the statistical world about the efficiency of such an algorithm. Some of the raised problems are the acceptance rates of dimension-changing samples [64], the computational time required to get independent samples, the construction of proposal distributions [65], etc. The reader of this thesis needs to be aware at this moment in time that the RJMCMC algorithm is introduced, to the best knowledge of the author, for the first time by the author in the context of structural dynamics with the purpose of doing SID. This is the key contribution of the thesis. As such, the work of the author with the RJMCMC algorithm is proven as an efficient and a strong computational tool for nonlinear system identification.

Historically, the RJMCMC algorithm was firstly introduced by Peter J Green in 1995 in the context of statistical mathematics [34]. The fact that the method is quite mathematically heavy made it less popular for usage in other areas of research. Even so, the RJMCMC algorithm penetrated mostly the biological world where it was used for various research [32].

As it is already understood, the main advantage of the RJMCMC algorithm is that it can move between spaces of varying dimension while simultaneously providing estimates of the parameters the spaces depend on.

Let one think what that means for dynamical systems. In the context of structural dynamics, it is possible that there are multiple models, let one denote them by $\boldsymbol{M} = \{M^{(1)}, M^{(2)}, ..., M^{(l)}, ..., M^{(L)}\}$, that may explain the physical behaviour of the experimental system of interest. Even if the models are of the same dimension, they could be differently parameterized. If one takes into consideration that the models might be of different dimensions as well, the issue is amplified. So what one needs is a computational tool for low DOF systems to deal with the selection of models in order to answer the question of which robust model best represents the data of a system. Furthermore, in the context of nonlinear SID in structural dynamics, the question is, if there exists a computational tool to be able to do SID for low DOF systems, would it be useable when nonlinearity is present.

4.3 Theory of Reversible Jump Markov Chain Monte Carlo

With Reversible Jump MCMC one can address parameter estimation and model selection simultaneously. The way the algorithm works is that it can move/'jump' between parameter spaces of different dimension. The vector set of parameters is not fixed to each model; it can vary in both length and values. Reversible Jump MCMC is capable of 'jumping' from one model to the other and select the most likely model while generating parameter samples for that particular type of system.

In the set of models, $\boldsymbol{M}$, the models will be arranged in the order of their complexity for a better understanding; in this work it follows naturally that a more complex model is defined as one constrained by more parameters. It should be noted that the models do not need to be arranged in any particular order for the algorithm to be applied but it is useful for the user to keep indices of each model considered.

As designed for the use in SID, there are three main moves in the RJMCMC method:

- Birth move - which implies that if the birth move condition is satisfied, the algorithm moves to a model with an additional parameter;
- Death move - which implies that if the death move condition is satisfied, the algorithm 'jumps' to a model with one less parameter;
- Update move - which implies that if neither the birth move condition nor death move condition were satisfied, the algorithm remains within the same model and the use of the normal MH algorithm is employed, as presented before, for parameter estimation.

It is relevant to know that the moves can be flexible in the sense that the algorithm is not limited in any form to only three types of moves, as long as the Markov chain is kept to obey detailed balance, the user can update it to however many moves are useful.

The 'birth move' means that if one is in a less complex model, one might want to attempt a move to a more complex model - from a model defined by less parameters

to a model defined by more parameters. The variables used in order to define the probabilities below are defined as: l is an indicator of the current model, the variable p adjusts the proportion of the update move in relation with the birth and death moves, and the probabilities, $P(l)$ and $P(l+1)$, are prior probabilities of the models at iteration l and $l+1$ respectively, with the priors defined as in **Chapter 3**.

The birth move will be randomly attempted with probability b_l, where :

$$b_l = p \times \min\left\{ 1, \frac{P(l+1)}{P(l)} \right\} \tag{4.1}$$

The 'death move' implies that one is attempting to check a less complex model, i.e. wanting to attempt moving from a model defined by more parameters to a model defined by less parameters. The death move will be randomly attempted with probability d_l, where:

$$d_{l+1} = p \times \min\left\{ 1, \frac{P(l)}{P(l+1)} \right\} \tag{4.2}$$

Lastly, the last move, the update is considered when none of the birth or death moves are. In this case, the parameters in the current model will be updated according to the MH algorithm. The update move will be randomly attempted with probability $u_{l,l+1}$. There is no need to actually define the probability for the update move due to the fact that one only has three moves and that means that their probabilities of happening need to add up to 1. This means that the probability of the update move can be more easily defined as:

$$b_l + d_{l+1} + u_{l,l+1} = 1 \tag{4.3}$$

Now let one have a closer look at equations 4.1 and 4.2. A little bit of rearrangement gives:

$$\frac{b_l P(l)}{P(l+1)} = p \tag{4.4}$$

and

$$\frac{d_{l+1}P(l+1)}{P(l)} = p \tag{4.5}$$

Putting the equations 4.4 and 4.5 together, one gets:

$$\frac{b_l P^2(l)}{d_{l+1}P^2(l+1)} = 1 \tag{4.6}$$

For a better understanding, let one define at this point in time a simplistic case scenario.

It is assumed that one has a set of two candidate models for a dynamical SDOF system,$\boldsymbol{M} = \{ M^{(1)}, M^{(2)}\}$,where model $M^{(1)}$ has one unknown, $\theta^{(1)}$, while model $M^{(2)}$ has two unknown parameters, $\{\theta_1^{(2)}, \theta_2^{(2)}\}$.

It is further assumed that the first model is described by the well-known SDOF linear equation:

$$m\ddot{y} + c\dot{y} + k^{(1)}y = F \tag{4.7}$$

The second model considered is a Duffing oscillator (the nonlinear system can be of any form, but for this case it was chosen to be a Duffing oscillator), described by the following SDOF nonlinear equation:

$$m\ddot{y} + c\dot{y} + k^{(2)}y + k_3^{(2)}y^3 = F \tag{4.8}$$

where $y \in D$ is the assumed gathered output data from the system that was supposedly excited by $F \in D$. **Figure 4.1** is a graphical representation of the two competing models.

In the context of the two equations of motion and the models presented above, the mass m is assumed known as well as the damping coefficient, c, while $k^{(1)} = \theta_1^{(1)}$ and $k^{(2)} = \theta_1^{(2)}$, $k_3^{(2)} = \theta_2^{(2)}$ are the unknown parameters for each of the considered models.

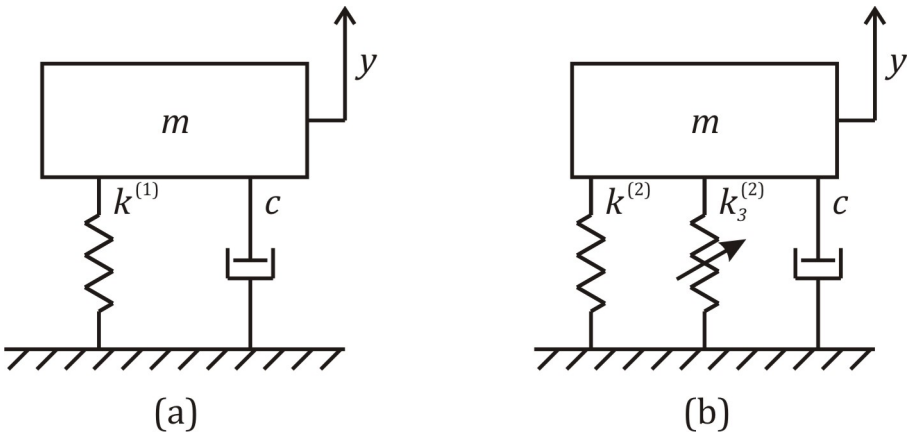

Figure 4.1: (a) Linear system, (b) Nonlinear system

With this described scenario, one wants to do model selection and parameter estimation for the system in question, considering that the only available information about these is the assumed gathered data. The solution proposed is applying the RJMCMC algorithm within a Bayesian framework. In order to do that, one needs to go through the steps of understanding how this method is implemented.

To start off, an explanation of the RJMCMC method is provided through pseudocode, which can be found below.

The quantities α_m, α'_m, are the acceptance probabilities of the birth and death moves respectively. These will be further explained and defined later on.

The generated random number u_m dictates what the proposed move will be. Within each accepted move, the randomly generated numbers u_b and u_d dictate what the proposed samples will be. This is an important aspect as one needs to remember that the MH algorithm is based on a random walk. Another point to remember is that for $l = 1$, there is no death move and for $l = L$, there is no birth move.

Now that one has a general idea of what happens within the RJMCMC algorithm, one needs to delve into the details of it. The main two important ones are respecting the detailed balance of the Markov chain when moving between spaces of varying dimensions and making the dimensions match so that the jump itself could be possible.

With that in mind, remember the system and the two candidate models introduced earlier: at this point in time , there are two models: $M^{(1)}$ which represents a SDOF

Algorithm 2 RJMCMC Algorithm as implemented in SID of dynamical structures

for $n = 1 : N$ **do**
  Get u_m *from* $U[0,1]$ $-$*generates a random number from an uniform distribution*;
  if $u_m \leq b_l^{(n)}$ $-$*birth move condition*; **then**
    $-$*propose birth move*
    Generate u_b *from* $U[0,1]$ $-$*generates a random number from an uniform distribution*;
    Evaluate α_m (*detailed below*)
    if $u_b \leq \alpha_m$ **then**
      $-$*update to model* $l+1$ *and update model parameters*;
    else
      $-$*stay in model* l;
    end if
  else
    if $u_m \leq (b_l^{(n)} + d_{l+1}^{(n)})$ $-$*death move condition*; **then**
      $-$*propose death move*
      Get u_d *from* $U[0,1]$ $-$*generates a random number from an uniform distribution*;
      Evaluate α'_m (*detailed below*)
      if $u_d \leq \alpha'_m$ **then**
        $-$*go to model* $l-1$ *and update model parameters*;
      else
        $-$*stay in model* l;
      end if
    else
      normal MH algorithm, model remains at l *state, update parameters only*;
    end if
  end if
end for

linear system that contains one unknown parameter - the linear stiffness $k^{(1)}$ that is added in vector $\boldsymbol{\theta^{(1)}}$ and $M^{(2)}$ which represents a SDOF nonlinear system that contains two unknown parameters - the linear stiffness $k^{(2)}$ and the cubic stiffness $k_3^{(2)}$, that are added in vector $\boldsymbol{\theta^{(2)}}$.

4.3.1 Detailed Balance

Firstly mentioned was the principle of detailed balance which must be respected in order for MCMC methods to generate samples from the target probability density function(PDF). This is simpler to demonstrate for MH as the dimension space remains the same. With RJMCMC it becomes harder to maintain because the dimension space varies. As such, the detailed balance principle will be first proved for the MH algorithm in order to provide a base for the explanation on detailed balance for the RJMCMC algorithm.

4.3.2 Detailed Balance and the MH algorithm

As mentioned previously, the most important aspect of MCMC methods is that detailed balance is respected. In general terms, detailed balance states that at equilibrium every process should be balanced out by its reverse process. In the context of MH sampling, detailed balance ensures that ergodicity and a limiting distribution are respected [34]. The purpose of this thesis is not to discuss statistical terms, so that for the interested reader details on ergodicity and limiting distributions can be found in [12].

So, let one look at the system defined but with the assumption that one already knows that the gathered data is best approximated with the linear SDOF model. That gives only one model to consider, $M^{(1)}$ and only one issue, that of parameter estimation, $\boldsymbol{\theta^{(1)}}$. This simplifies things such that everything will happen only in one space so there is no change in dimensions. One only needs to be able to move between the current value of the parameter and the proposed value.

The following analysis will make use of the concept of mappings, which are functionals, or simply put functions, that relate states between themselves [33].

For this case, one will define the mapping $h : \boldsymbol{\theta^{(1)}} \rightarrow \boldsymbol{\theta^{(1)'}}$ which is the path from $\boldsymbol{\theta^{(1)}}$ to $\boldsymbol{\theta^{(1)'}}$ with its inverse, h^{-1} being defined as the path from $\boldsymbol{\theta^{(1)'}}$ to $\boldsymbol{\theta^{(1)}}$.

Detailed balance states that, for the standard Metropolis-Hastings algorithm, for a model $M^{(1)}$ that depends on one parameter, i.e. $\boldsymbol{\theta^{(1)}} = \{\ k^{(1)}\}$, the following equation needs to hold.

$$\pi(\boldsymbol{\theta^{(1)}})T(\boldsymbol{\theta^{(1)}} \rightarrow \boldsymbol{\theta^{(1)'}}) = \pi(\boldsymbol{\theta^{(1)'}})T(\boldsymbol{\theta^{(1)'}} \rightarrow \boldsymbol{\theta^{(1)}}) \tag{4.9}$$

Equation (4.9) can be explained as: the target PDF,$\pi(\boldsymbol{\theta^{(1)}})$, of the vector of parameters $\boldsymbol{\theta^{(1)}}$, multiplied by the transition function $T(\boldsymbol{\theta^{(1)}} \rightarrow \boldsymbol{\theta^{(1)'}})$ between the vector of parameters $\boldsymbol{\theta^{(1)}}$ and the new state $\boldsymbol{\theta^{(1)'}}$, has to be equal to the target PDF,$\pi(\boldsymbol{\theta^{(1)'}})$, of the vector of parameters $\boldsymbol{\theta^{(1)'}}$, multiplied by the transition function $T(\boldsymbol{\theta^{(1)'}} \rightarrow \boldsymbol{\theta^{(1)}})$ between the new state of the vector of parameters $\boldsymbol{\theta^{(1)'}}$ and the current state of parameters vector $\boldsymbol{\theta^{(1)}}$.

The transition function mentioned in equation (4.9) can also be written in the following form:

$$T(\boldsymbol{\theta^{(1)}} \rightarrow \boldsymbol{\theta^{(1)'}}) = q(\boldsymbol{\theta^{(1)}}|\boldsymbol{\theta^{(1)'}})\alpha(\boldsymbol{\theta^{(1)}} \rightarrow \boldsymbol{\theta^{(1)'}}) \tag{4.10}$$

where $q(\boldsymbol{\theta^{(1)}}|\boldsymbol{\theta^{(1)'}})$ is the proposal density used in the sampling process. Considering that equation (4.9) is valid for all paths available between the vectors:

$$\int \pi(\boldsymbol{\theta^{(1)}})T(\boldsymbol{\theta^{(1)}} \rightarrow \boldsymbol{\theta^{(1)'}})\,\mathrm{d}\boldsymbol{\theta^{(1)}} = \int \pi(\boldsymbol{\theta^{(1)'}})T(\boldsymbol{\theta^{(1)'}} \rightarrow \boldsymbol{\theta^{(1)}})\,\mathrm{d}\boldsymbol{\theta^{(1)'}} \tag{4.11}$$

or

$$\int \pi(\boldsymbol{\theta^{(1)}})q(\boldsymbol{\theta^{(1)'}}|\boldsymbol{\theta^{(1)}})\alpha(\boldsymbol{\theta^{(1)}} \rightarrow \boldsymbol{\theta^{(1)'}})\,\mathrm{d}\boldsymbol{\theta^{(1)}} = \int \pi(\boldsymbol{\theta^{(1)'}})q(\boldsymbol{\theta^{(1)}}|\boldsymbol{\theta^{(1)'}})\alpha(\boldsymbol{\theta^{(1)'}} \rightarrow \boldsymbol{\theta^{(1)}})\,\mathrm{d}\boldsymbol{\theta^{(1)'}} \tag{4.12}$$

If the considered model would have depended on more than one parameters, that would simply mean that the integration needs to be expanded over the entire space of parameters, which leads to multiple integration signs being employed as it will be

shown later on.

Going back to equation 4.12, $\boldsymbol{\theta^{(1)'}}$ is simply sampled from a Gaussian distribution centered around the current $\boldsymbol{\theta^{(1)}}$ and with variance decided by the user - that is the proposal distribution. This means that:

$$\mathrm{d}\boldsymbol{\theta^{(1)}} = \mathrm{d}\boldsymbol{\theta^{(1)'}} \tag{4.13}$$

There is no need in this case for a special functional that can map the two vector of parameters. The two vectors, $\boldsymbol{\theta^{(1)}}$ and $\boldsymbol{\theta^{(1)'}}$ are part of the same space.

This basically concludes the proof of detailed balance in the case of the MH algorithm. Of course, things can get a little bit more complicated when one is working within a model that depends on more than one parameter but even so, the calculations required are straightforward as once again, the Markov chain is moving within the same space.

The next step is to prove that detailed balance holds in the case of the RJMCMC algorithm. In order to do that, the case of the two models introduced at the beginning of the chapter is employed.

4.3.3 Detailed Balance and RJMCMC

Detailed balance, when it comes to RJMCMC, gets more complicated to evaluate because the variables are part of a space of different dimension. This implies that one cannot be sure that the space dimension of the right hand side is necessarily equal to the space dimension of the left hand side. In most cases it is actually not.

As model $M^{(1)}$ has only one unknown parameter and model $M^{(2)}$ contains two unknown parameters, one can easily spot a mismatch in dimensions. Due to the fact that the dimensions do not match, jumping from one model to the other in order to do model selection proves impossible when using the MH algorithm. What one wants to do is be able to move between the two models in order to decide which of them best fits the available data. In order for the forward and backward moves to occur, the principle of detailed balance needs to be respected (at equilibrium each process should be balanced out by its reverse process). As the dimensions

of the two vectors of parameters for each of the two models do not match, the process of jumping from $M^{(1)}$ to $M^{(2)}$ cannot be balanced out by its reverse process, i.e. jumping from $M^{(2)}$ to $M^{(1)}$. An approach to solving the problem of matching dimensions is by introducing a new parameter in model $M^{(1)}$. This newly introduced parameter (λ) is a randomly generated number. At this point one needs to state the relation between the two models, $M^{(1)}$ and $M^{(2)}$. This is done by using mappings (functionals that can relate different states [33]). In order to move from $M^{(1)}$ to $M^{(2)}$, the following mapping, $h : \mathbb{R}^2 \to \mathbb{R}^2$, was introduced:

$$\begin{Bmatrix} k^{(2)} \\ k_3^{(2)} \end{Bmatrix} = \begin{bmatrix} 0 \\ \mu \end{bmatrix} + \begin{Bmatrix} k^{(1)} \\ \lambda \end{Bmatrix} \tag{4.14}$$

where μ is a chosen value that needs to be tuned in order for the algorithm to reach a solution efficiently.

The inverse of the mapping is applied for a death move, i.e. for going from $M^{(2)}$ to $M^{(1)}$.

The form of the mapping is not chosen randomly. Green [34] also mentions that in order for the mapping to be used in jumping between spaces it needs to be diffeomorphic. A diffeomorphic mapping means one that is differentiable and unique. It is obvious that the created mapping is differentiable. One is left with proving that the mapping is unique. According to the inverse function theorem, as long as the Jacobian at whichever chosen point is nonzero, the mapping is said to be unique [33]. Furthermore, in this particular case, the determinant of the Jacobian is calculated to be unity under all conditions. This concludes the mapping to be differentiable and unique, i.e. diffeomorphic.

Now, having introduced a new parameter in $M^{(1)}$ and having proven the chosen mapping to be diffeomorphic, one can endeavour to prove that the detailed balance principle holds with the RJMCMC algorithm.

If one considers equation (4.12) in the case of the two models $M^{(1)}$ and $M^{(2)}$, one has model $M^{(1)}$ depending on a vector of parameters $\boldsymbol{\theta^{(1)}} = \{k^{(1)}, \lambda\}$ (as the vector of parameters was augmented with a a new parameter λ) and model $M^{(2)}$ depending on $\boldsymbol{\theta^{(2)}} = \{k^{(2)}, k_3^{(2)}\}$ vector of parameters. Writing equation (4.12) using this information:

$$\iint \pi(\boldsymbol{\theta^{(1)}})q(\boldsymbol{\theta^{(2)}}|\boldsymbol{\theta^{(1)}})\alpha(\boldsymbol{\theta^{(1)}} \rightarrow \boldsymbol{\theta^{(2)}})\,\mathrm{d}k^{(1)}\mathrm{d}\lambda = \iint \pi(\boldsymbol{\theta^{(2)}})q(\boldsymbol{\theta^{(1)}}|\boldsymbol{\theta^{(2)}})\alpha(\boldsymbol{\theta^{(2)}} \rightarrow \boldsymbol{\theta^{(1)}})\,\mathrm{d}k^{(2)}\mathrm{d}k_3^{(2)} \quad (4.15)$$

Using the mapping introduced in equation (4.14) and the Jacobian matrix, the relation between $\mathrm{d}k^{(1)}\mathrm{d}\lambda$ and $\mathrm{d}k^{(2)}\mathrm{d}k_3^{(2)}$ can be expressed as:

$$\mathrm{d}k^{(2)}\mathrm{d}k_3^{(2)} = \left|\frac{\partial\ (k^{(2)}, k_3^{(2)})}{\partial\ (k^{(1)}, \lambda)}\right| \mathrm{d}k^{(1)}\mathrm{d}\lambda \quad (4.16)$$

where

$$\left|\frac{\partial\ (k^{(2)}, k_3^{(2)})}{\partial\ (k^{(1)}, k^{(2)})}\right| = \det \begin{bmatrix} \frac{\partial k^{(2)}}{\partial k^{(1)}} & \frac{\partial k^{(2)}}{\partial \lambda} \\ \frac{\partial k_3^{(2)}}{\partial k^{(1)}} & \frac{\partial k_3^{(2)}}{\partial \lambda} \end{bmatrix} \quad (4.17)$$

Calculating the determinant of the Jacobian, one can see that it is always equal to 1 in this case.

Putting everything together, gives the equation of detailed balance for the RJMCMC algorithm:

$$\pi(\boldsymbol{\theta^{(1)}})q(\boldsymbol{\theta^{(2)}}|\boldsymbol{\theta^{(1)}})\alpha(\boldsymbol{\theta^{(1)}} \rightarrow \boldsymbol{\theta^{(2)}}) = \pi(\boldsymbol{\theta^{(2)}})q(\boldsymbol{\theta^{(1)}}|\boldsymbol{\theta^{(2)}})\alpha(\boldsymbol{\theta^{(2)}} \rightarrow \boldsymbol{\theta^{(1)}})\left|\frac{\partial\ (k^{(2)}, k_3^{(2)})}{\partial\ (k^{(1)}, \lambda)}\right| \quad (4.18)$$

At this point one has the following:

- one vector of parameters for $M^{(1)}$, $\boldsymbol{\theta^{(1)}} = \{k^{(1)}\} \in \mathbb{R}^n$;
- one vector of parameters for $M^{(2)}$, $\boldsymbol{\theta^{(2)}} = \{k^{(2)}, k_3^{(2)}\} \in \mathbb{R}^{n'}$;
- one vector of generated random numbers, from a known density g, to help with the transition from $\boldsymbol{\theta^{(1)}}$ to $\boldsymbol{\theta^{(2)}}$, $\boldsymbol{u} \in \mathbb{R}^r$;
- one vector of generated random numbers, from a known density g', to help with the transition from $\boldsymbol{\theta^{(2)}}$ to $\boldsymbol{\theta^{(1)}}$, $\boldsymbol{u'} \in \mathbb{R}^{r'}$;

Using the mapping described in equation (4.14),one has $h : \mathbb{R}^n \times \mathbb{R}^r \rightarrow \mathbb{R}^{n'} \times \mathbb{R}^{r'}$. In order for the mapping h and its inverse to be diffeomorphic, then one needs to ensure $n+r = n'+r'$ (dimension matching). In this particular case, one knows that $n = 1$ and $n' = 2$. The linear stiffness parameters, $k^{(1)}$ and $k^{(2)}$, of $M^{(1)}$ and $M^{(2)}$ were kept the same. The only random number needed in order to jump from one model to the other was λ, $\boldsymbol{u} = (\lambda)$, and it was used to move from $M^{(1)}$ to $M^{(2)}$. According to the mapping h, no random numbers were necessary to move from $M^{(2)}$ to $M^{(1)}$, so $\boldsymbol{u'} = 0$. This translates into $r = 1$ and $r' = 0$. Under these conditions, the relation $n + r = n' + r'$ is met.

Now one is left with evaluating the acceptance probability of moves, one uses the subscript m to indicate the move attempted such that: for $m = 1$, a birth move is attempted, for $m = 2$ a death move is attempted, for $m = 3$ an update move is attempted.

The acceptance probability for the birth move can be evaluated as follows:

$$\alpha_m(\boldsymbol{\theta^{(1)}} \rightarrow \boldsymbol{\theta^{(2)}}) = \min\left\{ 1, \frac{\pi(\boldsymbol{\theta^{(2)}}) j_m(\boldsymbol{\theta^{(2)}})}{\pi(\boldsymbol{\theta^{(1)}}) j_m(\boldsymbol{\theta^{(1)}}) g(\lambda)} \right\} \tag{4.19}$$

where all the values are known or can be estimated and j_m is the Jacobian that was calculated to be equal to 1. The $g(\lambda)$ term comes from the transition function, $T(\boldsymbol{\theta^{(1)}} \rightarrow \boldsymbol{\theta^{(2)}})$. The transition function can be also written as:

$$T(\boldsymbol{\theta^{(1)}} \rightarrow \boldsymbol{\theta^{(2)}}) = g(\lambda)\alpha(\boldsymbol{\theta^{(1)}} \rightarrow \boldsymbol{\theta^{(2)}}) \tag{4.20}$$

where g is a Gaussian PDF with mean 0. One can notice that the g' PDF is not represented in equation 4.19 for simplicity, as $\boldsymbol{u'} = 0$.

The term $g(\lambda)$ can also be evaluated:

$$g(\lambda) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(\frac{-\lambda^2}{2\sigma^2}) \tag{4.21}$$

where σ is the standard deviation and is arbitrarily chosen and tuned accordingly.

Another way of writing the acceptance probability for a birth move is:

$$\alpha_m(\boldsymbol{\theta^{(1)}} \rightarrow \boldsymbol{\theta^{(2)}}) = \min\{\ 1, r_m\} \tag{4.22}$$

where r_m is the ratio of move and is computed as:

$$r_m = \frac{\pi(\boldsymbol{\theta^{(2)}})}{\pi(\boldsymbol{\theta^{(1)}})g(\lambda)} \tag{4.23}$$

The acceptance probability of the death move can now be evaluated:

$$\alpha'_m(\boldsymbol{\theta^{(2)}} \rightarrow \boldsymbol{\theta^{(1)}}) = \min\left\{\ 1, r_m^{-1}\right\} \tag{4.24}$$

At this point one gets a better grasp of the concepts needed to understand the RJMCMC algorithm and the mathematical tools for implementing it with either real or simulated data.

4.4 Summary

The purpose of this chapter was to introduce and fully explain the RJMCMC algorithm as used in the context of SID of dynamical structures. A theoretical SDOF scenario was used in order to provide an example of where the RJMCMC method might be useful. The main important aspects treated within this chapter were the dimension matching issue of models with varying dimensions and the construction of a suitable Markov chain that can move between space of varying dimension while respecting the principle of detailed balance.

A full explanation of the principle of detailed balance was provided for both the MH algorithm and the RJMCMC algorithm in order to show how they might differ.

The next chapter, **Chapter 5** covers a series of numerical case studies were the author applied the RJMCMC algorithm in order to show its capabilities as a promising tool in SID of dynamical structures, prior to the application of the RJMCMC algorithm on real systems in **Chapter 6**.

Chapter 5

RJMCMC on Numerical Case Studies

Chapter 5 covers the numerical case studies where the RJMCMC was applied in order to prove its effectiveness and capability in the context of system identification within structural dynamics. This chapter covers four numerical case studies. The first numerical case study considers a SDOF system for which two models are available, one describing a liner system and one describing a nonlinear system. The second numerical study is an extension to the first one in such that it considers a MDOF system for which two competing models are available, the first describing a linear MDOF system and the second describing a nonlinear MDOF system with the same characterised type of nonlinearity as for the SDOF system. The third numerical case study takes into account a MDOF structure for which two competing models exist, one linear model and a nonlinear model with a bilinear stiffness element. The fourth numerical case study is an extension to the work done so far toward SHM. The RJMCMC algorithm was applied as explained in **Chapter 4**. The purpose of this work is to introduce the RJMCMC algorithm in the context of structural dynamics. As with any other algorithm or computational tool, the first step is to explain how it works, followed by examples of case studies. Firstly, numerical case studies will be covered, where the data is simulated by a computer, using software such as Matlab [79] for this particular case.

5.1 RJMCMC on a Nonlinear Numerical Case Study in System Identification - SDOF system

In order to demonstrate the potential of the RJMCMC method, the author applied it first on the case introduced in **Chapter 4**. Let one summarise the case study:

Assuming that one has a SDOF system for which two competing models exist as represented in **Figure 5.1** , a SDOF linear model and a SDOF nonlinear model of Duffing type , conduct SID on that system by applying the RJMCMC algorithm.

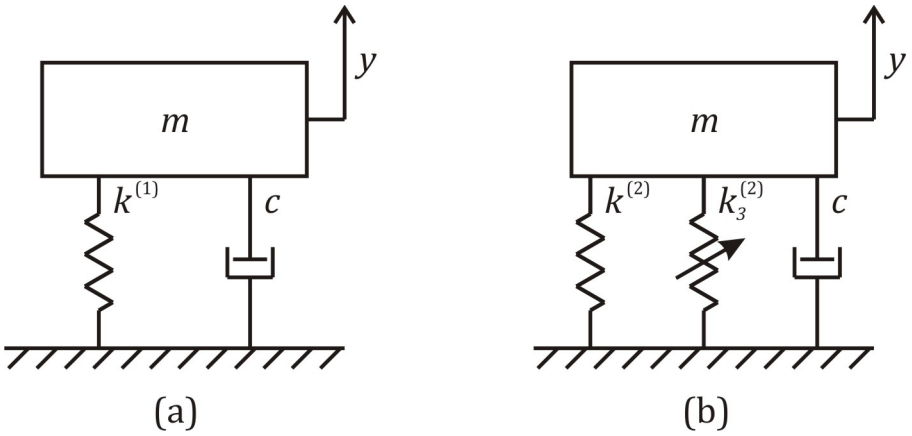

Figure 5.1: (a) Linear system, (b) Nonlinear system with cubic stiffness

The mathematical model for the linear system is:

$$m\ddot{y} + c\dot{y} + k^{(1)}y = F \tag{5.1}$$

while for the nonlinear system with a Duffing nonlinearity it takes the form:

$$m\ddot{y} + c\dot{y} + k^{(2)}y + k_3^{(2)}y^3 = F \tag{5.2}$$

where m is the mass of the structure, k is the stiffness, c is the damping coefficient, k_3 is the cubic stiffness, $\ddot{y}$ is the absolute acceleration of the structure, $\dot{y}$ is the absolute velocity of the structure, y is the absolute displacement of the structure, and finally F is the excitation.

The objectives of the SDOF numerical case study are as follows:

- Apply the RJMCMC algorithm to do SID on a SDOF simulated dynamical system;
- Show that the RJMCMC method can do parameter estimation of a SDOF simulated linear dynamical system;
- Show that the RJMCMC method can do parameter estimation of a SDOF simulated nonlinear dynamical system;
- Show that the RJMCMC method can select between a SDOF simulated linear model and a SDOF simulated nonlinear model accurately, simultaneously estimating parameters;

The training data used was simulated as 2000 points of displacement signal with artificial measurement noise and corresponding samples of excitation. Initially, the data was simulated using equation (5.1) which means that one has data from a linear model. This means that the algorithm should favor $M^{(1)}$ in the process of model selection and should provide an estimation of the linear stiffness of $M^{(1)}$, $k^{(1)}$.

Both models, $M^{(1)}$ and $M^{(2)}$, were simulated applying a fixed-step fourth-order Runge-Kutta method. The sampling frequency chosen was of $f_s = 100$ Hz which means a true step value of $h_t = 0.01$ s. The values of the parameters for the exemplar system were as follows: $m = 0.5kg$, $c = 0.1Ns/m$ and $k^{(1)} = 50N/m$. The values of the mass, m, damping coefficient, c and linear stiffness, $k^{(2)}$ for the nonlinear model, $M^{(2)}$ were kept the same and the cubic stiffness, $k_3^{(2)}$ was set as $k_3^{(2)} = 10^3 N/m^3$. The excitation used for both systems was a Gaussian white noise sequence of mean 0 and variance 100.

Model	$m(kg)$	$k(N/m)$	$c(Ns/m)$	$k_3(N/m^3)$	$f_s(Hz)$	$h_t(s)$
$M^{(1)}$	0.5	50	0.1	0	100	0.01
$M^{(2)}$	0.5	50	0.1	1000	100	0.01

Table 5.1: True parameter values for $M^{(1)}$ and $M^{(2)}$

After creating the training data, the RJMCMC algorithm was implemented to do

the system identification of the structure. The priors for the two models were kept equal which translates into the probability of proposing a birth being equal to the probability of proposing a death, $b_l = d_l = 0.25$ at the current state l. This results in the probability of proposing an update in the current model being $u_l = 0.5$. The number of iterations used was 10000 samples.

The posterior distributions for both models were created as separate functions that included the likelihoods and the priors of the parameters. The priors were set as lower and upper limits to the parameters. In the case of both models the linear stiffness k was kept between 0 and 100. Limits of 100 and 10^4 were set for the cubic stiffness,$k_3^{(2)}$ of model $M^{(2)}$. The proposal density was chosen to be a Gaussian distribution and the width of the proposal was set to 0.5 with the possibility of changing it in the tuning stage. The initial values of the parameters were set to be close to the values of the parameters for the exemplar system (also called true values). A burn in of 500 samples was chosen to allow the settling of the chains.

5.1.1 Results

It is only natural to first start with the training data simulated with the linear system. In this case, one expects the RJMCMC algorithm to correctly identify that the model to best represent the data should be the linear model. Under this circumstances, the algorithm should also provide an estimate for the linear stiffness, i.e. $k^{(1)}$. **Figure 5.2** and **Figure 5.3** demonstrate exactly that.

Figure 5.2 shows the results of applying the RJMCMC algorithm for model selection. What one can see is a bar chart of the number of iterations the algorithm spent in $M^{(1)}$, the linear model and $M^{(2)}$, the nonlinear model. It is clear from this visualisation that the RJMCMC algorithm strongly recommended the linear model to represent the linear system, as expected. The RJMCMC method spent the least time in $M^{(2)}$. The reason for this is the fact that the 'true' data is not from a real structure. The nonlinearity characterised in this example was designed to provide a strong nonlinear effect which led to a significant enough difference between the two sets of data.

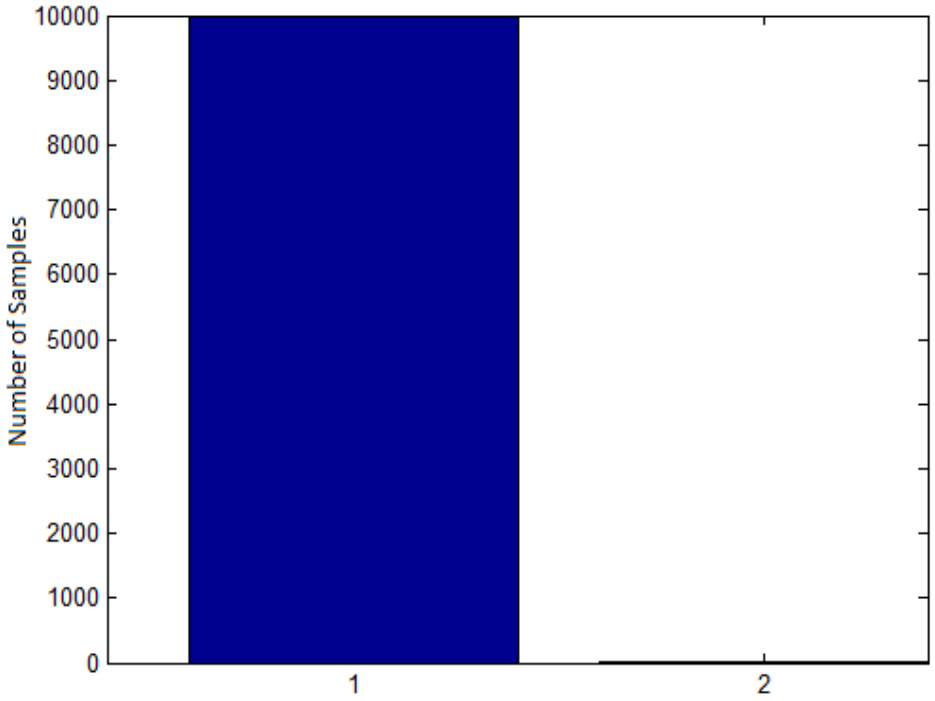

Figure 5.2: Acceptance frequency of Model 1 vs Model 2 - using linear training data

Figure 5.3 illustrates the parameter estimation of the $k^{(1)}$ parameter for the linear model. On the y axis one can find the frequency of the samples, while on the x axis one can observe the values of the samples. The red line indicates the 'true' value of the parameter, as imposed by the user when creating the training data. The RJMCMC successfully and accurately estimates it as being $k^{(1)} = 50N/m$. Lower in **Figure 5.3**, one can see the Markov chain of samples or where the chain travelled to before reaching stationarity. With the numerical case study being simulated, the starting point for the chains is chosen to be the 'true' values of the parameter so that the algorithm could find the estimate within a lower number of iterations.

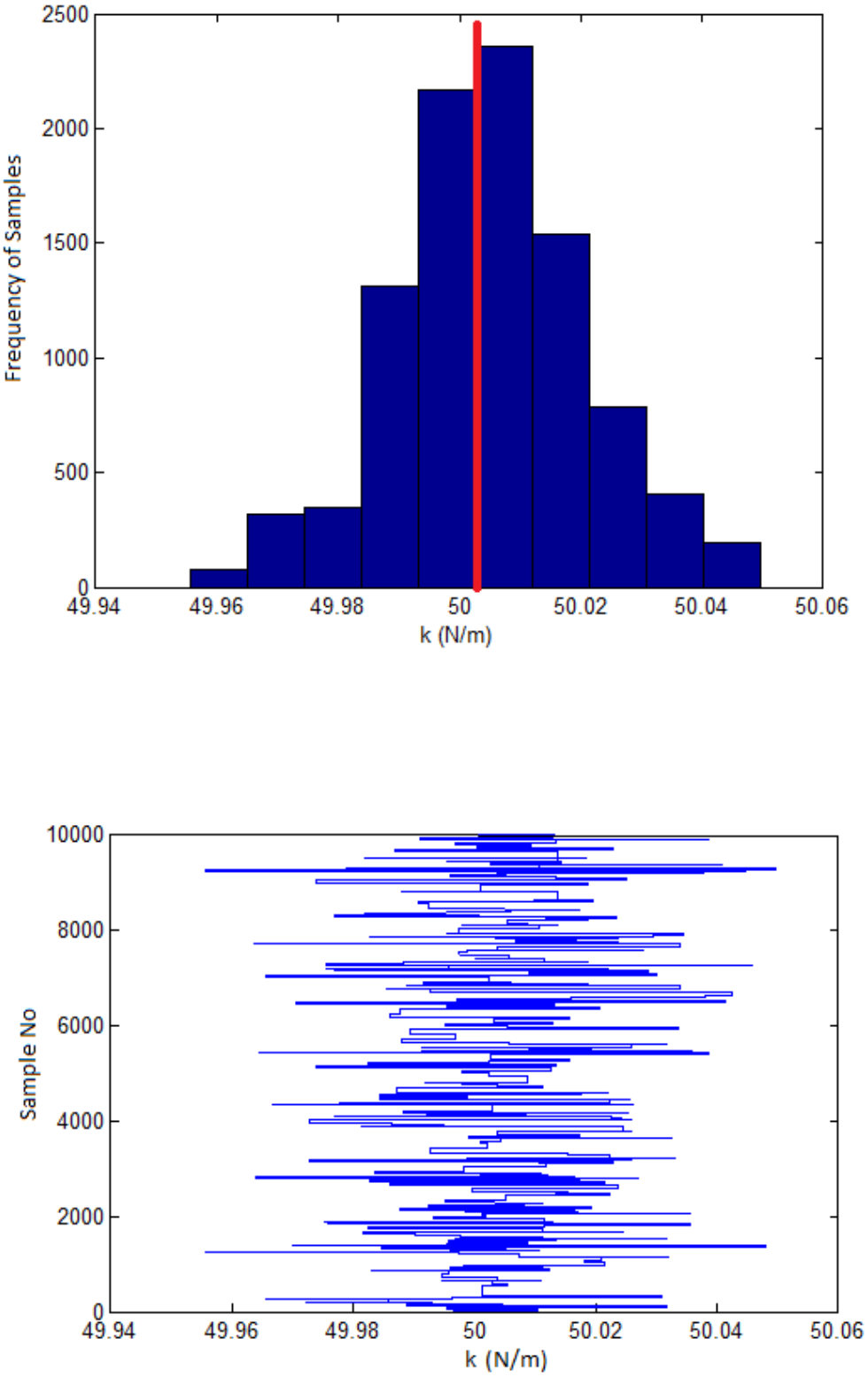

Figure 5.3: $M^{(1)}$ - parameter estimation of linear stiffness $k^{(1)}$(blue), most frequent value(red) and history of samples

The results of identifying a linear system are truly satisfactory in this study but the author was more concerned with the capabilities of the RJMCMC algorithm in proving its capabilities in nonlinear SID. So let one quickly move toward the second part of the SDOF study.

The second part of the study involves using the training data simulated with the nonlinear system by using equation (5.2). In this case, one expects the RJMCMC algorithm to correctly identify that the model to best represent the training data is the nonlinear or Duffing type, $M^{(2)}$ model and to also properly estimate the two unknown parameters, $k^{(2)}$ and $k_3^{(2)}$. One needs to understand that these case study are only numerical. The performance of the RJMCMC algorithm is expected to be higher as no external noise is included, as it would be the case in an experimental study. **Figure 5.4** and **5.5** provide a visualisation of the results obtained in applying the RJMCMC method.

In **Figure 5.4** one can see again, a bar chart for the number of iterations spent in each model. That clearly concludes that the RJMCMC algorithm showed a strong inclination toward the nonlinear model, $M^{(2)}$ as hoped. Due to the fact that the RJMCMC method is used within a Bayesian framework, it will try always to penalise complexity. For this reason, it can be seen that the algorithm visited the linear model several times. It was clear in the end though that the nonlinear model, $M^{(2)}$, fitted the data provided.

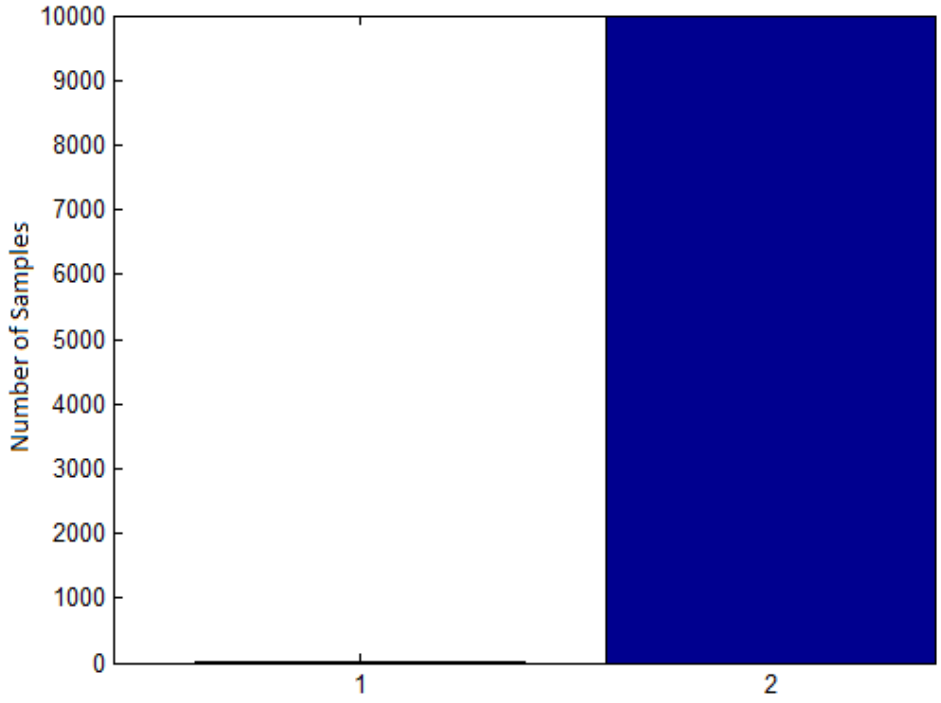

Figure 5.4: Acceptance frequency of Model 1 vs Model 2 - using nonlinear training data

Figure 5.5 shows the frequency of samples for the two estimated parameters, $k^{(2)}$ and $k_3^{(2)}$, while the red lines indicate the 'true' parameters as imposed by the user. What this shows is that the RJMCMC algorithm is a powerful candidate in doing SID in nonlinear system identification, an area that is still quite enigmatic for the structural engineer.

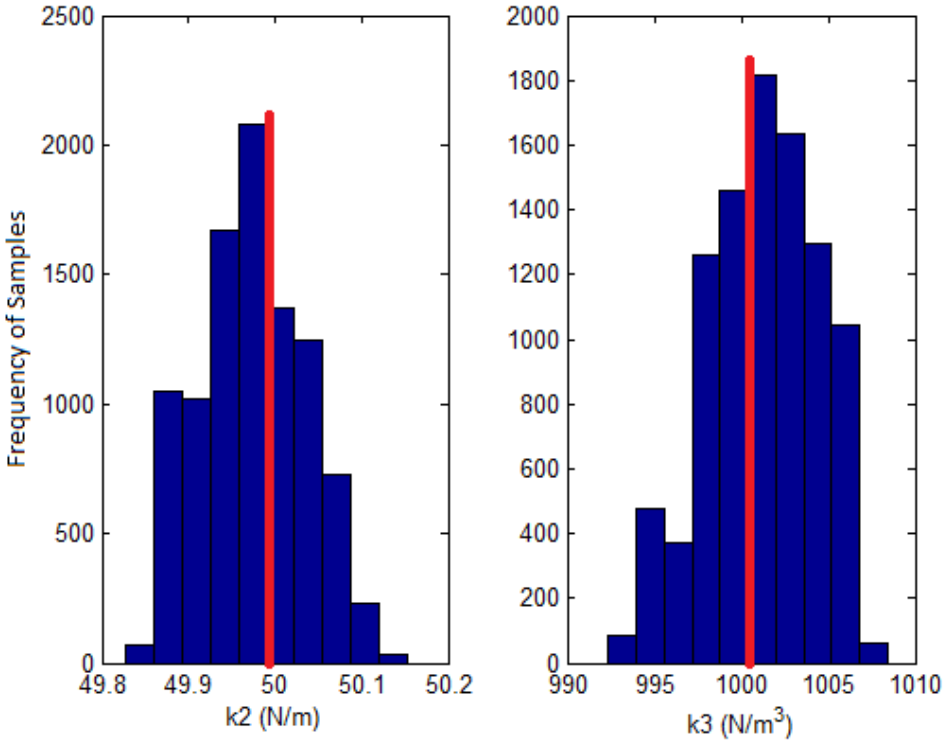

Figure 5.5: $M^{(2)}$ - parameter estimation of linear stiffness $k^{(2)}$ and nonlinear stiffness $k_3^{(2)}$(blue) and most frequent values(red)

Due to the fact that the considered case was a simulation and the user was aware of the 'true' parameter values, the Markov chains were set to start the iteration process at the 'true' values in order to save computational time. This being mentioned, the initial value are not required to be set near the 'true' values of the parameters in order for the algorithm to perform.

For a different perspective, let one have a look at the two sets of training data. In **Figure 5.6** one can observe the plots for the two sets of training data used. In blue one has the linear training data set, in red the nonlinear training data set. It is not surprising to observe that they illustrate different behaviour. One set of data is generated using the linear SDOF model and the other set of data is generated using the nonlinear SDOF model with a cubic stiffness.

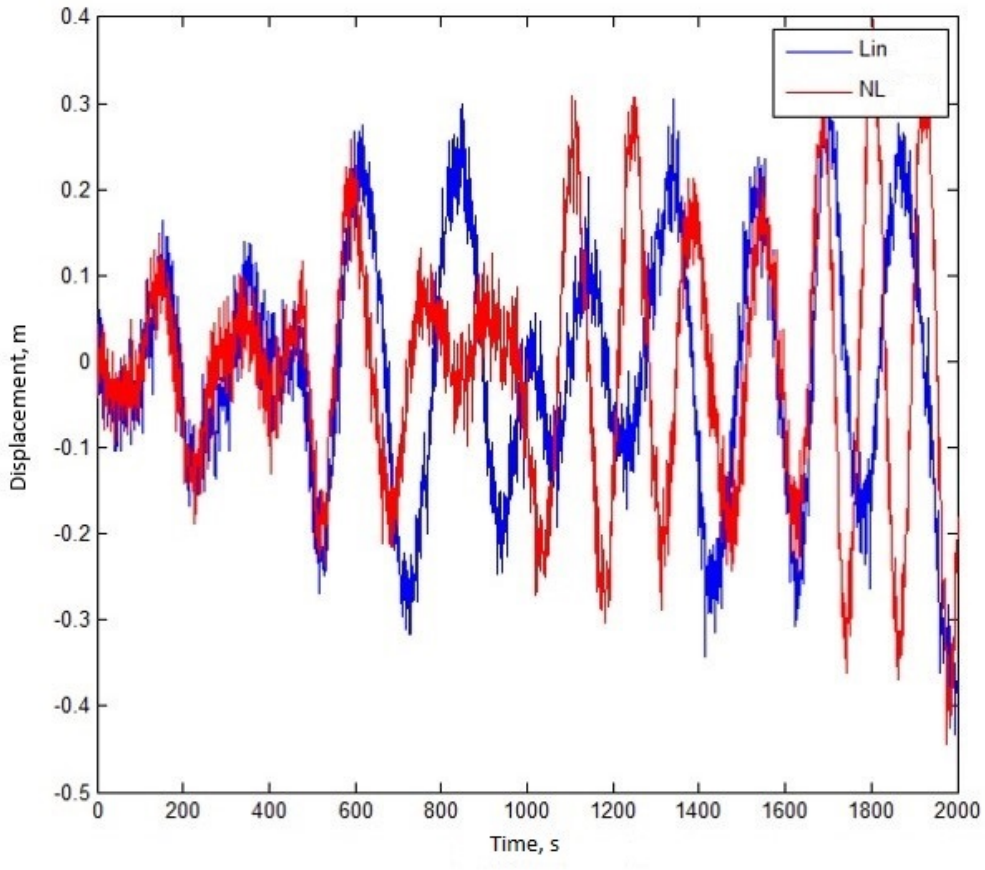

Figure 5.6: Linear vs Nonlinear training data sets

At a closer look though, in **Figure 5.7**, one can observe a reduced range of the two sets of training data. Now, the observation that can be made is that actually, up to data point 280, indicated on the plot, the data looks quite similar. This means, that if the RJMCMC algorithm had only 280 data points to work with and the training data would be from the nonlinear system, it would choose the less complex model to represent it by, the linear one, because the behaviour represented through that particular range of data is linear. This is what one expects to happen with an appropriate computational method. What one requires in the end, is a robust but simplest model for the system of study. At this point is where the Bayesian approach proves its effectiveness in preventing overfitting.

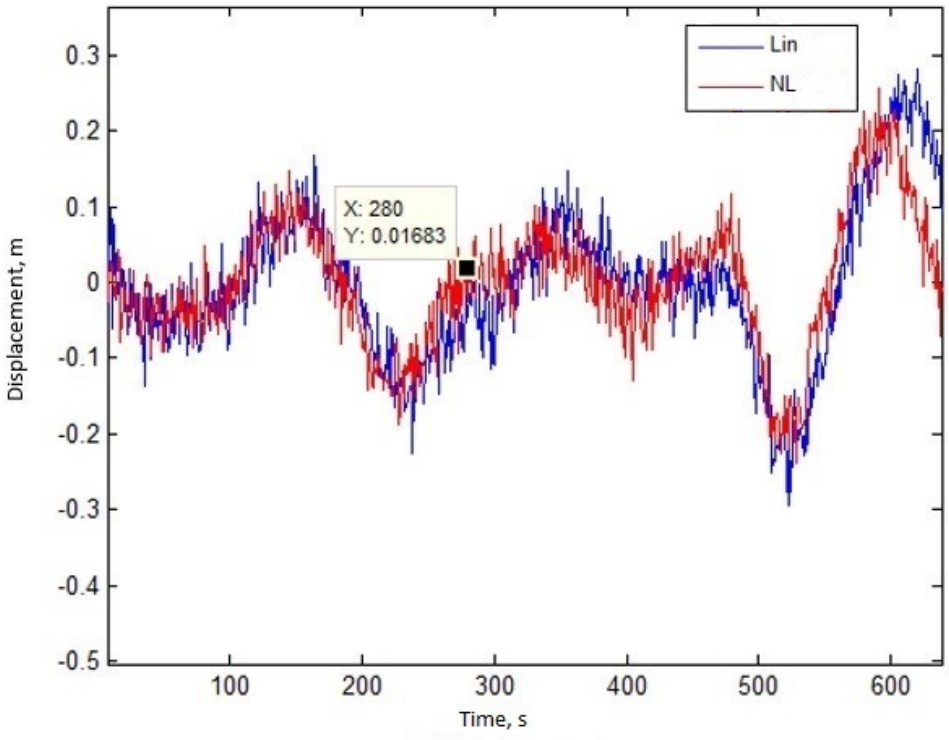

Figure 5.7: Linear vs Nonlinear training data sets-reduced range

For completion of the last objective, let one see what happens once more data is fed into the RJMCMC algorithm. This analysis is illustrated through a bar chart of Bayes factor in **Figure 5.8**. So, the training data used is from the nonlinear system. Due to the similarity in the linear training data and nonlinear training data up to the data point 280, one can see that the algorithm is choosing the linear model to best represent the nonlinear training data. Immediately after data point 280 though, the algorithm is choosing the nonlinear model to best represent the nonlinear training data. This is a very important analysis as it shows the sensitivity of the RJMCMC algorithm to data sets, which is relevant in scenarios where the main information one has about ones system is pure data.

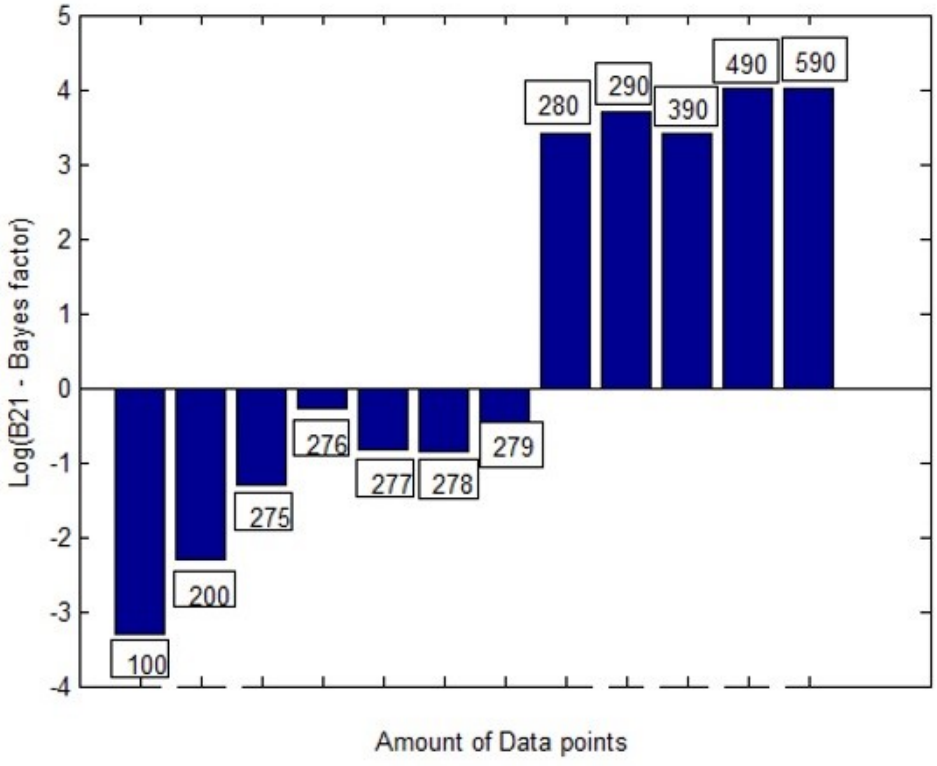

Figure 5.8: Bayes Factor comparison with increasing number of data points used for training

In this case, the Bayes' factor is obtained by dividing the likelihood of $M^{(2)}$ to the likelihood of $M^{(1)}$, taking into account the training data. If the Bayes' factor is less than 0, then the linear model is accepted. Otherwise, the nonlinear model is accepted.

The RJMCMC algorithm was applied so far on scenarios in which the system was randomly excited. This does not mean that the algorithm can only be applied to these type of system. With this in mind, the SDOF system was excited by a periodic excitation as seen in **Figure 5.9**.

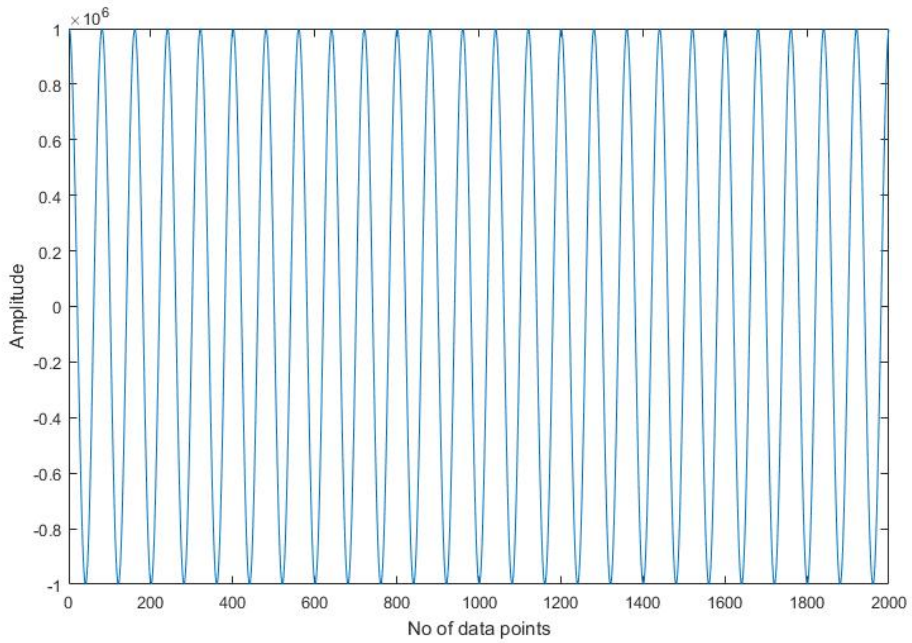

Figure 5.9: SDOF Periodic Excitation signal

The scenario considered was as before, a system that can be defined by a linear model, $M^{(1)}$, in which the parameter to be estimated is the linear stiffness, $k^{(1)}$ or by a nonlinear model, $M^{(2)}$, in which the parameters to be estimated are the linear stiffness, $k_2^{(2)}$ and the cubic stiffness, $k_3^{(2)}$. All of the parameters used in generating the training data were kept the same:

Model	$m(kg)$	$k(N/m)$	$c(Ns/m)$	$k_3(N/m^3)$
$M^{(1)}$	0.5	50	0.1	0
$M^{(2)}$	0.5	50	0.1	1000

Table 5.2: True parameter values for $M^{(1)}$ and $M^{(2)}$

The RJMCMC algorithm was applied first on the linear acceleration training data to which artificial noise was applied, see **Figure 5.10**.

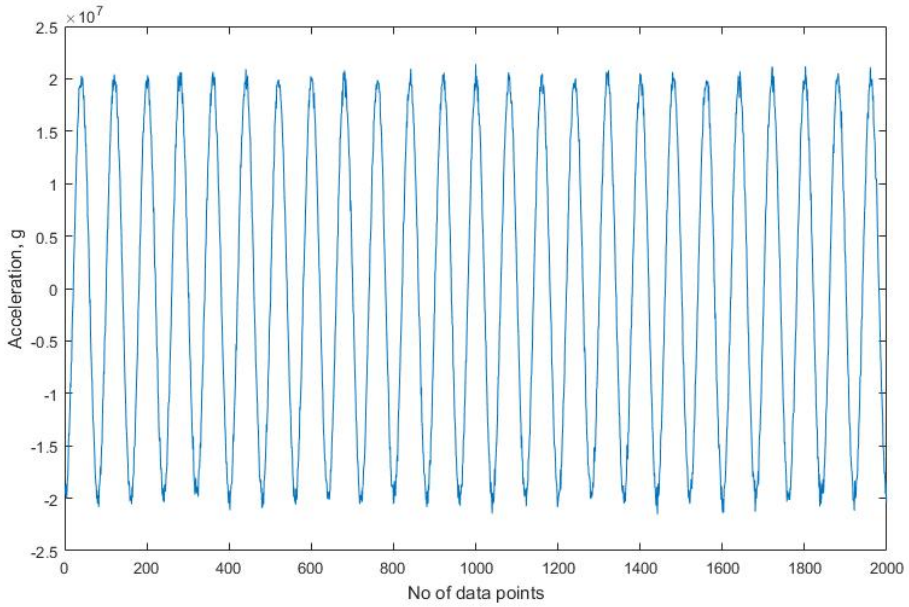

Figure 5.10: Linear output data from SDOF system under periodic excitation

A number of 5000 iterations were used. In **Figure 5.11**, the results of applying the RJMCMC method by using the linear training data are presented. The top plot in the figure represents the histogram of estimating the linear stiffness, $k^{(1)}$ while the second plot represents the Markov chain. As it was expected, the algorithm preferred the linear model to represent the data. This can be observed in the bar chart in **Figure 5.12**.

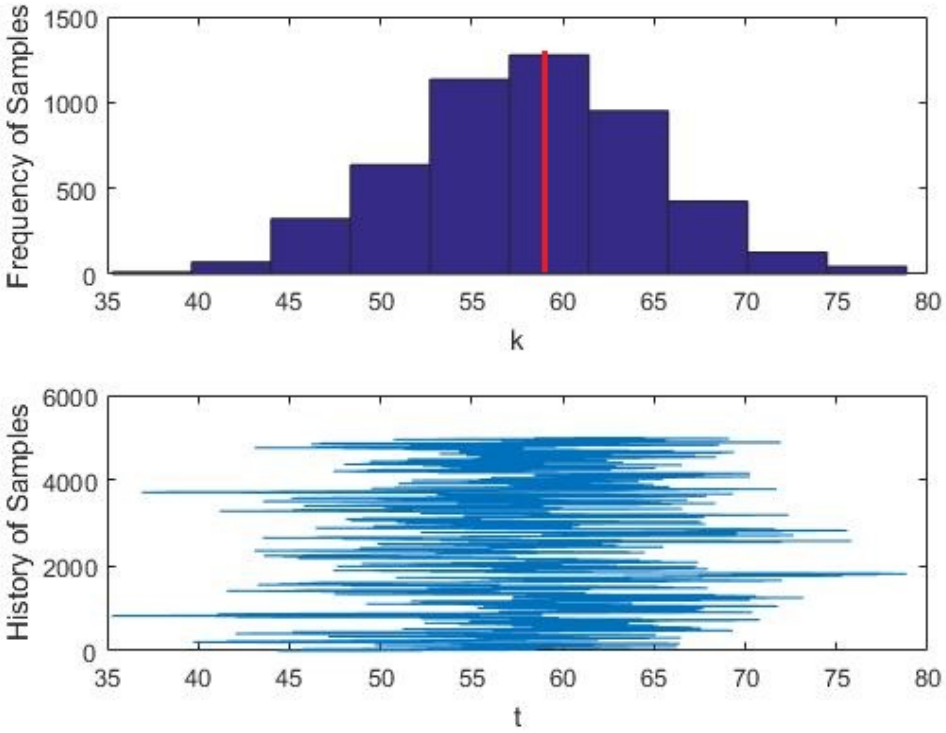

Figure 5.11: $M^{(1)}$ - parameter estimation of linear stiffness $k^{(1)}$(blue), most frequent value(red) and history of samples

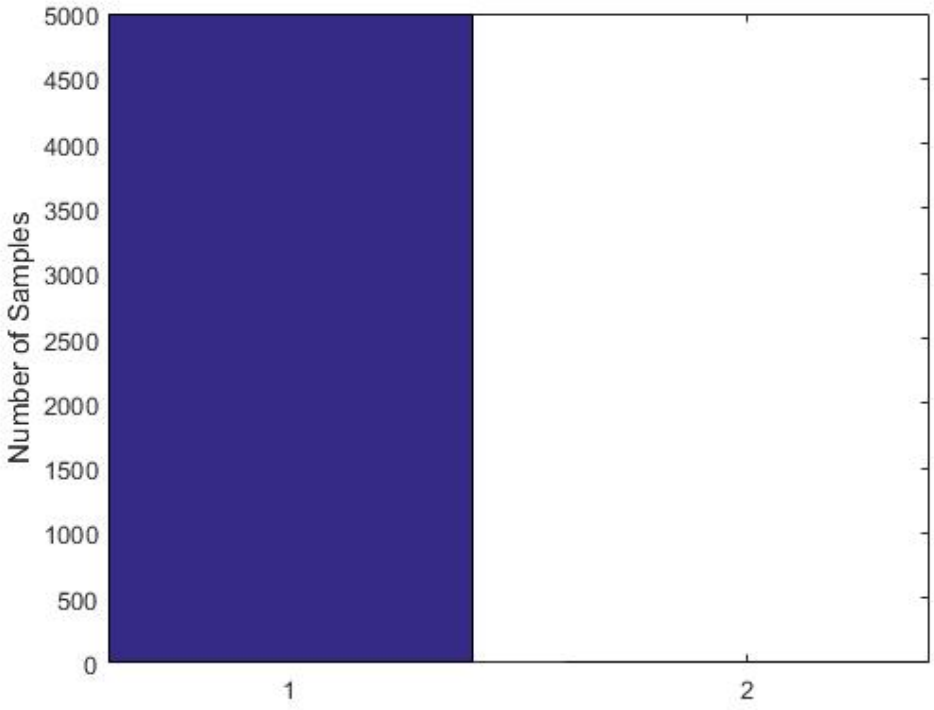

Figure 5.12: Acceptance frequency of Model 1 vs Model 2 - using linear training data

The second step was to apply the RJMCMC algorithm on the nonlinear training acceleration data set to which artificial noise was applied, see **Figure 5.13**.

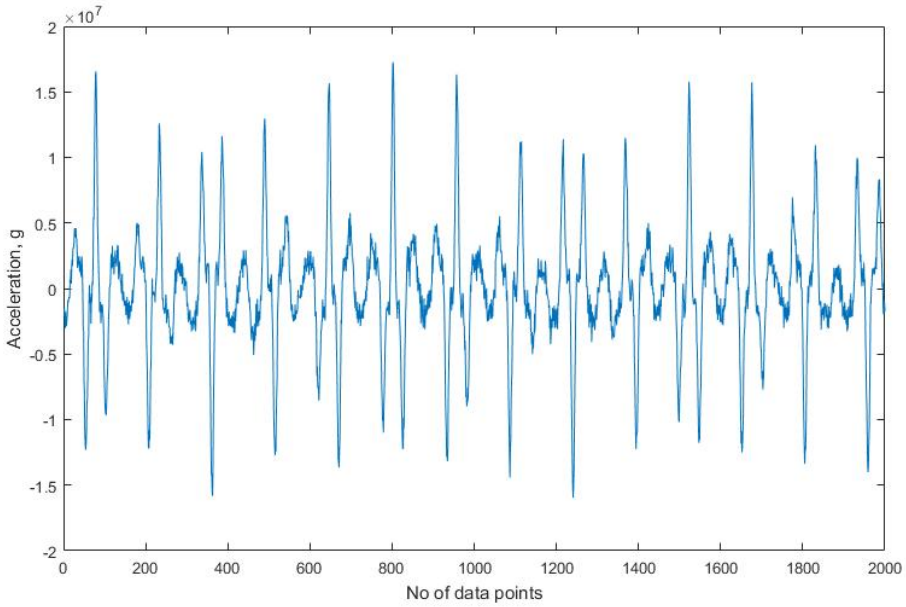

Figure 5.13: Nonlinear output data from SDOF system under periodic excitation

Again, a number of 5000 iterations of the algorithm were used. **Figure 5.14**, **Figure 5.15** and **Figure 5.16** present the results of applying the method on the nonlinear data.

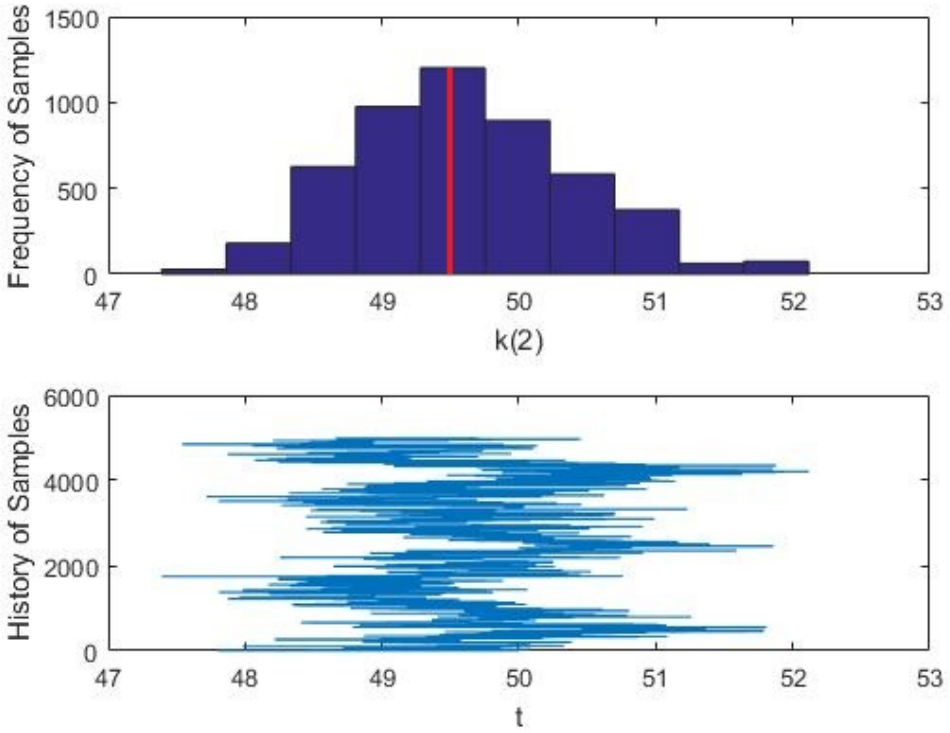

Figure 5.14: $M^{(2)}$ - parameter estimation of linear stiffness $k_2^{(2)}$(blue), most frequent value(red) and history of samples

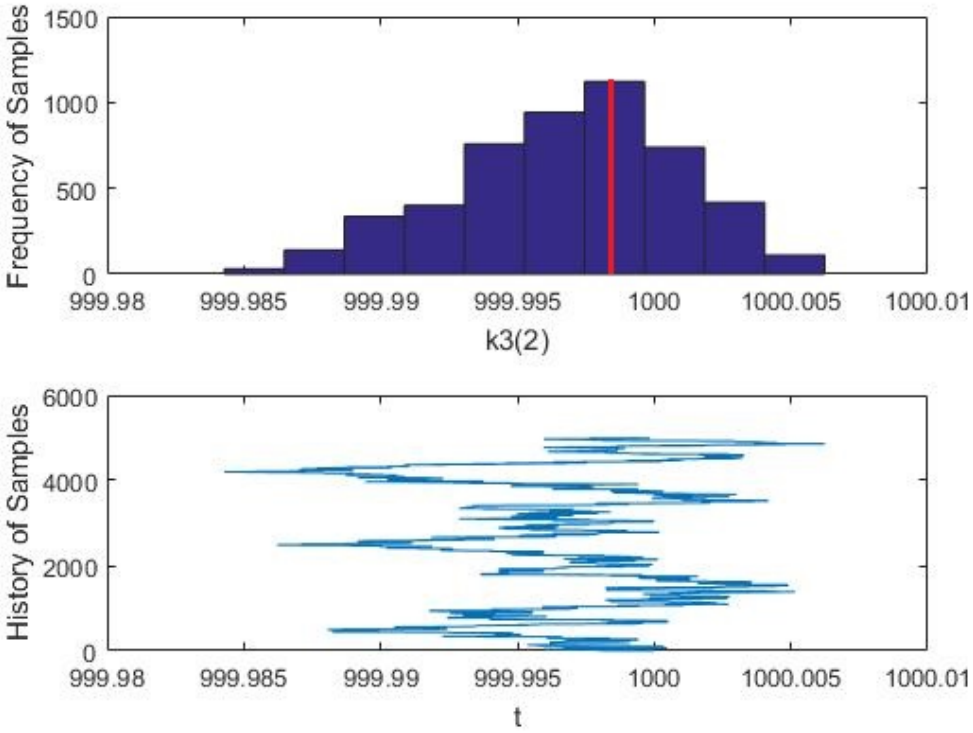

Figure 5.15: $M^{(2)}$ - parameter estimation of cubic stiffness $k_3^{(2)}$(blue), most frequent value(red) and history of samples

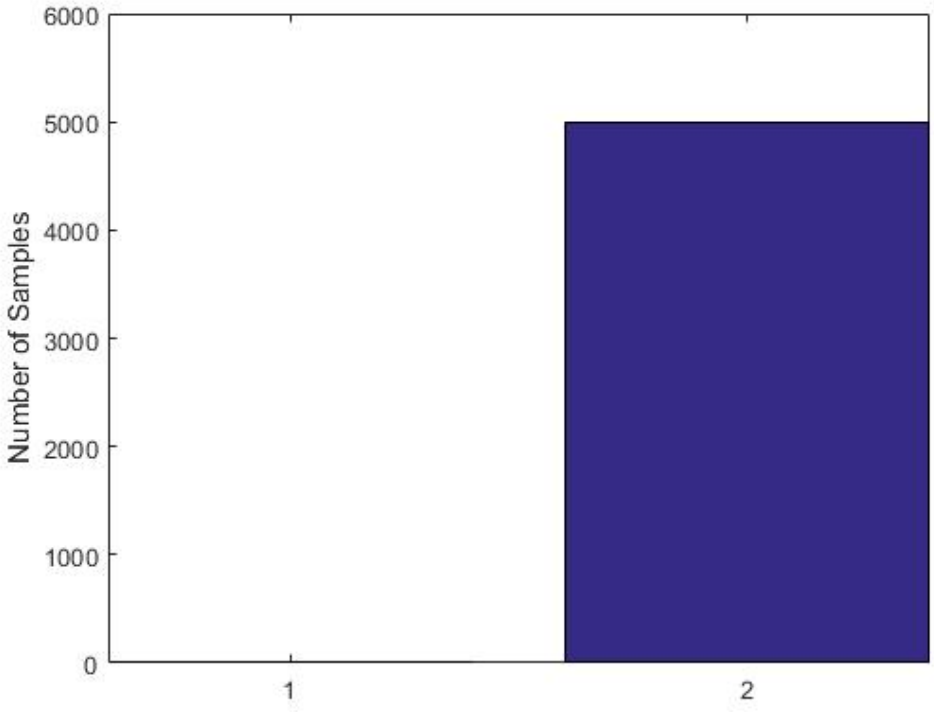

Figure 5.16: Acceptance frequency of Model 1 vs Model 2 - using nonlinear training data

As expected, the RJMCMC method chose the nonlinear model to represent the nonlinear data set. Both parameters, $k_2^{(2)}$ and $k_3^{(2)}$, were estimated. This last test was used in order to show that the RJMCMC algorithm can be used in nonlinear system identification with both random and periodic excitation.

5.2 RJMCMC on a Nonlinear Numerical Case Study in System Identification - Cubic Stiffness

Now that a SDOF nonlinear SID case was covered, it follows that the second numerical case study should be by increasing the number of degrees of freedom. It is by now well understood that once a system has more than 1-DOFs, the process of system identification can get a bit complicated. The computational time of running the RJMCMC algorithm also increases when the number of parameters to be iden-

tified increases. To such extent, for this case study, a choice was made to keep the damping coefficient, c as an unknown as well in order to assess the time needed for the algorithm to run.

For the MDOF case study, the following objectives were set:

- Apply the RJMCMC algorithm on a MDOF numerical case study in structural dynamics;
- Show that the RJMCMC method can accurately estimate parameters for a simulate linear MDOF system;
- Show that the RJMCMC method can accurately estimate parameters for a simulated nonlinear MDOF system;
- Show that the RJMCMC method can accurately choose between a linear and a nonlinear model for a simulated dynamical system;
- Asses the RJMCMC algorithm when the nonlinearity is moved between masses.

For this particular case study the author chose to look at a 3-DOFs system for which two competing models exist, a linear model and a nonlinear model with a cubic stiffness. The diagram of the linear system can be visualized in **Figure 5.17** while the diagram for the nonlinear system can be visualised in **Figure 5.18**.

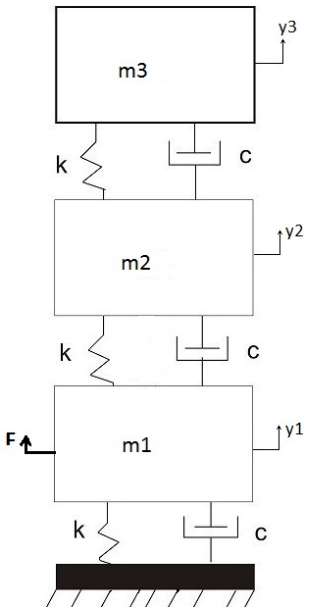

Figure 5.17: 3-DOFs Linear system

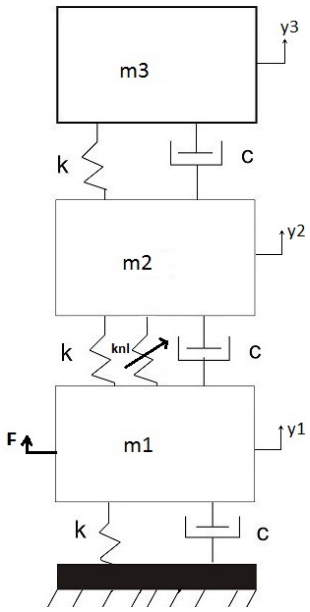

Figure 5.18: 3-DOFs Nonlinear system

For the system introduced above, the mathematical expression for the equations of motion for the linear model denoted $M^{(1)}$ is:

$$\begin{cases} m_1\ddot{y}_1 + 2c\dot{y}_1 - c\dot{y}_2 + 2ky_1 - ky_2 = F \\ m_2\ddot{y}_2 - c\dot{y}_1 + 2c\dot{y}_2 - c\dot{y}_3 - ky_1 + 2ky_2 - ky_3 = 0 \\ m_3\ddot{y}_3 - c\dot{y}_2 + c\dot{y}_3 - ky_2 + ky_3 = 0 \end{cases} \quad (5.3)$$

where m_1, m_2, m_3 are the masses of levels 1, 2 and 3 (counting from the lowest level to the highest level), $\ddot{y}_1$, $\ddot{y}_2$ and $\ddot{y}_3$ are the absolute accelerations of each corresponding level , $\dot{y}_1$, $\dot{y}_2$ and $\dot{y}_3$ are the absolute velocities of each corresponding level, y_1, y_2 and y_3 are the absolute displacements of each corresponding level and finally F is the excitation.

In a similar way, the equations of motion for the nonlinear model (denoted $M^{(2)}$) that includes the cubic stiffness are:

$$\begin{cases} m_1\ddot{y}_1 + 2c\dot{y}_1 - c\dot{y}_2 + 2ky_1 - ky_2 + k_{nl}(y_2 - y_1)^3 = F \\ m_2\ddot{y}_2 - c\dot{y}_1 + 2c\dot{y}_2 - c\dot{y}_3 - ky_1 + 2kz_2 - kz_3 - k_3(y_2 - y_1)^3 = 0 \\ m_3\ddot{y}_3 - c\dot{y}_2 + c\dot{y}_3 - ky_2 + ky_3 = 0 \end{cases} \quad (5.4)$$

As before, k is linear stiffness, c is damping coefficient and the only addition compared to the linear equations of motion is the k_3 term which is the cubic stiffness term.

The training data sets used were simulated, 2000 points of displacement signal with artificial measurement noise and corresponding samples of excitation.

Both models, $M^{(1)}$ and $M^{(2)}$, were simulated, applying a fixed-step fourth-order Runge-Kutta method. The sampling frequency chosen was of $f_s = 100$ Hz corresponding to a true step value of $h_t = 0.01$ s. The values of the parameters for the exemplar system were as follows: $m_1 = m_2 = m_3 = m = 5kg$, $c = 0.1Ns/m$ and $k^{(1)} = 50N/m$. The values of the mass, m, damping coefficient, c and linear stiffness, $k^{(2)}$ for the nonlinear model, $M^{(2)}$ were kept the same and the cubic stiffness, $k_3^{(2)}$ was set as $1000N/m^3$. The excitation used for both systems was a Gaussian white noise sequence of mean 0 and variance 1.

Model	$m(kg)$	$k^{(1)}/k^{(2)}(N/m)$	$c(Ns/m)$	$k_3^{(2)}(N/m^3)$	$f_s(Hz)$	$h_t(s)$
$M^{(1)}$	0.5	50	0.1	0	100	0.01
$M^{(2)}$	0.5	50	0.1	1000	100	0.01

Table 5.3: True parameter values for $M^{(1)}$ and $M^{(2)}$

After creating the training data, the RJMCMC algorithm was implemented using MATLAB to do the system identification of the 3-DOFs structure. The priors for the two models were kept equal which translates into the probability of proposing a birth move being equal to the probability of proposing a death move, $b_l = d_l = 0.25$ at the current state l. The probabilities of proposing a birth move and a death move were kept the same so that there is no bias in the choice. These probabilities need to be chosen according to the best knowledge of the user. This results in the probability of proposing an update in the current model being $u_l = 0.5$. The number

of iterations used was 5000 samples.

The posterior distributions for both models were created as separate functions that included the likelihoods and the priors of the parameters. The priors were set as lower and upper limits to the parameters. In the case of both models the linear stiffness k was kept between 0 and 100 and the damping coefficient c was kept between 0 and 10. Limits of 0 and 1100 were set for the cubic stiffness,$k_3^{(2)}$ of model $M^{(2)}$. The proposal density was chosen to be a Gaussian distribution and the width of the proposal was set to 0.1 for each parameter, with the possibility of changing it in the tuning stage. The initial values of the parameters were set to be close to the values of the parameters for the exemplar system (also called true values).

5.2.1 Results

As with the SDOF system, the starting point was using the linear training data generated by the linear 3-DOFs system.

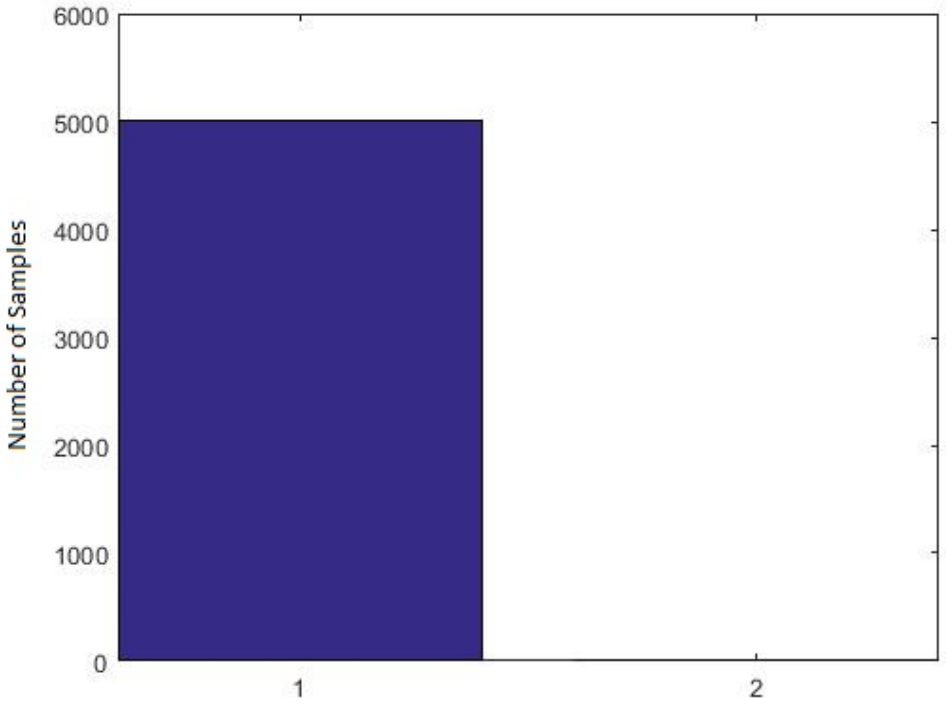

Figure 5.19: Acceptance frequency of Model 1 vs Model 2 - using linear training data

In **Figure 5.19** one is presented with the results of the comparison of numbers of samples spent in $M^{(1)}$ versus the number of samples spent in $M^{(2)}$. As expected, the algorithm spent its time in the linear model, concluding without a doubt that the linear model is the best choice in representing the linear system.

Figure 5.20 shows the estimates of the two parameters of interest, $k^{(1)}$ and c, with the red line indicating the 'true' values imposed by the user. Looking at the mean values, one can conclude that even in the case of MDOF system identification the RJMCMC algorithm proves its effectiveness.

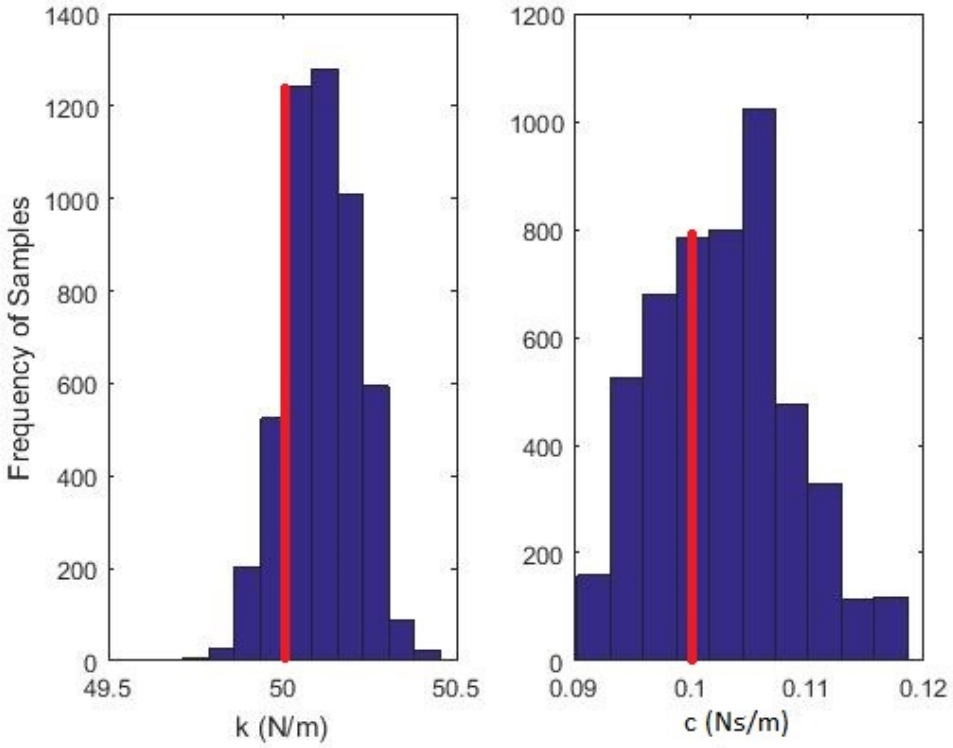

Figure 5.20: $M^{(1)}$ - parameter estimation of linear stiffness $k^{(1)}$ and damping coefficient c(blue) and most frequent values(red)

It follows that the second part of the case study and the most relevant one was using the algorithm with the nonlinear training data set.

In **Figure 5.21** one can observe that the RJMCMC algorithm spent most of its time in the second model, which proves its capabilities of doing SID for a MDOF nonlinear system.

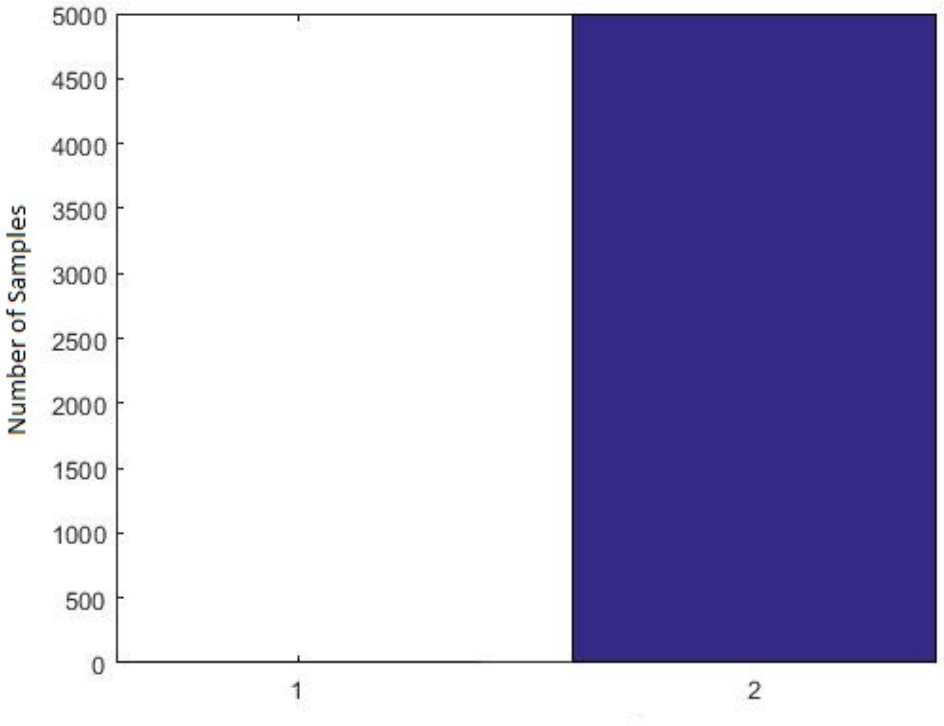

Figure 5.21: Acceptance frequency of Model 1 vs Model 2 - using non-linear training data

Figure 5.22 supports the success of using the RJMCMC method for SID of non-linear systems by showing an accurate estimation of the parameters, with the red line representing, as before, the 'true' values proposed by the user.

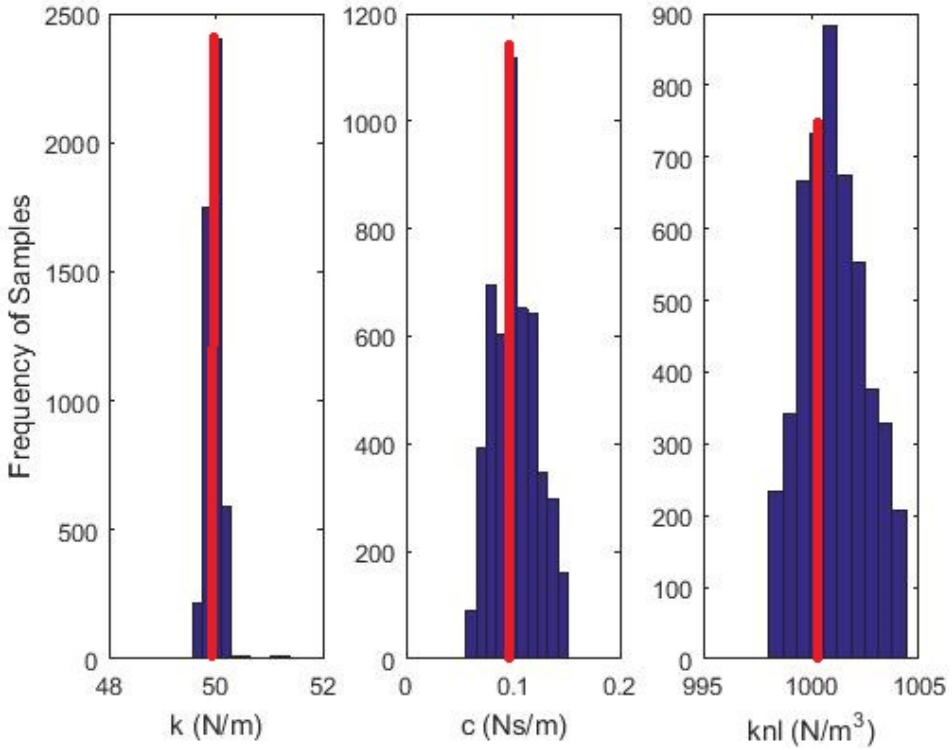

Figure 5.22: $M^{(2)}$ - parameter estimation of linear stiffness $k^{(1)}$, damping coefficient c and cubic stiffness $k_3^{(2)}$(blue) and most frequent values(red)

The results of applying the RJMCMC algorithm on the scenario discussed above are tabulated as follows:

Model	$m(kg)$	$k^{(1)}/k^{(2)}(N/m)$	$c(Ns/m)$	$k_3^{(2)}(N/m^3)$
$M^{(1)}$	0.5	50	0.1	0
$M^{(2)}$	0.5	50	0.1	1000

Table 5.4: Most frequent parameter values for $M^{(1)}$ and $M^{(2)}$

The expectation was for the RJMCMC algorithm to correctly choose the model and identify its parameters according to the training data made available. That expectation was met. The starting values for the Markov chains were the 'true' parameter values already known by the user in order to save computational time. In the case of a real system, those values will not be known a priori which translates into a considerable increase in computational time before convergence of the Markov chains into stationary chains. In this case no burn-in of the chains was necessary

and no thinning of the chains was used as the computational time was in the range of minutes(i.e. no time needed to be saved) and the number of samples was low(i.e. no space needed to be saved).

A study was conducted to observe the RJMCMC algorithm when the nonlinearity is moved between floors. **Figure 5.23** shows the three sets of data generated for each case of nonlinearity placement for all three levels of the simulated structure.

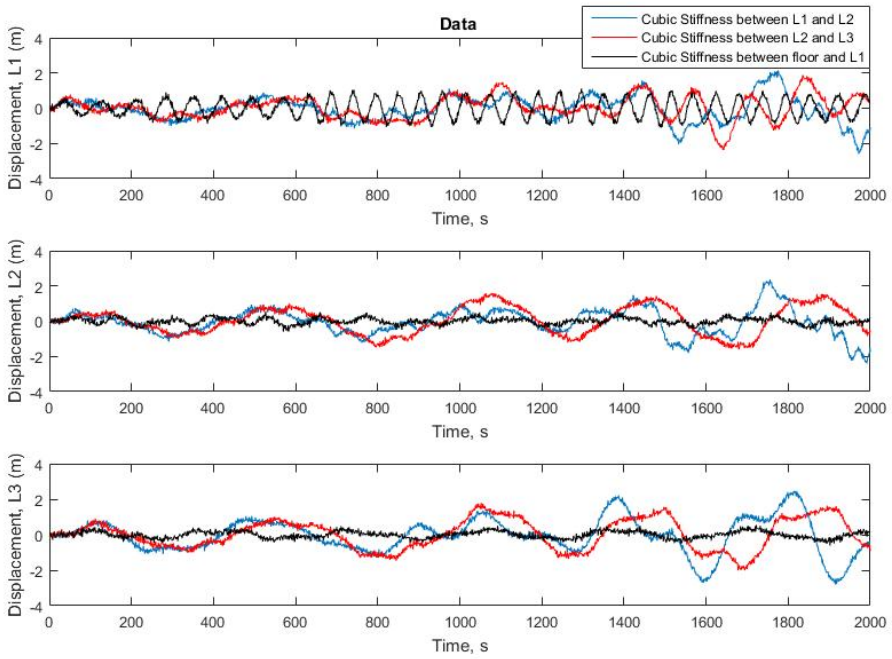

Figure 5.23: Data sets for nonlinear study

Initially, the cubic stiffness was placed between m_1 and m_2. The following results are from the nonlinearity placed between m_2 and m_3. The RJMCMC algorithm was applied on the newly generated data in order to do system identification.

Figure 5.24 and **Figure 5.25** represent the results of applying the RJMCMC algorithm on the nonlinear data with the nonlinearity between m_1 and m_2. The first figure is a plot of the histograms for the three identified parameters, all three centered around the true values of the parameters. The second figure is a plot of the samples. A number of 100 samples were discarded in the burn-in stage in order to account for the transition period. For this case, 10000 iterations were used as before for the RJMCMC algorithm but the obtained results suggest that the algorithm

could be ran for a higher number of iterations, but this comes at a time cost. $M^{(2)}$ was chosen over $M^{(1)}$ by a ratio of 9997 to 5.

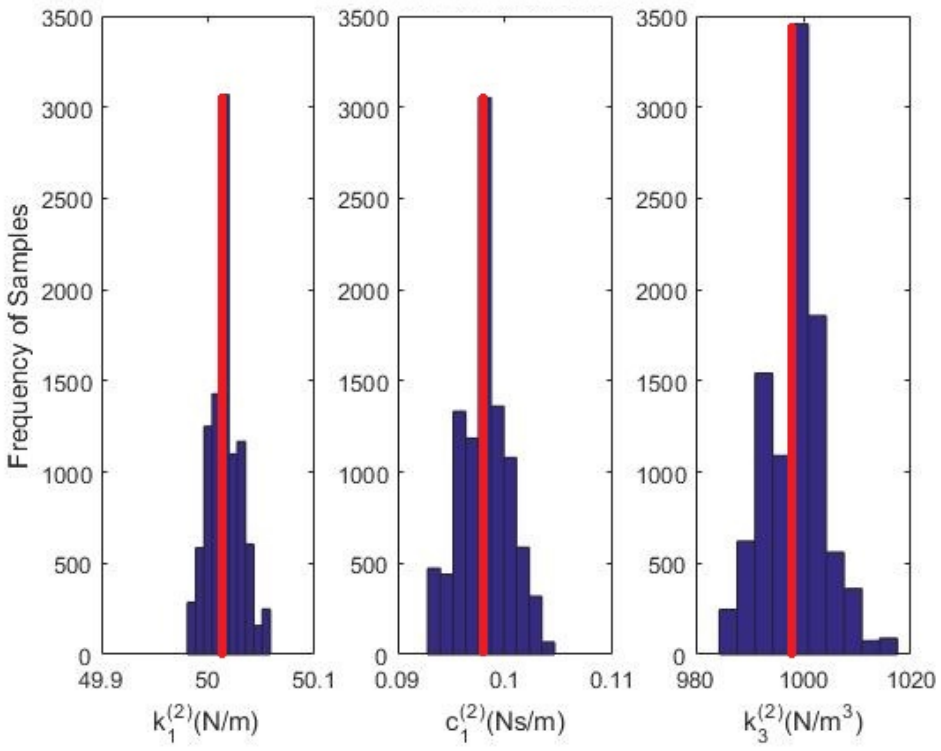

Figure 5.24: $M^{(2)}$ - parameter estimation of linear stiffness $k^{(1)}$, damping coefficient c and cubic stiffness $k_3^{(2)}$(blue) and most frequent values(red)

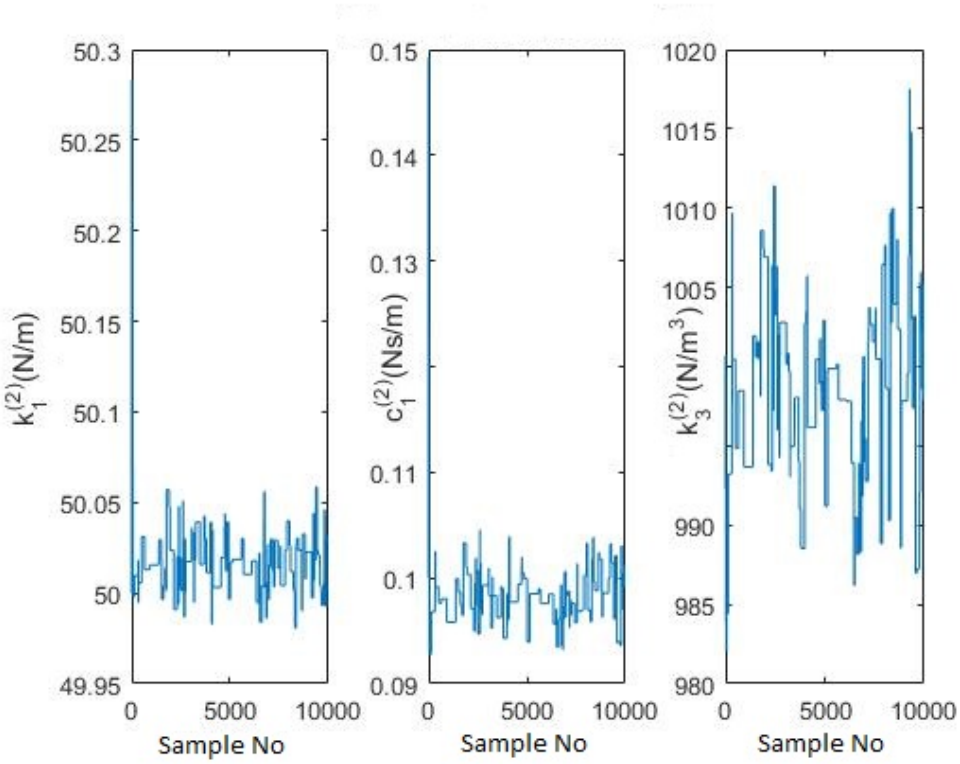

Figure 5.25: $M^{(2)}$ - history of the samples of parameters

The last application was with the cubic stiffness between the base floor and m_1. The RJMCMC was applied on the nonlinear data and the results are presented in **Figure 5.26** and **Figure 5.27**. The histograms for the identified parameters are plotted in **Figure 5.26**. It can be noticed that a higher number of iterations could provide smoother results. For comparison purposes, the number of iterations for the RJMCMC algorithm were kept the same throughout the three cases, 10000 iterations. A closer look at the chains in **Figure 5.27** suggests that the proposal density can be tuned further for an improved acceptance of samples. This raises the issue of automation for the algorithm. Nonetheless, the RJMCMC identified that $M^{(2)}$ is best suited to fit the data than $M^{(1)}$ by a ratio of 9998 to 4.

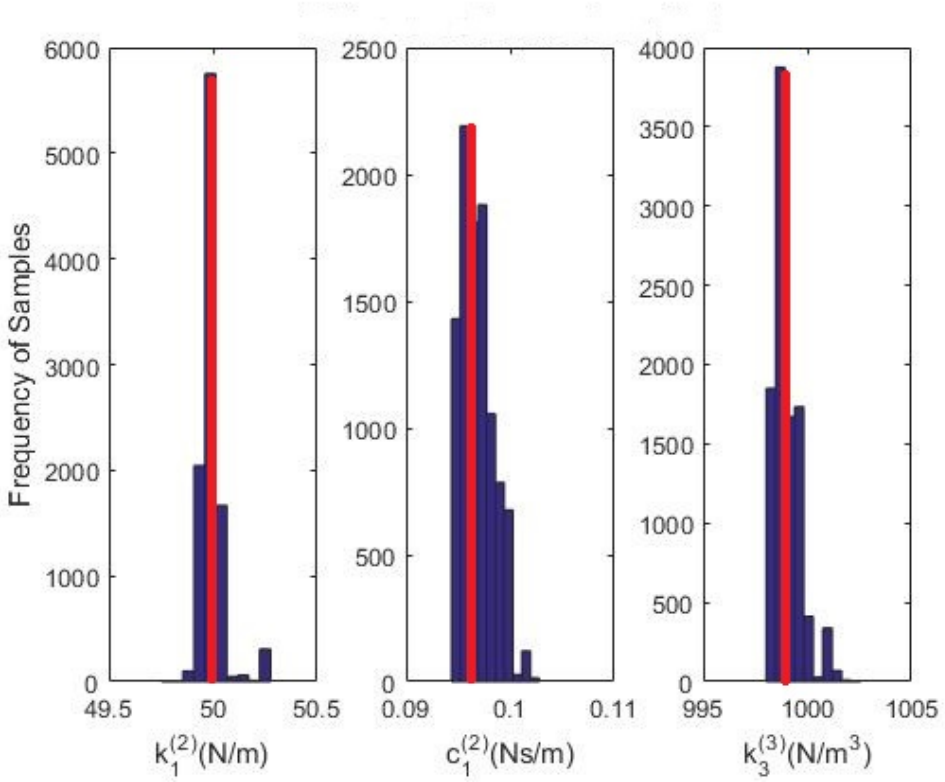

Figure 5.26: $M^{(2)}$ - parameter estimation of linear stiffness $k^{(1)}$, damping coefficient c and cubic stiffness $k_3^{(2)}$(blue) and most frequent values(red)

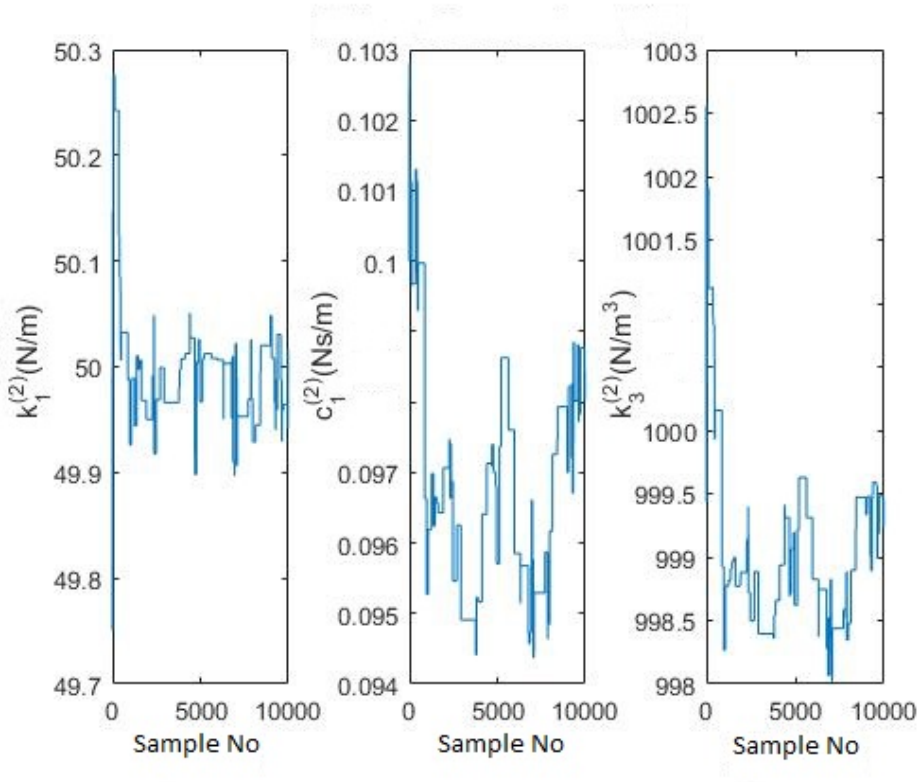

Figure 5.27: $M^{(2)}$ - history of the samples of parameters

The following case study is ran on a similar structure, but with the nonlinearity replaced for a bilinear stiffness. The sensitivity of the RJMCMC algorithm to the change in stiffness is studied.

5.3 RJMCMC on a Nonlinear Numerical Case Study - Bilinear Stiffness

Another MDOF numerical case study is introduced, this time with a nonlinearity caused by a bilinear stiffness behaviour. This nonlinear scenario comes closer to the experimental work conducted in **Chapter 6** and as such it is relevant in the present context. The aims and objectives were kept similar to the previous case study:

- Apply the RJMCMC algorithm on a MDOF numerical case study in structural dynamics;
- Show that the RJMCMC method can accurately estimate parameters for a

simulated linear MDOF system;

- Show that the RJMCMC method can accurately estimate parameters for a simulated nonlinear MDOF system;
- Show that the RJMCMC method can accurately choose between a linear and a nonlinear model for a simulated dynamical system;
- Asses the effects of using the RJMCMC algorithm at different stiffness values.

The studied scenario involves a 3-DOF structure for which two competing models are considered. The structure can display linear and nonlinear behaviour represented by a bilinear stiffness.

The linear behaviour is assigned a linear mathematical model for which the equations of motion are:

$$\begin{cases} m_1\ddot{y}_1 + 2c_1^{(1)}\dot{y}_1 - c_1^{(1)}\dot{y}_2 + 2k_1^{(1)}y_1 - k_1^{(1)}y_2 = F \\ m_2\ddot{y}_2 - c_1^{(1)}\dot{y}_1 + 2c_1^{(1)}\dot{y}_2 - c_1^{(1)}\dot{y}_3 - k_1^{(1)}y_1 + 2k_1^{(1)}y_2 - k_1^{(1)}y_3 = 0 \\ m_3\ddot{y}_3 - c_1^{(1)}\dot{y}_2 + c_1^{(1)}\dot{y}_3 - k_1^{(1)}y_2 + k_1^{(1)}y_3 = 0 \end{cases} \quad (5.5)$$

The linear system is described by three lumped masses, m_1, m_2 and m_3. Under external forcing, displacements y_1, y_2 and y_3 can be observed for each mass respectively. The forcing is applied on the first lumped mass, m_1 as seen in **Figure 5.28**.

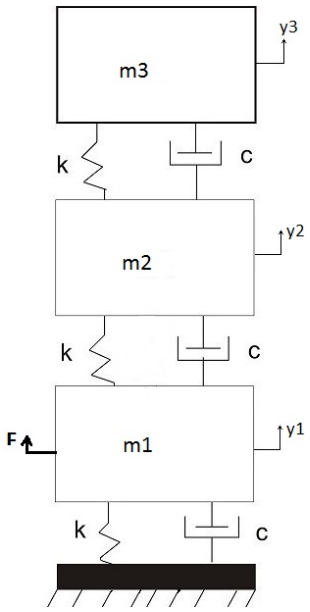

Figure 5.28: 3-DOFs Linear system

The nonlinear system is similar to the linear one, with the added element of a bilinear stiffness between masses m_1 and m_2 as pictured in **Figure 5.29**.

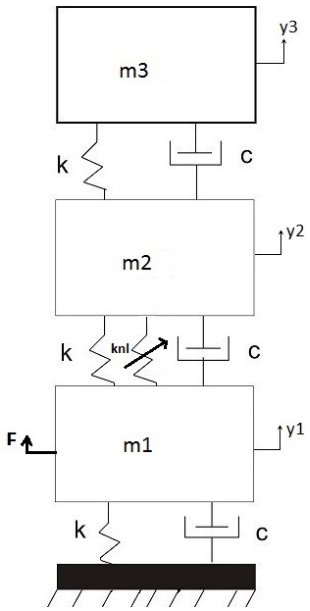

Figure 5.29: 3-DOFs Nonlinear system

The nonlinear system is described by the following equations of motion:

$$\begin{cases} m_1\ddot{y}_1 + 2c_1^{(2)}\dot{y}_1 - c_1^{(2)}\dot{y}_2 + 2k_1^{(2)}y_1 - k_1^{(2)}y_2 - k_2^{(2)}(y_2 - y_1) = F \\ m_2\ddot{y}_2 - c_1^{(2)}\dot{y}_1 + 2c_1^{(2)}\dot{y}_2 - c_1^{(2)}\dot{y}_3 - k_1^{(2)}y_1 + 2k_1^{(2)}y_2 - k_1^{(2)}y_3 + k_2^{(2)}(y_2 - y_1) = 0 \\ m_3\ddot{y}_3 - c_1^{(2)}\dot{y}_2 + c_1^{(2)}\dot{y}_3 - k_1^{(2)}y_2 + k_1^{(2)}y_3 = 0 \end{cases} \tag{5.6}$$

where $k_2^{(2)}$ is the added stiffness and $k_2^{(2)} = 0$ if $(y_2 - y_1) < 0$.

Training data is generated from both models in order to apply the RJMCMC algorithm. For both systems, a sampling frequency $f_s = 100Hz$ was used, step size $h_t = 0.01s$. The used parameters were $m_1 = m_2 = m_3 = m = 5kg$, $k_1^{(2)} = 50N/m$, $c_1^{(2)} = 0.1Ns/m$ and for the bilinear case, $k_2^{(2)} = 1000N/m$.

Model	$m(kg)$	$k_1^{(1)}/k_1^{(2)}(N/m)$	$c_1^{(1)}/c_1^{(2)}(Ns/m)$	$k_2^{(2)}(N/m)$	$f_s(Hz)$	$h_t(s)$
$M^{(1)}$	5	50	0.1	0	100	0.01
$M^{(2)}$	5	50	0.1	1000	100	0.01

Table 5.5: True parameter values for $M^{(1)}$ and $M^{(2)}$

The RJMCMC algorithm was then applied. The probability of proposing a birth move was kept equal to the probability of proposing a death move, $b_l = d_l = 0.25$ at the current state l. The probabilities of proposing a birth move and a death move were kept the same so that there is no bias in the choice of model. This resulted in the probability of proposing an update in the current model being $u_l = 0.5$. The number of iterations used was 10000 samples.

The priors were chosen to be uniform distributions set as lower and upper limits to the parameters. In the case of both models, the linear stiffness limits for $k_1^{(1)}$ and $k_1^{(2)}$ were kept between 0 and 100 and for the damping coefficients, $c_1^{(1)}$ and $c_1^{(2)}$, between 0 and 10. Limits of 0 and 2000 were set for the bilinear stiffness,$k_2^{(2)}$. The proposal density was chosen to be a Gaussian distribution and the width of the proposal was tuned by trial and error.

5.3.1 Results

As with all the other case studies, the first step was to make sure the RJMCMC algorithm is doing what is expected: parameter estimation and model selection when the appropriate data is provided. To such extent, the RJMCMC algorithm was applied on linear training data and nonlinear training data with the expectation of it choosing the correct model and providing the parameters of interest. In this case, for the linear model, the parameters to be identified were the stiffness, $k_1^{(1)}$ and damping coefficient $c_1^{(1)}$. For the nonlinear model, the parameters of interest were the stiffness $k_1^{(2)}$, damping coefficient $c_1^{(2)}$ and the second stiffness $k_2^{(2)}$.

Figure 5.30 and **Figure 5.31** represent the results of applying the RJMCMC algorithm on the case when the linear data was made available. As expected, the linear model is identified as the best to fit the data while the parameters are identified, with the mean standing close to the true values, $k_1^{(1)} = 50N/m$, $c_1^{(1)} = 0.1Ns/m$.

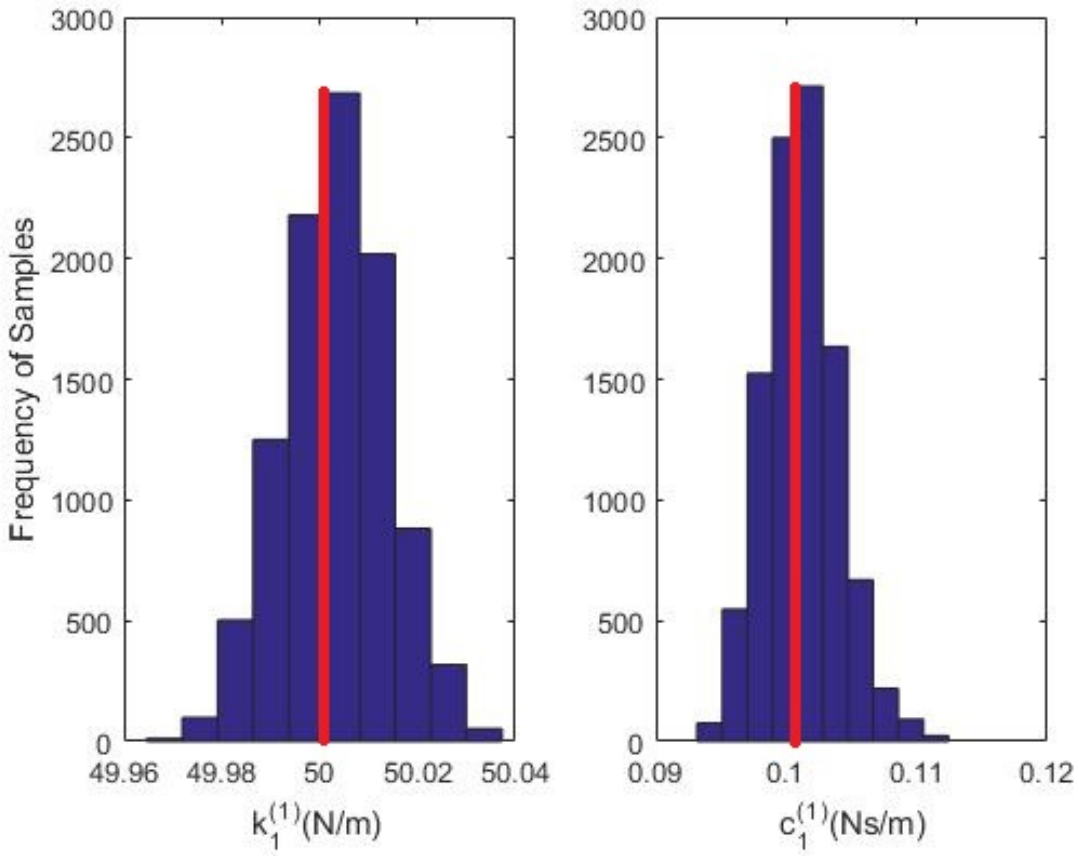

Figure 5.30: RJMCMC results for model $M^{(1)}$, histogram of samples(blue) and most frequent values(red)

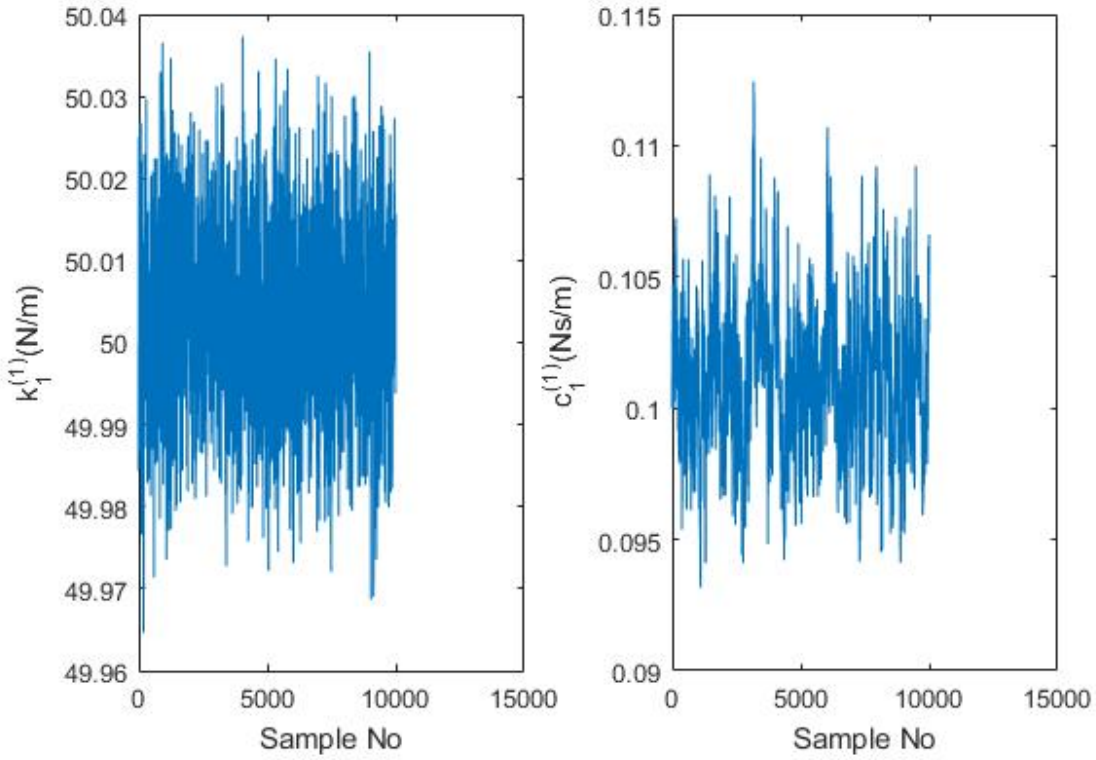

Figure 5.31: RJMCMC results for $M^{(1)}$, history of samples

Figure 5.32 and **Figure 5.33** show the results of applying the RJMCMC algorithm by using the nonlinear training data produced. The model chosen is the second model and the parameters of the second model are identified with mean values of $k_1^{(2)} = 50N/m$, $c_1^{(2)} = 0.1Ns/m$ and $k_2^{(2)} = 1000N/m$.

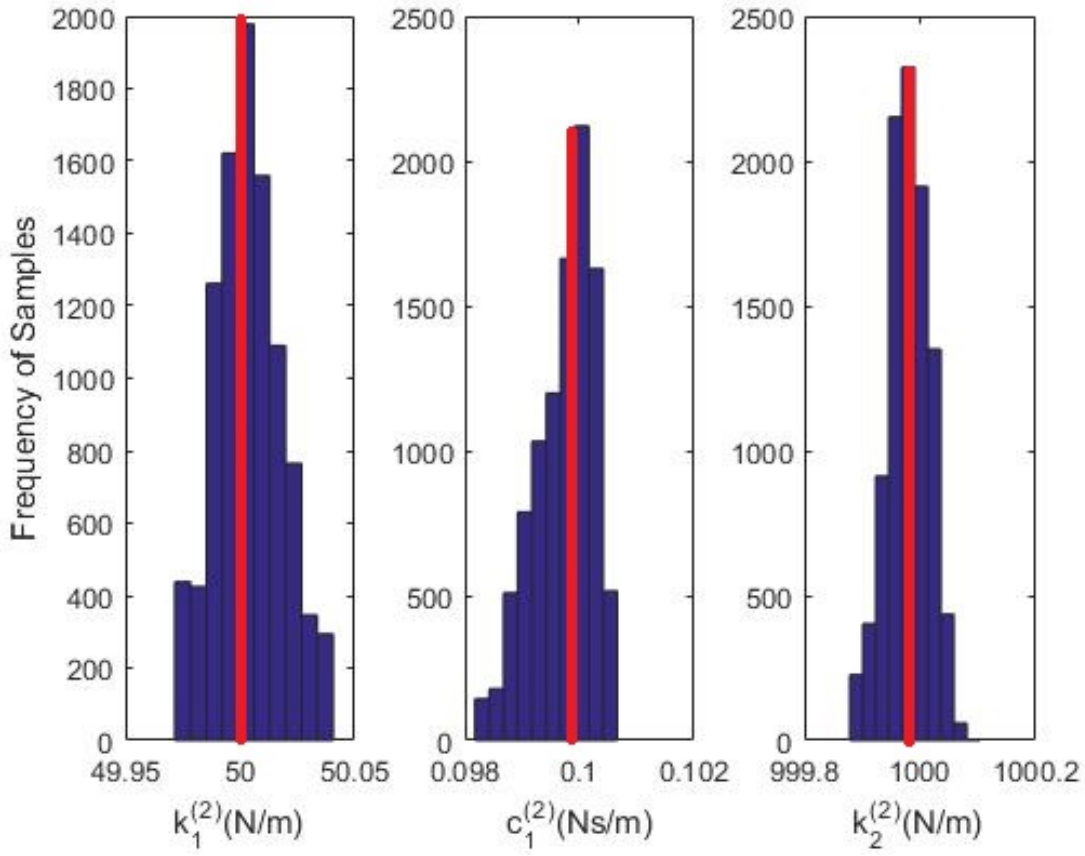

Figure 5.32: RJMCMC results for $M^{(2)}$, histogram of parameters(blue) and most frequent values(red)

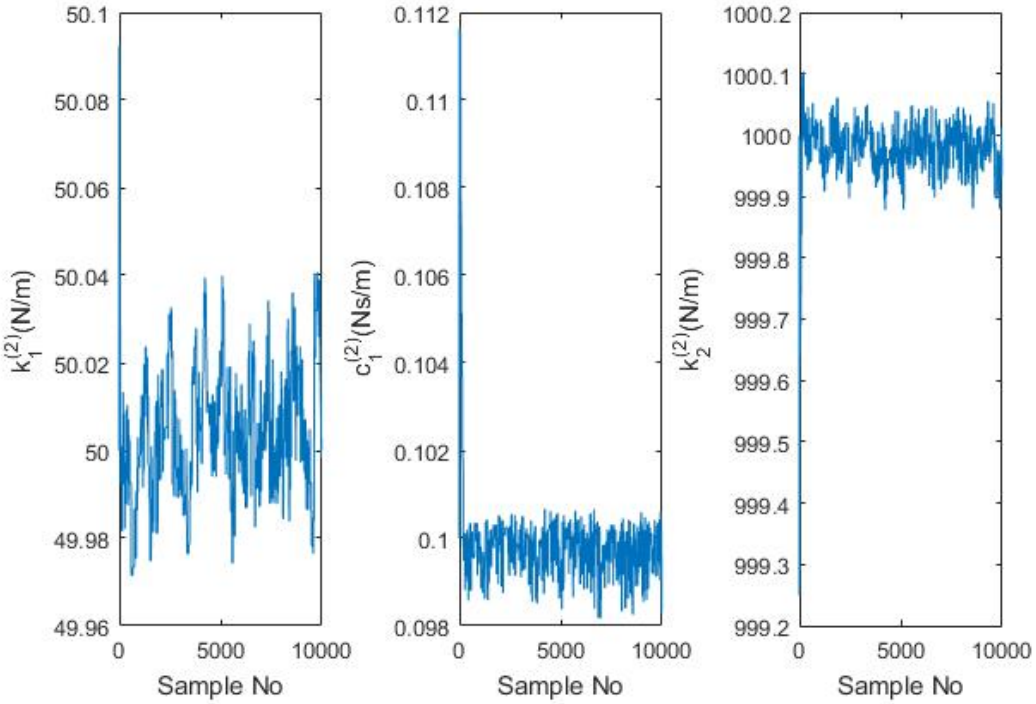

Figure 5.33: RJMCMC results for $M^{(2)}$, history of samples

Once the RJMCMC algorithm proved to work as expected in extreme cases, a study was conducted on the sensitivity of the RJMCMC method on the strength of the nonlinear element. In this scenario, the extra stiffness, $k_2^{(2)}$ was varied.

The stiffness $k_2^{(2)}$ was given values of $0N/m$, $5N/m$, $10N/m$, $25N/m$ and $50N/m$. Training data was generated with each considered value of $k_2^{(2)}$ using the nonlinear model. The obtained displacements were compared to the generated linear training data in order to observe the differences.

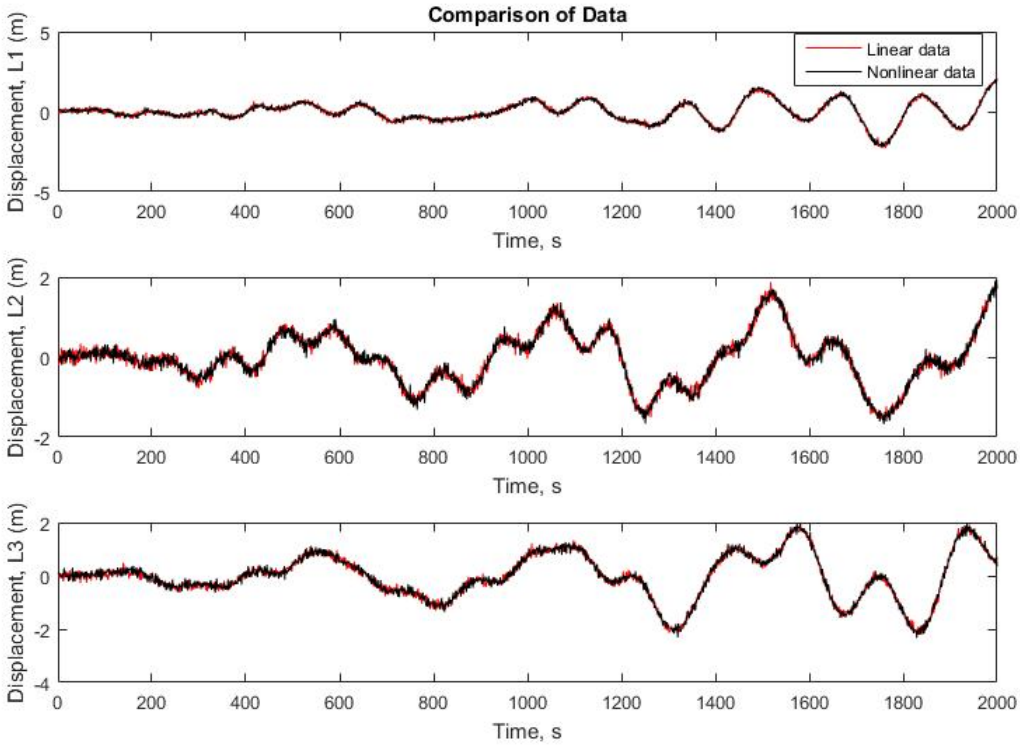

Figure 5.34: Linear training data VS Nonlinear training data with $k_2^{(2)} = 0N/m$

For the first test, the RJMCMC algorithm approximated $M^{(1)}$ to be the appropriate model to represent the data over $M^{(2)}$, by a ratio 10001 to 1. This was to be expected by looking at the data in **Figure 5.34**. The mean squared error is a measure used to quantify the goodness of fit between sets of data, in this case between the two sets of training data. The training data sets are following the same patterns and the calculated mean squared errors are $MSE = \{0.0199, 0.0201, 0.0200\}$, for each of the three levels respectively. The errors were calculated according to the following equation:

$$MSE = \frac{\sum (D - \bar{D})^2}{N} \tag{5.7}$$

where D is the original training data, $\bar{D}$ is the data obtained by changing the extra stiffness parameter and N is the number of data points.

A good fit is when the MSE estimator is closer to 0 but anything less than 0.1 is

considered a good enough fit [80].

The second value considered was $k_2^{(2)} = 5N/m$. The comparison of the training data is plotted in **Figure 5.35**.

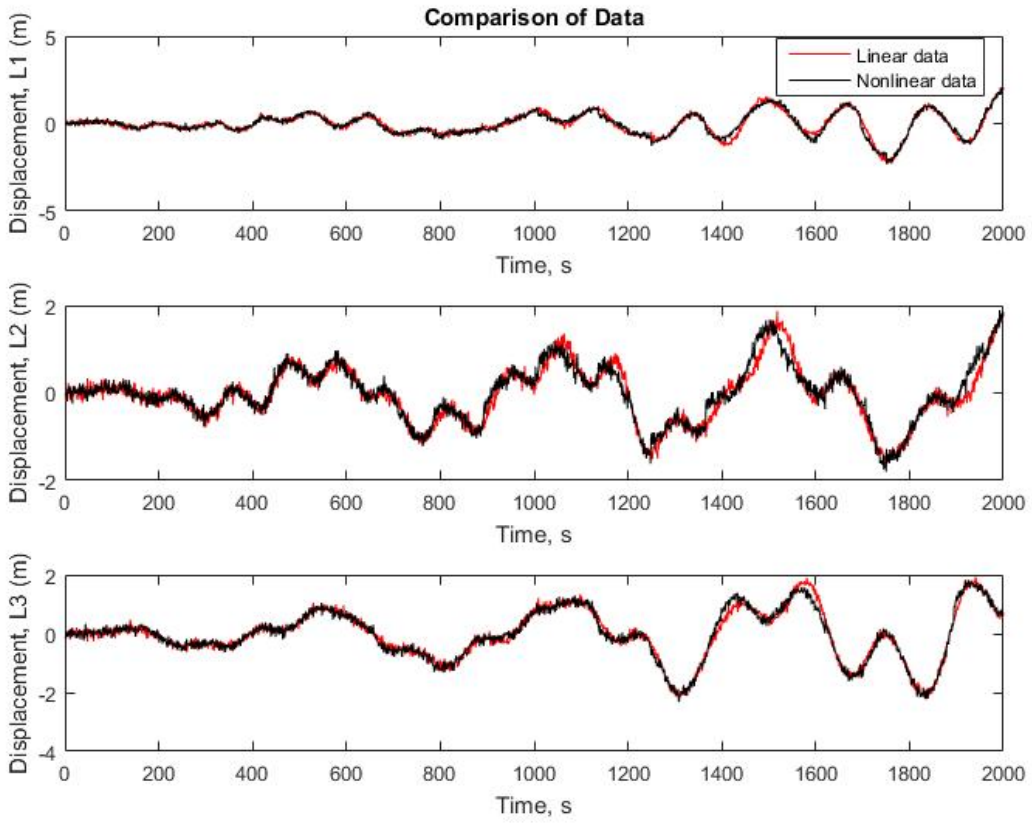

Figure 5.35: Linear training data VS Nonlinear training data with $k_2^{(2)} = 5N/m$

A slight difference between the patterns of the training data is already observed around data point 800 in the displacement of the second level. What is of interest is if the RJMCMC algorithm will pick that up. The algorithm was applied as before and this time $M^{(2)}$ was preferred over $M^{(1)}$. This suggests that the RJMCMC algorithm is sensitive to these changes. At the same time, it needs to be kept in mind that these are numerical studies and in real structures, the sources of uncertainty are numerous. The calculated mean squared errors are: $MSE = \{0.0354, 0.0459, 0.0313\}$.

For the case of $k_2^{(2)} = 10N/m$ the comparison of the data can be observed in **Figure 5.36**.

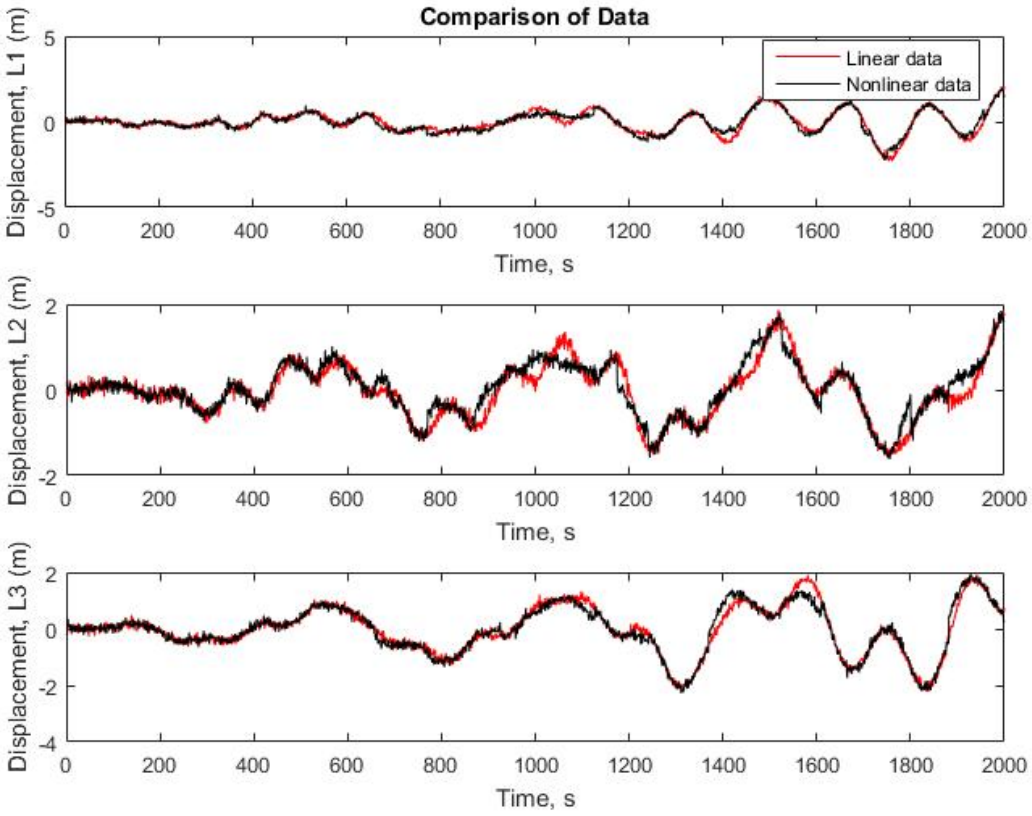

Figure 5.36: Linear training data VS Nonlinear training data with $k_2^{(2)} = 10N/m$

The RJMCMC algorithm was applied and $M^{(2)}$ was chosen as the best to represent the data over $M^{(1)}$ by a ratio of 9998 to 4. The calculated mean squared errors are: $MSE = \{0.0504, 0.0658, 0.0474\}$.

Figure 5.37 shows the comparison plots of the linear training data and the nonlinear training data with $k_2^{(2)} = 25N/m$.

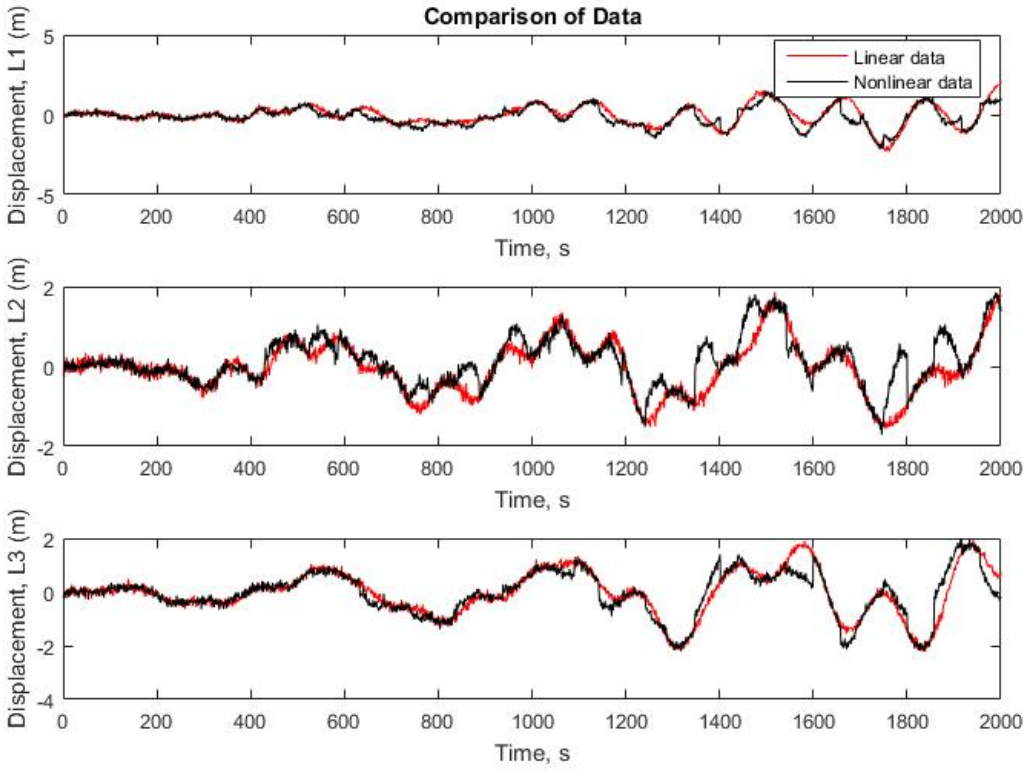

Figure 5.37: Linear training data VS Nonlinear training data with $k_2^{(2)} = 25N/m$

The results from the RJMCMC algorithm showed that $M^{(2)}$ is still more appropriate for the training data used than $M^{(1)}$ by a ratio of 9999 to 3. The calculated mean squared errors between the linear training data and the nonlinear training data are: $MSE = \{0.1235, 0.1673, 0.1252\}$.

The last nonlinear data set generated was with $k_2^{(2)} = 50N/m$. The comparison plots between the linear training data and the nonlinear training data with $k_2^{(2)} = 50N/m$ are shown in **Figure 5.38**.

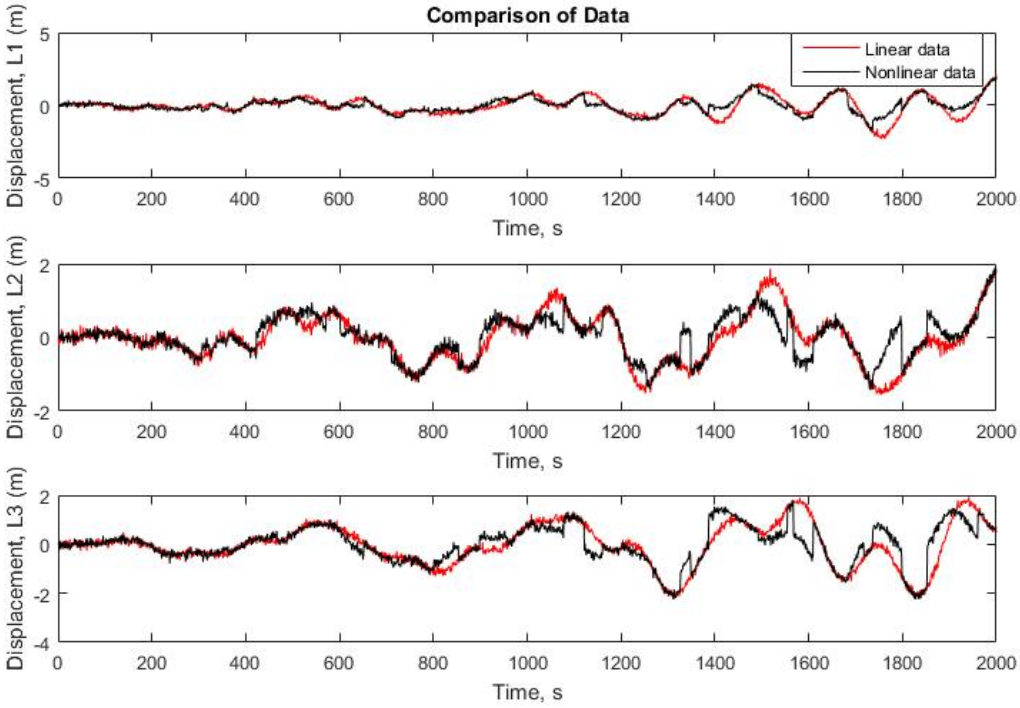

Figure 5.38: Linear training data VS Nonlinear training data with $k_2^{(2)} = 50N/m$

As expected, the RJMCMC chose $M^{(2)}$ over $M^{(1)}$ as the model to best represent the nonlinear training data by a ratio of 9997 to 5. For this case, $MSE = \{0.1382, 0.1481, 0.2256\}$.

The present study showed that the RJMCMC algorithm is sensitive to the effects of the nonlinearity as long as they can be observed. The method chose the linear model to best represent the nonlinear data only in the case when the extra stiffness was 0.

5.4 RJMCMC on a Numerical Case Study in Structural Health Monitoring

As a last example in numerical case studies, the author decided to apply the RJMCMC algorithm to a very simple and straightforward scenario in Structural Health Monitoring(SHM), in order to emphasize even more how relevant the RJMCMC

algorithm is in structural dynamics.

This study investigates the application of the RJMCMC algorithm to the problem of locating, characterizing and assessing structural damage. A great deal of interest has been paid to treating the damage identification task as one of model updating with both deterministic and non-deterministic updating methods having been extensively investigated. Among these frameworks, Bayesian Inference has found particular success when applied in the fields of both System Identification and Structural Health Monitoring (SHM). The advantages of using Bayesian Inference in SHM applications include its reliance upon prior knowledge, enabling beliefs regarding the state of the structure, the bounds on its physical parameters and the likely damage scenarios encountered to be included in the estimation process. A further advantage of adopting a Bayesian approach is that outcomes are presented as the posterior probability distributions of the variables of interest. As the geometries of these posterior probability distributions are generally complex it is typical for Markov Chain Monte Carlo (MCMC) samplers to be employed in their estimation. However, while MCMC methods offer an effective tool for estimation of damage parameters, they do not directly lend themselves to the model selection problem. Having the ability to consider multiple explanatory models simultaneously would offer new opportunities for damage classification and identification of multi-site damage. Reversible Jump Markov Chain Monte Carlo (RJMCMC) offers a potentially powerful extension of standard MCMC methods. The first strength of the RJMCMC method is that it addresses both parameter estimation and model selection simultaneously. As such, RJMCMC returns not only the parameter estimates for each model but also provides a probabilistic indication of which model is most consistent with the data. The second strength of RJMCMC is that it is sufficiently general that the models being compared may contain different numbers of parameters. These characteristics open up a number of potential approaches to the SHM problem. The damage classification task may be approached by generating a set of alternative damage models, each of which is a function of its own set of damage parameters (e.g. depth of crack, location, orientation). In this case the method would provide the evidence for each class of damage alongside the parameter estimates. Alternatively, the set of models may comprise single and multiple damage scenarios. RJMCMC would return both the likely number of damage locations and estimates of their parameters simultaneously, addressing one of the more challenging tasks within model-based SHM.

So, in this case study, the RJMCMC algorithm is used to identify the likely number

of damage locations as well as to assess the extent of damage.

A set of objectives were considered:

- Apply the RJMCMC on a simulated damage detection and identification numerical case study;
- Show that the RJMCMC method can identify the location of damage as well as characterise it;
- Show that the RJMCMC method can select between different damage scenarios;

Now, going back to **Chapter 4**, the way the algorithm moves between the two models is by using a mapping. For the purpose of this study, the defined mapping function between models is $h : \Re^{(n)} \longrightarrow \Re^{(m)}$, where n is the dimension of the current model while m is the dimension of the proposed model. The form of the mapping between $M^{(1)}$ and $M^{(2)}$, which is a birth move, is presented in 5.8.

$$\begin{Bmatrix} D_{(1)}^{(2)} \\ R_{(1)}^{(2)} \\ D_{(2)}^{(2)} \\ R_{(2)}^{(2)} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ a \\ b \end{Bmatrix} + \begin{Bmatrix} D_{(1)}^{(1)} \\ R_{(1)}^{(1)} \\ \lambda \\ \epsilon \end{Bmatrix} \tag{5.8}$$

where $D_{(1)}^{(2)}$ represents the first damage position in $M^{(2)}$, $R_{(1)}^{(2)}$ represents the first stiffness reduction in $M^{(2)}$, $D_{(2)}^{(2)}$ represents the second damage position in $M^{(2)}$, $R_{(2)}^{(2)}$ represents the second stiffness reduction in $M^{(2)}$. $D_{(1)}^{(1)}$ represents the first damage position in $M^{(1)}$ and $R_{(1)}^{(1)}$ represents the first stiffness reduction in $M^{(1)}$. λ and ϵ are numbers chosen by the user to construct the mapping.

5.4.1 Results

For the purpose of the numerical case study emulators were developed for damage scenarios of a simply supported beam.

The emulators were developed based on the results of a finite element (FE) model of a simply supported beam implemented using MATFEM, a MATLAB-based FE package. A $1m$ long aluminium beam was adopted with cross sectional dimensions of

height by base, Youngs modulus $E = 70GPa$, density $\rho = 2700kgm^{-3}$ and Poissons ratio $= 0.3$. The beam was meshed as 100 Timoshenko beam elements. Damage was introduced as a simple reduction in the Youngs modulus value of individual elements at the damage location. This was implemented by pre-multiplying the Youngs modulus value of the element by a damage parameter in the range [0,1], where 1 represents damage at the element and 0 would represent complete removal. It is acknowledged that this is a very simplistic means of introducing damage into the model but nonetheless served the needs of the numerical case study. For the purposes of training the emulator the damage location was taken as the midpoint of the affected element i.e. damage at element 1 was taken to occur at $0.005m$ from the end of the beam.

The developed FE model was executed using a full-factorial design of experiments (DOE) for both the single site and two site damage scenarios. For the single site scenario the damage multipliers considered were [0.4,0.6,0.8,1.0] at elements [5:10:95], resulting in 40 model executions. For the two-site case the same set of damage multipliers were considered at all combinations of elements [5:10:95] (location 1) and [6:10:96] (location 2) resulting in 1600 model executions. Choosing different elements for damage locations 1 and 2 avoided the question of how to deal with 'overlapping' damage at the same location. It should be noted that in more efficient DOE methods including Latin hypercube sampling are available for practical applications where the computational cost of executing the model may become significant.

The emulator used for illustration was a simple linear interpolant between the design points. It is acknowledged that a linear interpolant is likely to give a less accurate representation of the true behaviour between the design points than a more sophisticated alternative. However, this choice means the emulator cannot introduce any confounding effects thus allowing the efficacy of the RJMCMC approach to be judged in isolation. Two sets of emulators were trained, the first taking two parameters and the second four parameters as inputs with each returning natural frequency predictions for the first 20 modes as outputs.

In this particular case study the RJMCMC algorithm was applied on two emulators. The first emulator was simulating a single site damage in a fixed beam while the second emulator was simulating a two damage sites in the same simply supported beam. In the first scenario the damage was a 30% reduction in stiffness at 0.09m along the beam, see **Figure 5.39**. Second scenario included a 30% reduction in

stiffness at 0.09m along the beam together with a 60% reduction at 0.57m along the beam, see **Figure 5.40**. The natural frequencies at the first 20 modes were used as data for the RJMCMC algorithm.

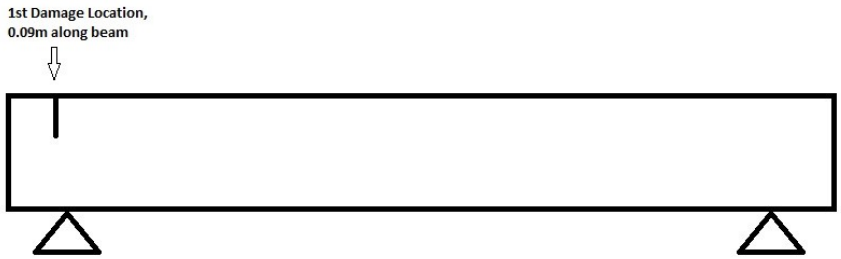

Figure 5.39: First damage location along the beam

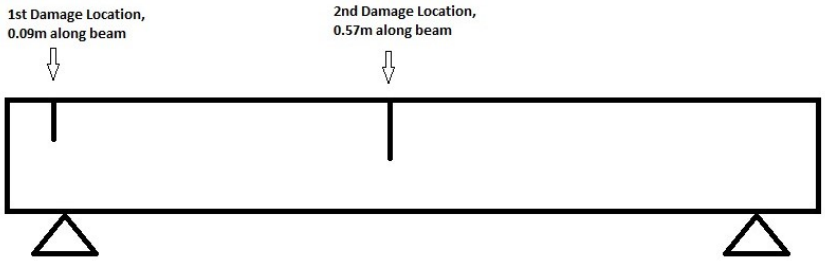

Figure 5.40: First and second damage location along the beam

When using the data for the single site damage scenario, the RJMCMC algorithm has successfully identified the first model to be the correct one as seen in **Figure 5.41**. In **Figure 5.42** one can see the results of parameter estimation for $M^{(1)}$. The RJMCMC method was employed using 10000 iterations. The proposal widths were both adjusted to 0.001. The algorithm spent most time in the first case scenario as 8052 samples were from $M^{(1)}$ and only 1950 from $M^{(2)}$. A burn in of 500 samples

was used.

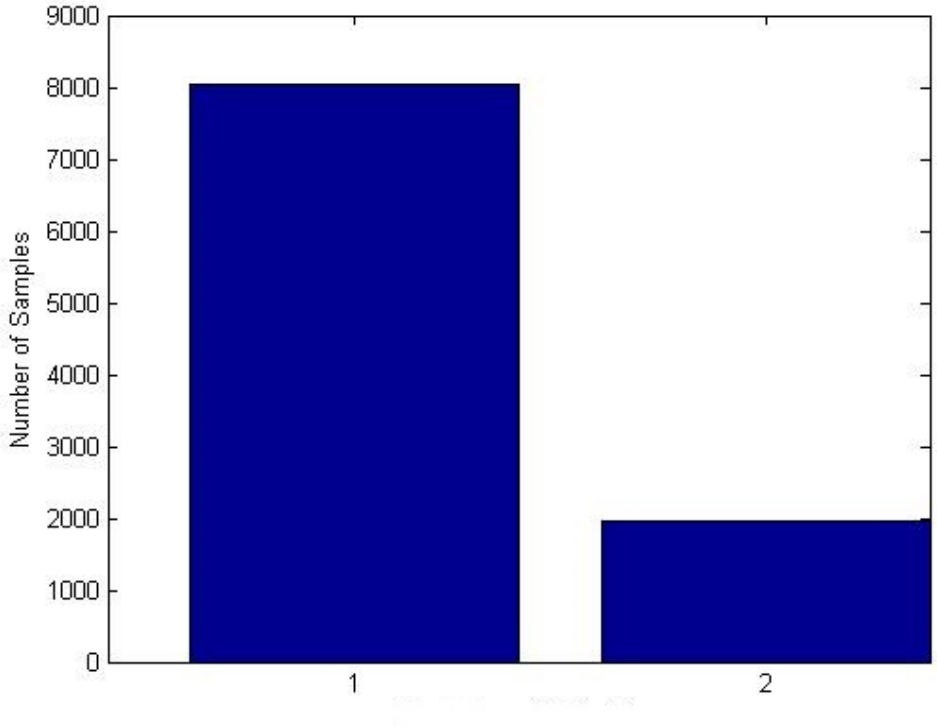

Figure 5.41: Acceptance frequency of Model 1 VS Model 2

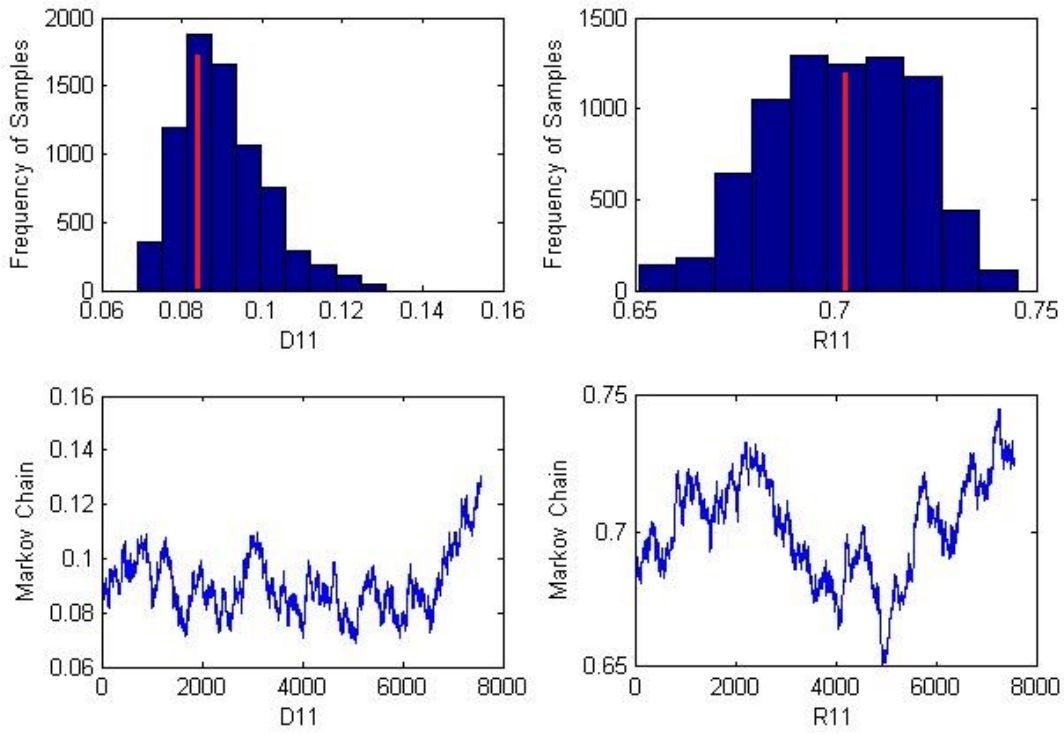

Figure 5.42: Histogram of samples(blue), most frequent values(red) and History of Samples

When using the data for the two site damage scenario, the RJMCMC algorithm has successfully identified the second model to be the correct one as seen in **Figure 5.43**. In **Figure 5.44** one can see the results of parameter estimation for $M^{(2)}$. The RJMCMC method was employed using 20000 iterations, due to the fact that the algorithm had to explore a bigger space. The proposal widths were adjusted to 0.0001, 0.0001 for the parameters of the first damage and 0.001 and 0.001 for the parameters of the second damage. The algorithm spent most time in the first case scenario as 19549 samples were from $M^{(2)}$ and only 453 from $M^{(1)}$. A burn in of

500 samples was used again.

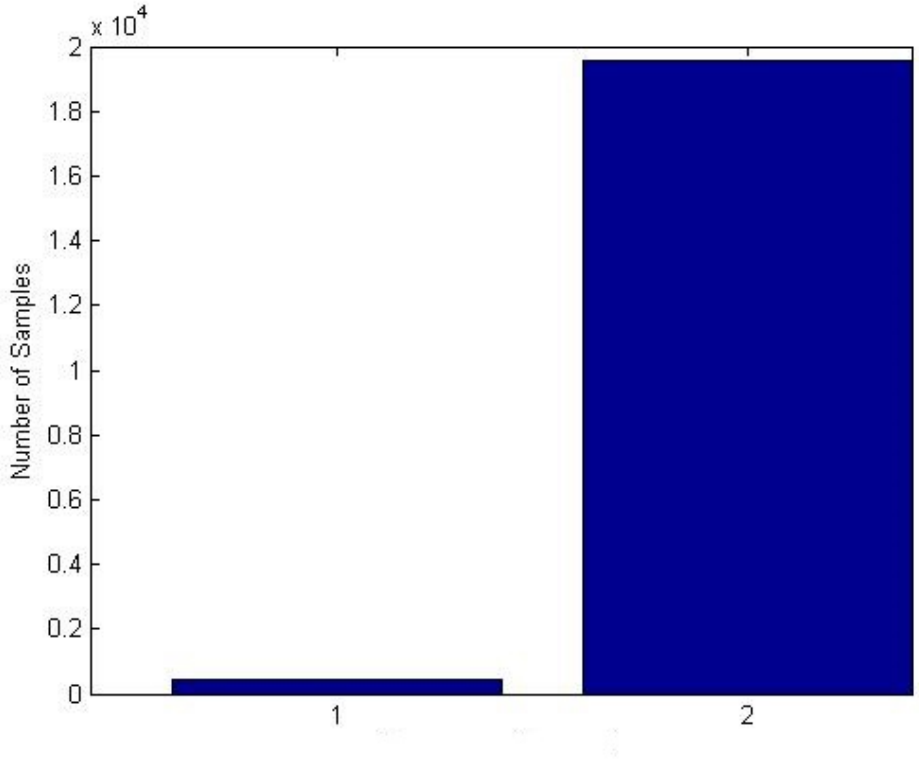

Figure 5.43: Acceptance frequency of Model 1 VS Model 2

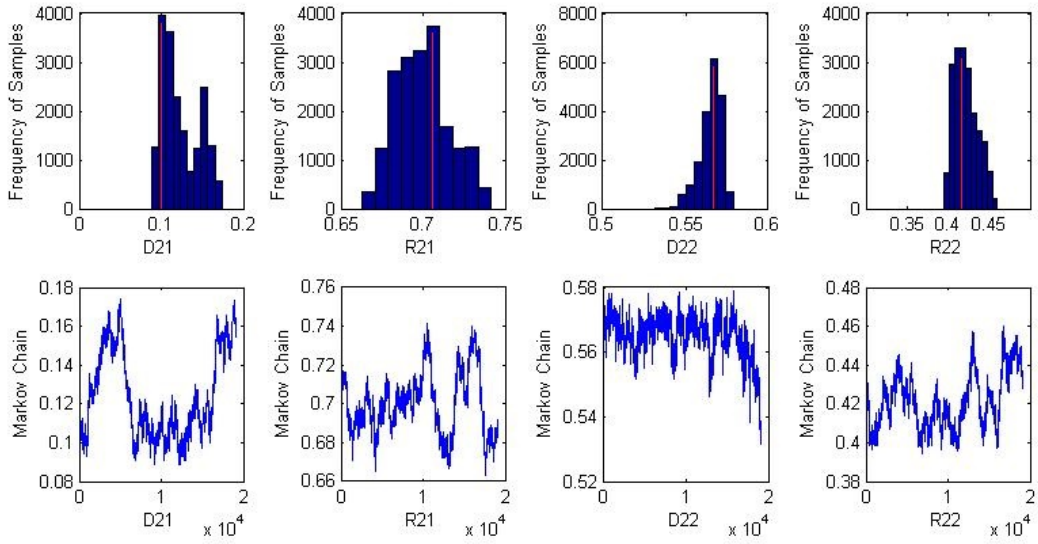

Figure 5.44: Histogram of samples(blue), most frequent values(red) and History of Samples

The results show that the RJMCMC is efficient also when used on emulators in the scope of damage detection.

The purpose of this initial work done on damage detection using RJMCMC algorithm was for it to continue on an experimental example of a beam and later on to be used for doing damage detection on the vertical beams of the rig built for the experimental work done in **Chapter 6**. The time constraint did not allow for further delving onto this path and as such, the numerical case study in SHM in presented as part of work that should be done in the future.

5.5 Summary

This chapter covered applications of the RJMCMC algorithm on numerical case study in nonlinear system identification and structural health monitoring in order to demonstrate that the method is a powerful tool in structural dynamics.

The first case study involved a SDOF system for which two models were presented - a linear model and a nonlinear model of Duffing type. The RJMCMC algorithm successfully achieved system identification of the system, in both linear and nonlinear situations. The algorithm was firstly applied on a set of linear simulated data

for which two competing models were developed, a SDOF model and a nonlinear SDOF model with a cubic stiffness nonlinearity. The RJMCMC methods correctly identified the linear model as the model that provided the best fit for the considered data, while also identifying its parameters. The second step was to apply the RJMCMC algorithm on a set of nonlinear data, simulated by using the second model that has a cubic stiffness term. The RJMCMC method appropriately identified the second model as the one that provided the best fit for the considered data, while estimating its parameters. A short study on the influence of data points on the RJMCMC algorithm was conducted. For this particular analysis, the two sets of data, linear and nonlinear, were compared and the RJMCMC method was applied in stages, by increasing the number of data points available. The algorithm proved to have a high sensitivity to the amount of data it works with, such that, at the first data point when the data sets were not following the same trend, the RJMCMC method could correctly distinguish between the linear and the nonlinear model. The last consideration for the SDOF system was to change the excitation forcing from random excitation to periodic excitation for which the RJMCMC algorithm was applied.

The second case study was mainly an extension of the first case study in the sense that this time one had to deal with a MDOF system, more precisely a 3-DOFs structure. As before, the RJMCMC algorithm proved successful in doing system identification for each training data set. For the second numerical study the degrees of freedom was increased and also the number of parameters to be identified was increased. Data sets were generated from a linear 3-DOF model and then the RJMCMC algorithm was applied. The number of iterations had to be increased to allow the Markov chains to settle. Nevertheless, the RJMCMC method correctly identified that the linear model was more appropriate for the linear data set. It followed that when the nonlinear data was generated and the RJMCMC algorithm was applied to it, the method accurately chose the second model, with the cubic stiffness, as the one that provides the best fit to the nonlinear data set. An observation was made that as the degrees of freedom increase, so does the algorithms computational time required to run, an expected consequence. A study was then conducted on how the RJMCMC algorithm behaves when the cubic nonlinearity is moved between masses so that results were presented when the nonlinearity was placed between the base floor and m_1, then m_1 and m_2 and respectively m_2 and m_3.

The third case study covered a 3-DOF structure for which two models were con-

sidered, a linear model and a bilinear stiffness model. Training data was generated by using a linear model and respectively the nonlinear model by including a bilinear stiffness element between m_1 and m_2. The RJMCMC was employed in order to identify the structure. When the algorithm had the nonlinear data available, it correctly chose the nonlinear model to best fit the data and in the same way, it chose the linear model to best fit the linear model. The parameters were estimated accordingly. The identification process was followed by a study on how the nonlinear element influences the choice of the RJMCMC algorithm when it comes down to the model selection aspect.

The fourth case study was used to prove the power of the RJMCMC algorithm in the area of structural dynamics. With that in mind, a study was conducted for an emulated simply-supported beam with the conclusion that the algorithm was effective in identifying the location of damage as well as the extent of it. The first model of damage had a single site damage for which the first 20 natural frequencies were considered, while the second model contained two sites for damage for which the same amount of natural frequencies were taken into account. The RJMCMC algorithm correctly identified the locations of the damage in the emulated beam, as well as the number of damages and the amount of damage by estimating the reduction in stiffness at damage location. There was no need for a baseline of an undamaged emulated beam for the method to provide estimates of the parameters of interest.

Chapter 6 will cover experimental case studies, where the RJMCMC algorithm was brought closer to validation by being used on experimental data.

Chapter 6

RJMCMC on Experimental Studies

Chapter 6 covers the application of the RJMCMC algorithm on experimental case studies. This work follows up the numerical case studies as a validation of the method on real systems. Two case studies are considered for which an experimental rig was built. For reasons explained within the chapter, the rig had to be redesigned before the second case study due to the challenge of finding a truly linear baseline structure. The first case study takes into consideration the variance of noise for a MDOF real structure and its effect on the identification process. Two different models are proposed for the same structure and the RJMCMC algorithm is employed in order to select the appropriate model and estimate its parameters. The second case study delves into the system identification of a real structure that is made to behave linearly and nonlinearly. Sets of data are collected and the RJMCMC algorithm is applied in order to identify the more appropriate model and provide estimates of the parameters.

6.1 Introduction

In **Chapter 5** one could see a few successful applications of the RJMCMC algorithm on numerical case studies. The main theme of the thesis was kept, as such the first two case studies presented were on nonlinear system identification. The last case

study was in SHM in order to provide a further illustration of the capabilities of the algorithm.

One could expect though, as the attention turns toward real systems, things become more challenging. This chapter covers how the challenges of applying this powerful algorithm, the RJMCMC, on experimental data were addressed and the full detailed results of that work.

The overall objectives of this chapter are:

- Show the application of the RJMCMC algorithm on experimental data;
- Discuss arising challenges when dealing with experiments;
- Provide analysis of results and concluding remarks;

Section 6.2 covers the application of the RJMCMC method on a 3-DOF modelled structure for which two competing models are considered. The difference in the models are in the variance of the noise at each level. **Section 6.3** discusses a case study in which the RJMCMC algorithm is applied on a 2-DOF modelled structure for which linear and nonlinear regimes are considered. The last section of this chapter will cover summary, general discussions and conclusions.

6.2 RJMCMC on an Experimental Case Study in System Identification - MDOF system

The rig used in the first experimental case study is presented below. At the end of this section, reasons for replacing it are given.

The initial aim was to start with a linear MDOF system and to apply the RJMCMC algorithm to it. The aims and objectives for this case study were set as follows:

- Gather experimental data from structure;
- Construct competing models for the studied structure based on noise variance;
- Identify parameters by employing the MH algorithm;
- Apply the RJMCMC on the two competing models to select the model to best

represent the data;

- Analyse results;

Experimental time series data was gathered for the proposed structure at a single excitation level, at four different outputs. The simplified model of the structure was chosen as a base excitation 3DOF system for which the parameters were identified using the MH algorithm. Two competing models were considered: for the first model, the noise variance was assumed the same for all of the three levels; the second model had individual noise variances for each level. Noise is a cause of uncertainty in the process of system identification. Considering that the RJMCMC algorithm is applied in a Bayesian framework, penalising model complexity is one of the advantages. The expectation of this study was to see how much influence does the noise variance modelling has on the overall modelling of the structure. It was thought that a stronger case could be made for the first model to be considered more appropriate, give its reduced complexity. However, the actual results allowed more room for discussion.

Section 6.2.1 covers details on the structure used and the procedure of gathering the data. This is followed in **Section 6.2.2** by a simplified modelling of the structure in equations of motion. In **Section 6.2.3** and **Section 6.2.5** the MH algorithm is applied in order to estimate the model parameters and the results are presented. **Section 6.2.7** covers the application of the RJMCMC algorithm on the two competing models. The case study is concluded with analysis of results and discussion.

6.2.1 The Experimental Rig

This case study focuses on the RJMCMC algorithm applied on experimental time data from an MDOF structure, in order to do SID of a linear dynamical system.

The structure used throughout is pictured in **Figure 6.1**.

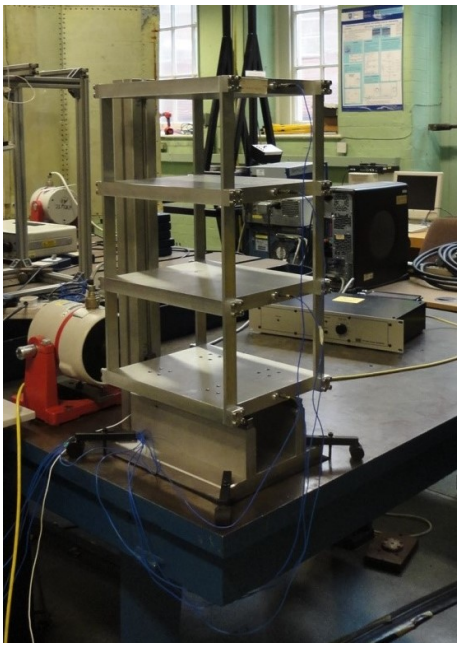

Figure 6.1: Experimental 4-DOFs Rig

The structure has 4 levels, with the lower level being considered the base(see **Figure 6.2**). Each main plate/level has an approximated weighted mass of 5.2 kg and approximate dimensions of 35 x 25.5 x 2.5cm (LxWxH). Each upright beam connecting the 4 levels has an approximated weighted mass of 238 g and approximated dimensions 55.5 x 2.5 x 0.6cm. Each square block used to connect the main plates and the upright beam has an approximated weighted mass of 18 g and approximated dimensions 2.5 x 2.5 x 1.3cm. For each block, 4 bolts were used with an approximated weighted mass of 10 g and of Viraj A2-70 grade. The structure was mounted on a rail system which was bolted in and then clamped onto a testing table. In order to introduce the excitation into the structure, a shaker with a force transducer was connected to the base. The output of the structure was recorded as acceleration, using 4 accelerometers positioned, as shown in **Figure 6.1** right hand side, at the middle of each main plate/level.

The data was gathered using a SCADAS-3 interface connected to a PC running LMS Test.Lab software. 93184 data points were recorded at a sampling frequency of $f_s = 1024$ Hz. Random excitation was used throughout and recordings were taken only under one excitation instance.

6.2.2 Model

The following linear model was proposed:

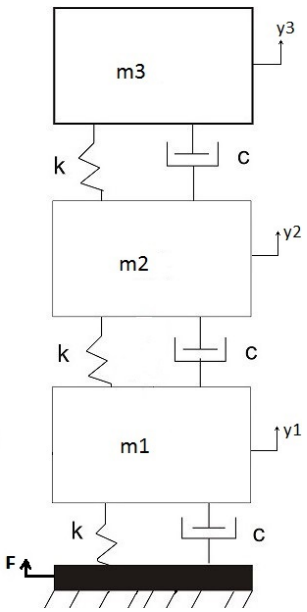

Figure 6.2: 3-DOFs Linear system

for which the equations of motion were constructed as follows:

$$\begin{cases} m_1\ddot{z}_1 + 2c\dot{z}_1 - c\dot{z}_2 + 2kz_1 - kz_2 = -m_1\ddot{y}_b \\ m_2\ddot{z}_2 - c\dot{z}_1 + 2c\dot{z}_2 - c\dot{z}_3 - kz_1 + 2kz_2 - kz_3 = -m_2\ddot{y}_b \\ m_3\ddot{z}_3 - c\dot{z}_2 + c\dot{z}_3 - kz_2 + kz_3 = -m_3\ddot{y}_b \end{cases} \quad (6.1)$$

where m_1, m_2, m_3 are the masses of levels 2, 3 and 4 (counting from the lowest level to the highest level), $\ddot{z}_1$, $\ddot{z}_2$ and $\ddot{z}_3$ are the relative accelerations of each corresponding level with respect to the base (for example, $\ddot{z}_1 = \ddot{y}_1 - \ddot{y}_b$), $\dot{z}_1$, $\dot{z}_2$ and $\dot{z}_3$ are the relative velocities of each corresponding level with respect to the base, z_1, z_2 and z_3 are the relative displacements of each corresponding level with respect to the base and finally $\ddot{y}_b$ is the acceleration of the base. The damping was kept the same for each level such that $c_1 = c_2 = c_3 = c$, as well as the stiffness, $k_1 = k_2 = k_3 = k$. This is likely to not be the case as the physical values will differ but the effect of it can be potentially small so that a reduced calibration parameter set is considered.

The model presented above is an obvious simplification of the true structure.

6.2.3 Metropolis-Hastings - on the experimental MDOF system, one noise variance

The MH algorithm was applied on the 3-DOF system in order to identify the parameters: stiffness, k and damping coefficient, c; and also to estimate the standard deviation of the noise, σ. For simplicity, the parameters were kept the same throughout the model. The likelihood function was chosen such that the noise is considered to be Gaussian with variance σ^2.

Accounting for the transient response, the data used was of 5000 samples, data points between 20000 and 25000, with a burn-in of 500 samples. The proposal probabilities used were Gaussian with widths tabulated below. The number of iterations was chosen to be 10000.

Parameters	$k(N/m)$	$c(Ns/m)$	σ
Proposal widths	100	0.1	0.01

Table 6.1: Proposal widths for the parameters to be identified, $M^{(1)}$

6.2.4 Results

Figure 6.3 and **Figure 6.4** show the graphical results, histogram of samples of parameters and the Markov chains of the samples.

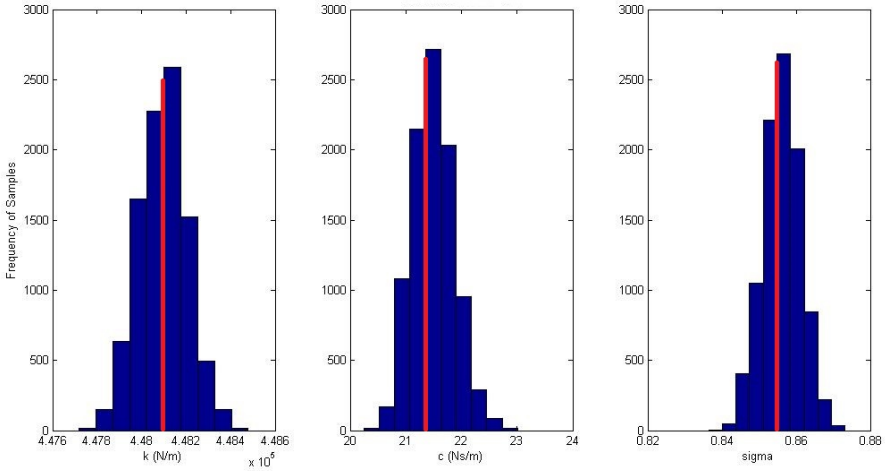

Figure 6.3: Histogram of Samples for $M^{(1)}$(blue), most frequent values(red)

Figure 6.3 shows the frequency of the samples for all three parameters of interest, stiffness, k, damping coefficient, c and standard deviation of noise, σ. It can be observed that the most frequent values are $k = 4.481 * 10^5 N/m$, $c = 21.5 Ns/m$ and $\sigma = 0.85$. In **Figure 6.4** one can check the stationary Markov chains of the samples for all three parameters. The plots show how the Markov chains efficiently explored the space of each parameter.

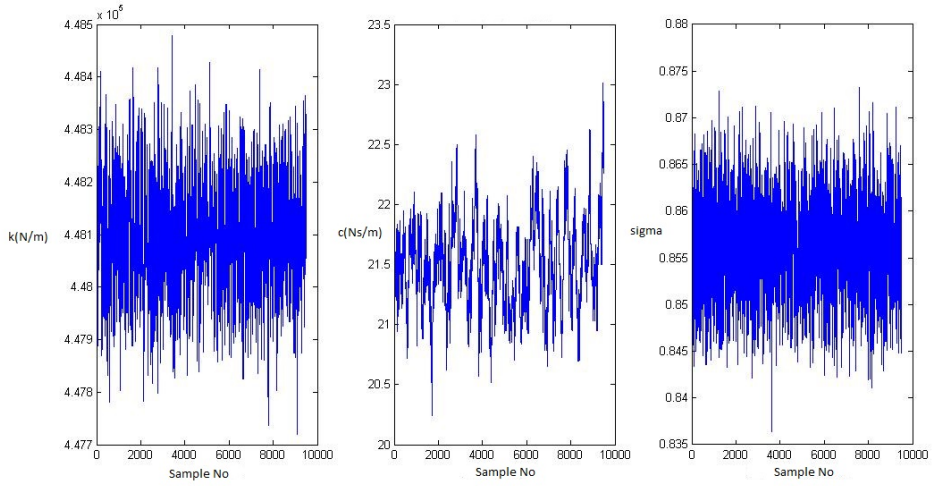

Figure 6.4: History of Samples for $M^{(1)}$

Model	$k(N/m)$	$c(Ns/m)$	σ
$M^{(1)}$	448100	21.5	0.85

Table 6.2: MH estimated parameter values for $M^{(1)}$

As a starting point, the mathematical model identified seems like a good fit. Noise is one of the reasons for uncertainty being present in the identification process. As discussed previously, the model was simplified to a single noise variance for the entire system. The next step is to include the results of parameter estimation for each level of the structure having its own noise standard deviation.

6.2.5 Metropolis-Hastings - on the experimental MDOF system, three noise variances

The MH algorithm was applied on the 3-DOF system in order to identify the parameters: stiffness, k and damping coefficient, c; and also to estimate the standard deviation of the noise at each level, $\sigma_1^{(2)}$, $\sigma_2^{(2)}$ and $\sigma_3^{(2)}$ for the second considered model, $M^{(2)}$. In order to avoid added complexity, the parameters were kept the same throughout $M^{(2)}$.

As before, accounting for the transient response, the data used was of 5000 samples, data points between 20000 and 25000. Again, the proposal probabilities used were Gaussian with widths presented in the table below. The number of iterations was chosen to be 10000.

Parameters	$k(N/m)$	$c(Ns/m)$	$\sigma_1^{(2)}$	$\sigma_2^{(2)}$	$\sigma_3^{(2)}$
Proposal widths	100	0.1	0.01	0.01	0.01

Table 6.3: Proposal widths for the parameters to be identified, $M^{(2)}$

6.2.6 Results

Figure 6.5 and **Figure 6.6** show the graphical results of applying the MH algorithm on the second proposed model, $M^{(2)}$.

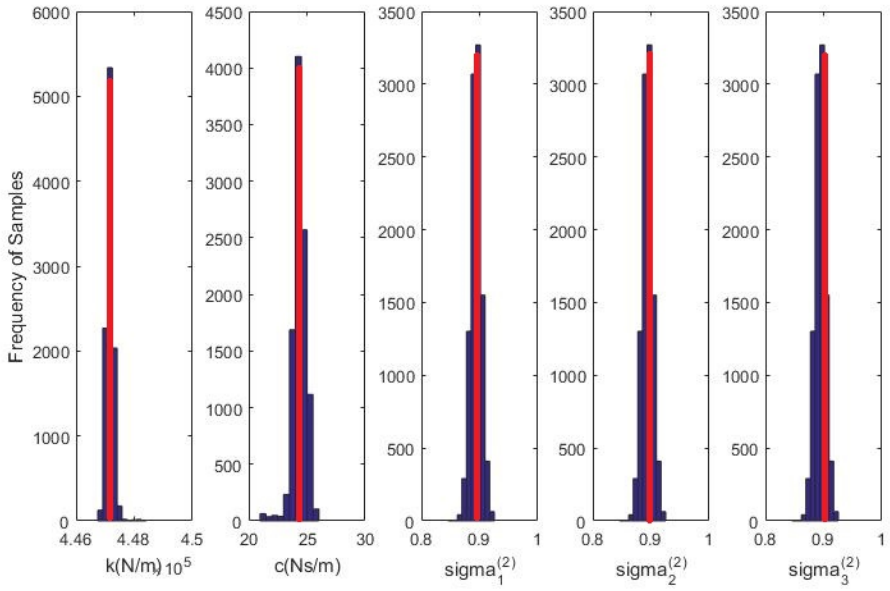

Figure 6.5: Histogram of Samples for $M^{(2)}$(blue), most frequent values(red)

Figure 6.5 shows the frequency of the samples for the two parameters of interest, stiffness, k, damping coefficient, c and for the standard deviations of noise at each floor, $\sigma_1^{(2)}$, $\sigma_2^{(2)}$ and $\sigma_3^{(2)}$. The most frequent values are $k = 4.470 * 10^5 N/m$, $c = 24 Ns/m$; $\sigma_1^{(2)} = 0.9$, $\sigma_2^{(2)} = 0.9$ and $\sigma_3^{(2)} = 0.9$. In **Figure 6.6** one can observe the Markov chains of the samples for all considered parameters. The plots show how the Markov chains efficiently explored the space of each parameter.

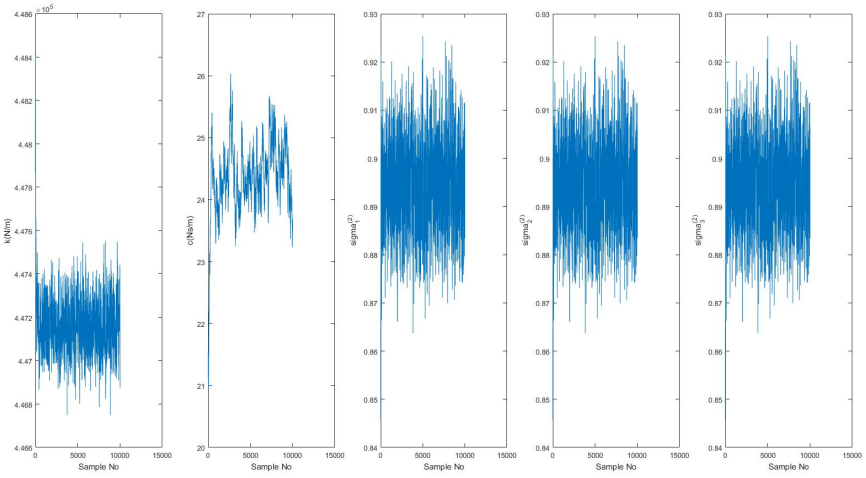

Figure 6.6: History of Samples for $M^{(2)}$

Model	$k(N/m)$	$c(Ns/m)$	$\sigma_1^{(2)}$	$\sigma_2^{(2)}$	$\sigma_3^{(2)}$
$M^{(2)}$	448100	21.5	0.9	0.9	0.9

Table 6.4: MH estimated parameter values for $M^{(2)}$

The MH algorithm was firstly applied on each considered model in order to get estimates for the parameters and to check that the models are set up in logical manner. The limitations of the MH algorithm create the need for the application of the RJMCMC method.

Due to the fact that the three-noise-standard-deviation model might be overly complex, the selection between the two models, $M^{(1)}$ and $M^{(2)}$ in a Bayesian context in considered;as this is not possible with the MH algorithm, the RJMCMC method will be used. Although the mathematical approximation of the real structure is simplified,the proposed approach gives robust results still.

Motivated by this aspect, the next section introduces the Reversible Jump Markov Chain Monte Carlo algorithm applied on the 3-DOF model with one variance of noise for the whole system and a 3-DOF model with three different variances of noise for each respective level of the system.

6.2.7 RJMCMC - on the experimental MDOF system

In this section, one is confronted with two models for the rig or the 3-DOF system. The first model has a noise variance for the entire system and the second model is defined with three different noise variances,$\sigma_1^{(2)}$, $\sigma_2^{(2)}$ and $\sigma_3^{(2)}$, one for each of the measured 3-DOF's respectively.

In order to approximate the best model that fits the data and also to estimate the parameters characterising the models, the RJMCMC algorithm was employed.

The RJMCMC algorithm was used to do model selection and parameter estimation on the 3-DOF experimental dynamical system. Equal prior probabilities, $P(M^{(1)}) = P(M^{(2)}) = 0.5$ were given to the first model, $M^{(1)}$, characterised by parameters $\boldsymbol{\theta} = \{k^{(1)}, c^{(1)}, \sigma^{(1)}, \lambda, \epsilon\}$ and the second model, $M^{(2)}$, characterised by the vector of parameters
$\boldsymbol{\theta'} = \{k^{(2)}, c^{(2)}, \sigma_1^{(2)}, \sigma_2^{(2)}, \sigma_3^{(2)}\}$.

A number of 25000 data points were necessary to account for the effects of the transient response. The RJMCMC employed 10000 iterations. The proposals used were of Gaussian type with widths tabulated below for both $M^{(1)}$ and $M^{(2)}$.

Parameters	$k(N/m)$	$c(Ns/m)$	σ	$\sigma_1^{(2)}$	$\sigma_2^{(2)}$	$\sigma_3^{(2)}$
Proposal widths	100	0.1	0.01	0.01	0.01	0.01

Table 6.5: Proposal widths for the parameters to be identified, $M^{(1)}$ and $M^{(2)}$

Starting values for the Markov chains were approximated through numerical calculations from the information gathered about the structure in the laboratory.

6.2.8 Results

Figures 6.7 and **6.8** illustrate the histograms and Markov chains for $M^{(2)}$, which was the model selected by the RJMCMC algorithm as being more appropriate to represent the studied data.

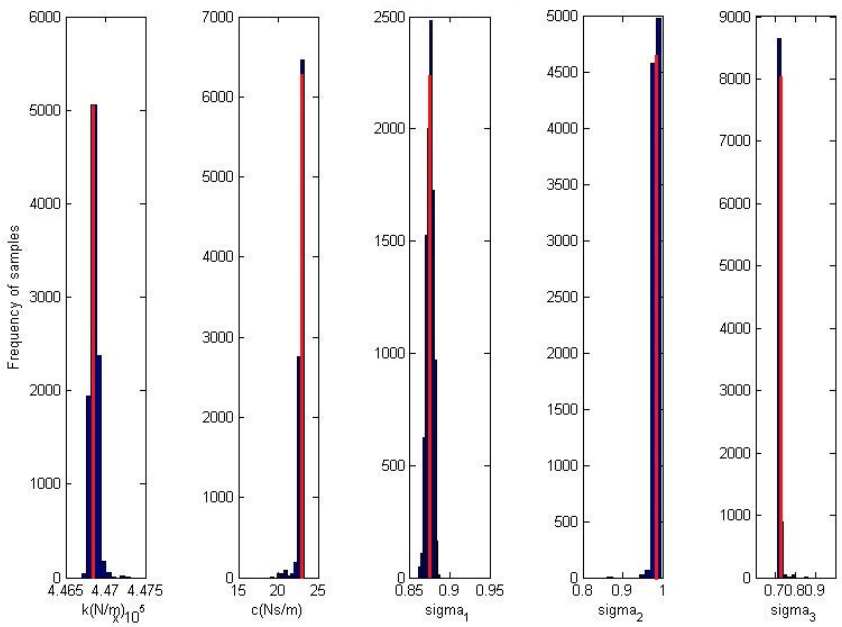

Figure 6.7: Histogram of Samples $M^{(2)}$(blue), most frequent values(red)

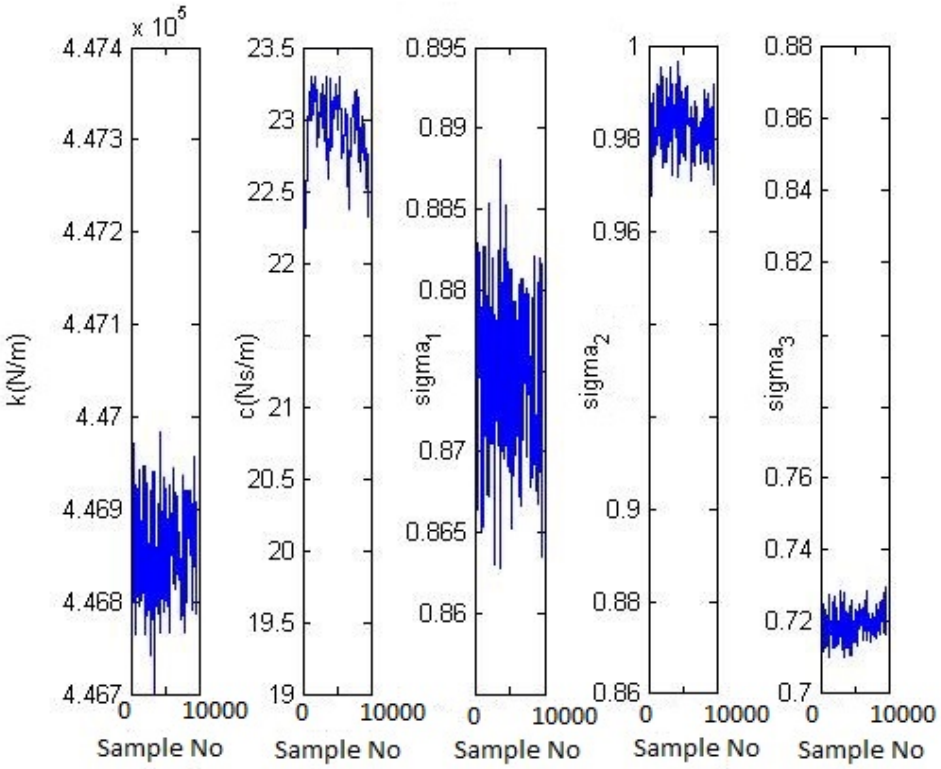

Figure 6.8: History of Samples $M^{(2)}$

Model	$k^2(N/m)$	$c^2(Ns/m)$	$\sigma_1^{(2)}$	$\sigma_2^{(2)}$	$\sigma_3^{(2)}$
$M^{(2)}$	468500	23.4	0.876	0.99	0.72

Table 6.6: RJMCMC parameter values for $M^{(2)}$

For comparison reasons, the results for $M^{(1)}$ are illustrated below, **Figures 6.9** and **6.10**. It is straightforward to see that the RJMCMC algorithm did not spend enough time in $M^{(1)}$ which resulted in non-stationary Markov chains, in the sense that little credibility could be given to the parameter estimates.

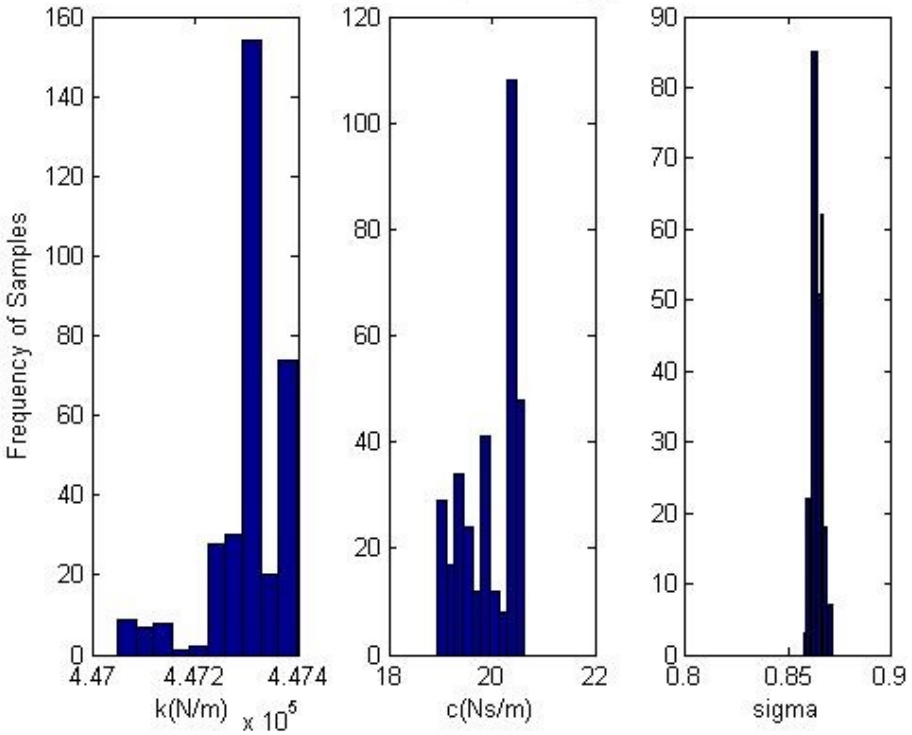

Figure 6.9: Histogram of Samples $M^{(1)}$(blue), most frequent values(red)

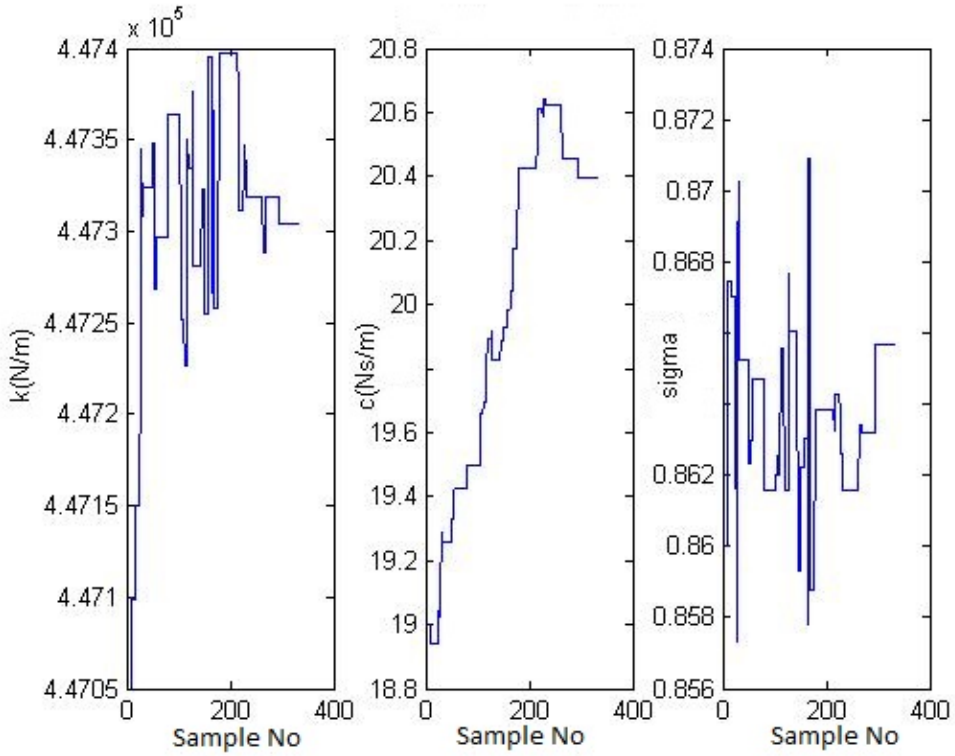

Figure 6.10: History of Samples $M^{(1)}$

The aim of the case study was to show the application of the RJMCMC algorithm in order to identify an experimental structure. For this purpose, two competing models were considered. The complexity of the linear model was increased by considering the effects of the noise variance:

- $M^{(1)}$ - identify parameters, k, c and include one noise variance for all levels, $\sigma_1^{(1)}$;
- $M^{(2)}$ - identify parameters, k, c and include noise variances for each individual level, $\sigma_1^{(2)}$,$\sigma_2^{(2)}$ and $\sigma_3^{(2)}$;

The RJMCMC algorithm chose $M^{(2)}$ to best fit the data set. It is difficult to know what to expect in a situation where there is no clear evidence for which model might be appropriate. The noise is a cause of uncertainty and its influence on the overall modelling of a structure has relevance. The change in noise variance is so subtle that it can be a reason why there are no conclusive results on which model should be preferred. In order to benchmark, to an extent, the estimates provided by RJMCMC, comparison was made to TMCMC. According to the TMCMC(Transitional Markov

Chain Monte Carlo) algorithm, $M^{(1)}$ is preferred which can be explained by the Bayesian influence in penalising complexity. On the other hand, the RJMCMC method chose $M^{(2)}$ which can be explained as a more accurate description and a potential better understanding of why the uncertainty brought in by noise is important.

6.2.9 Analysis of parameters estimates

In order to check the results of the RJMCMC algorithm, Monte Carlo simulations were run using samples of parameters of both models. Confidence bounds of 97% were used. The following figures, **Figures 6.11**, **6.12** and **6.13**, represent the bounds for each of the three levels,respectively. In order to visualise the results better, reduced ranges of the full data sets are provided due to the large number of data points. What this particular study revealed was that there existed little to no difference between the two models representing the data. However, in order to further explain the results obtained, another MCMC algorithm, the TMCMC [18], was applied on the two models, making use of the same set of data. Even though the approximations and estimations of the parameters were almost identical, the evidence terms of the posterior distributions for the two models suggested that $M^{(1)}$ rather than $M^{(2)}$ is the model preferred.

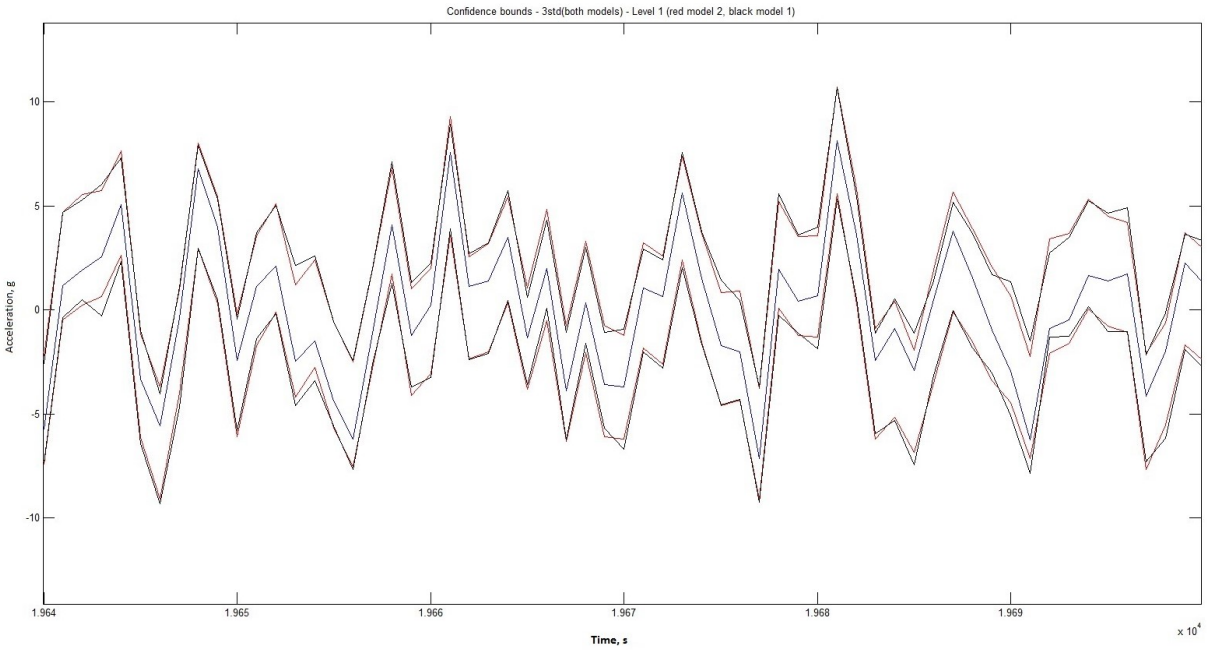

Figure 6.11: MC Simulation Results Level 1, red line $M^{(2)}$, black line $M^{(1)}$

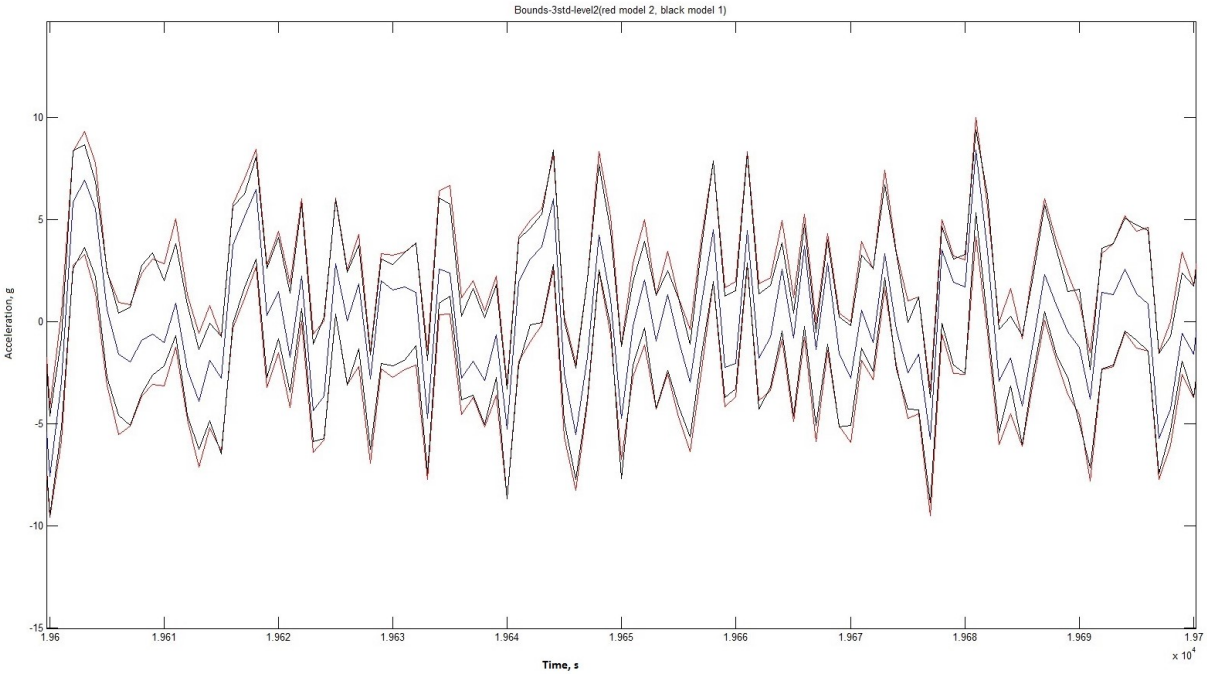

Figure 6.12: MC Simulation Results Level 2, red line $M^{(2)}$, black line $M^{(1)}$

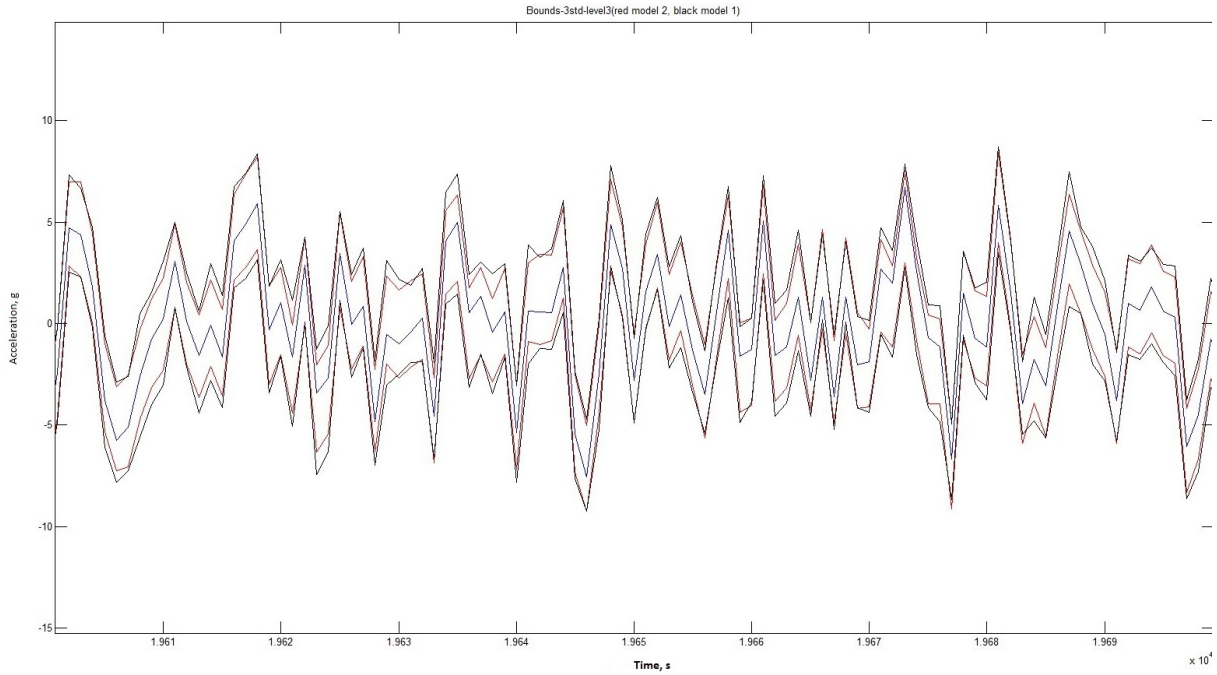

Figure 6.13: MC Simulation Results Level 3, red line $M^{(2)}$, black line $M^{(1)}$

6.2.10 Limitations of experimental rig

It was observed at the end of this study that the tests done on the experimental rig lacked a vigourous linearity proof. The author decided to take on the challenge and after several experiments it was discovered that the structure was not behaving linearly, not even under low excitations conditions (from 0.1 Volts to 0.9 Volts). The rig was displaying hardening effects on the first and second modes, and softening effects on the third mode. The rig was taken apart in order to analyse each component individually and what was found was that the railing system was causing the nonlinear effects in the results. The point of the bearing was to offer a frictionless sliding of the base but instead was causing too much friction to be ignored. For visualisation purposes, the following figures present some of the results of the analysis conducted.

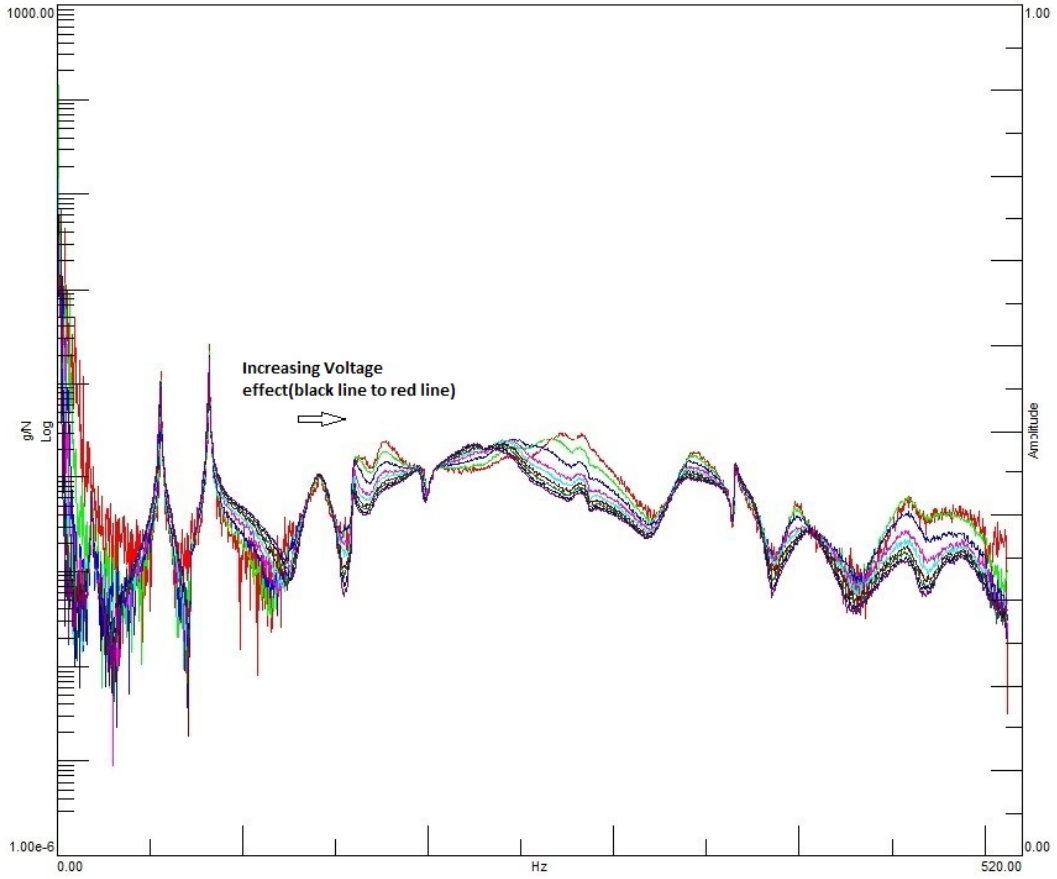

Figure 6.14: Initial rig design, FRF of 2^{nd} level of the new rig under low excitations, full range

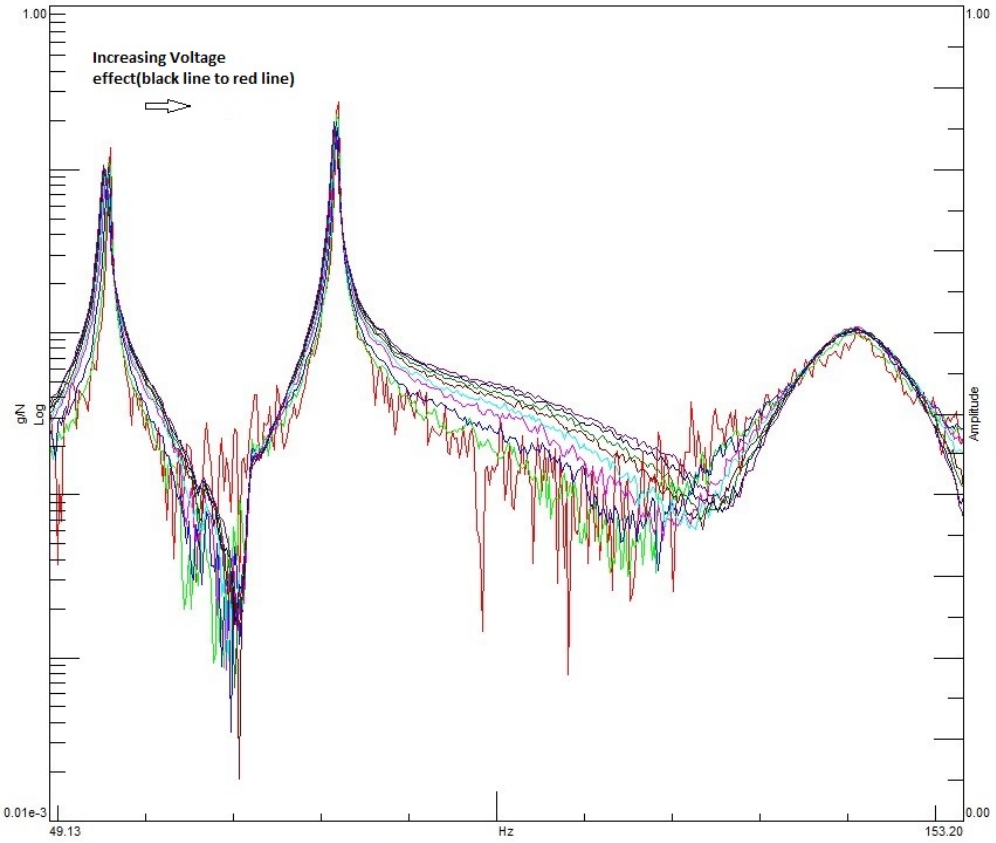

Figure 6.15: Initial rig design, FRF of 2^{nd} level of the new rig under low excitations, reduced range

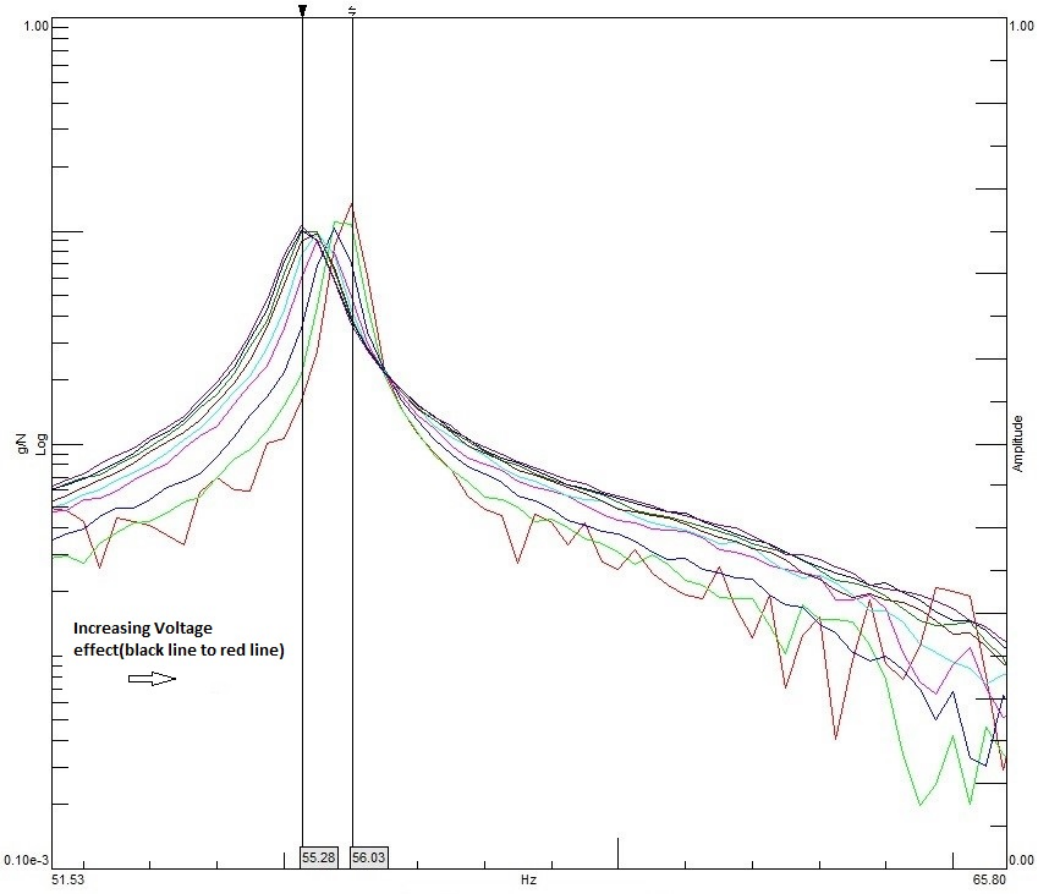

Figure 6.16: Initial rig design, FRF of 2^{nd} level of the rig, 1^{st} mode

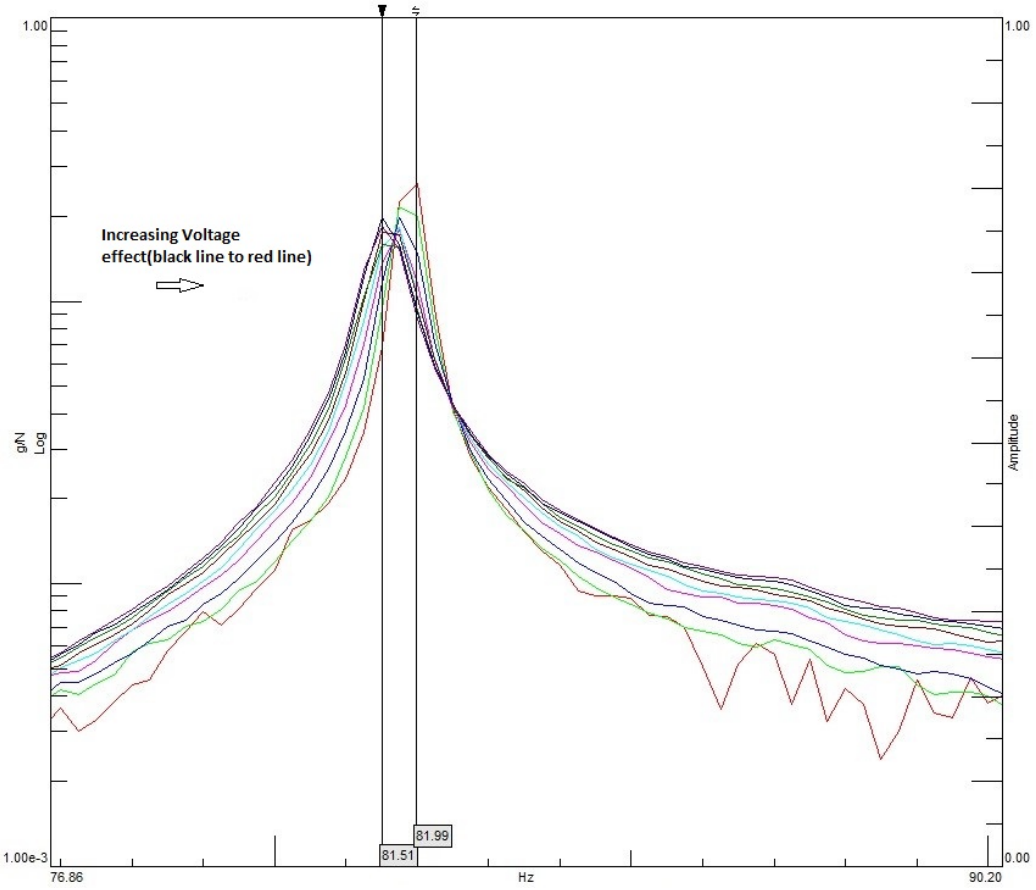

Figure 6.17: Initial rig design, FRF of 2^{nd} level of the rig, 2^{nd} mode

In **Figure 6.14** and **Figure 6.15** one can observe the FRF for the 2^{nd} level of the initial rig design. This is plotted in order of increasing excitation level, from 0.1 Volts to 0.9 Volts. For a linear system, under low excitations especially, there should be no shift in the natural frequencies. However, what can be seen in the zoomed in version is that there is a noticeable shift in the frequency responses. The next two figures, **Figure 6.16** and **Figure 6.17** give a look at the first two modes under the mentioned excitation range. Looking closely, almost a softening effect can be observed in the second mode (the first mode is lost in the noise). For the second mode the situation is more serious with a 1.34% decrease in natural frequency as the excitation level is slowly increased.

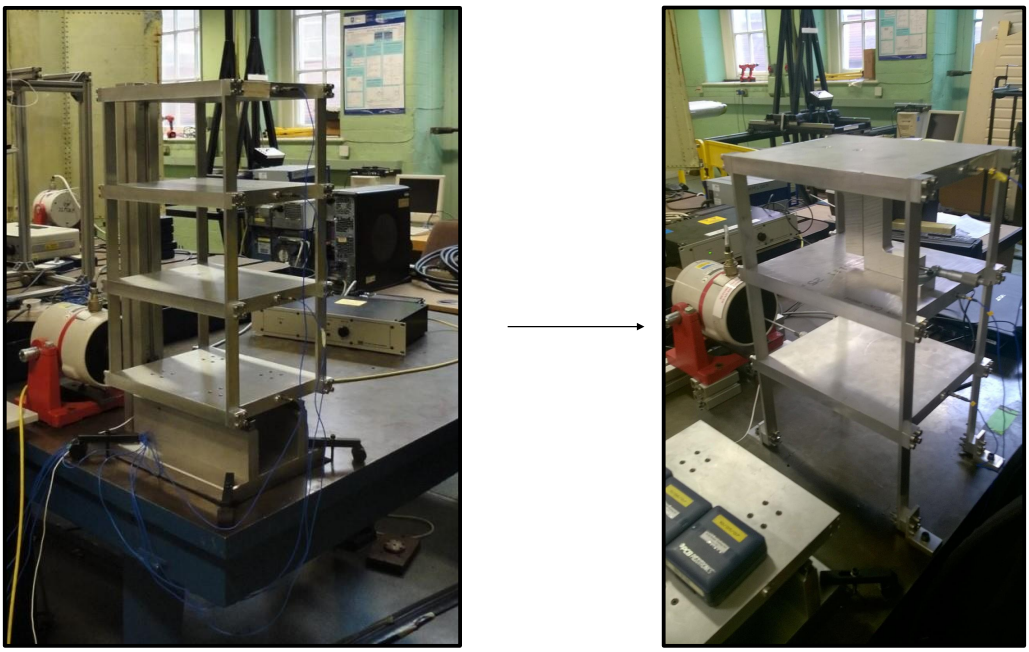

Figure 6.18: a) Initial Rig b) Improved Rig

Naturally, the next step included redesigning the rig. The same type of experimental work was conducted in order to check that this time, the improved rig was behaving linearly when expected to. The results are included for visualisation for the same case as for the initial designed rig.

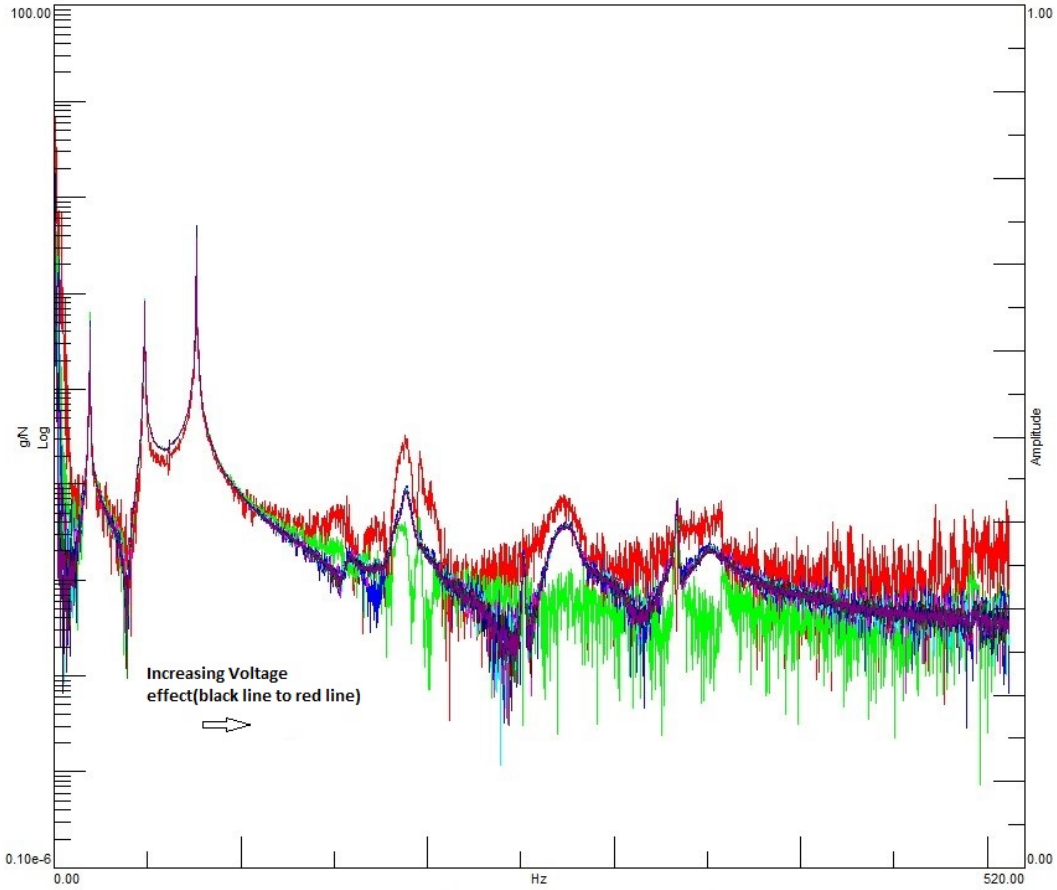

Figure 6.19: Improved rig design, FRF of 2^{nd} level of the new rig under low excitations, full range

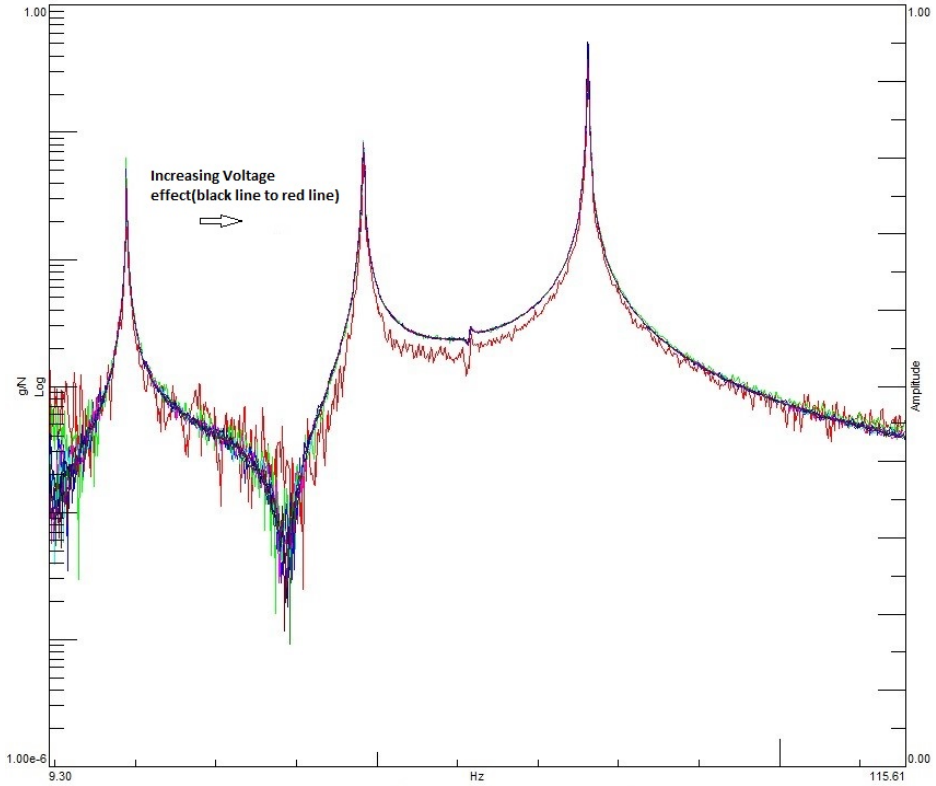

Figure 6.20: Improved rig design, FRF of 2^{nd} level of the new rig under low excitations, reduced range

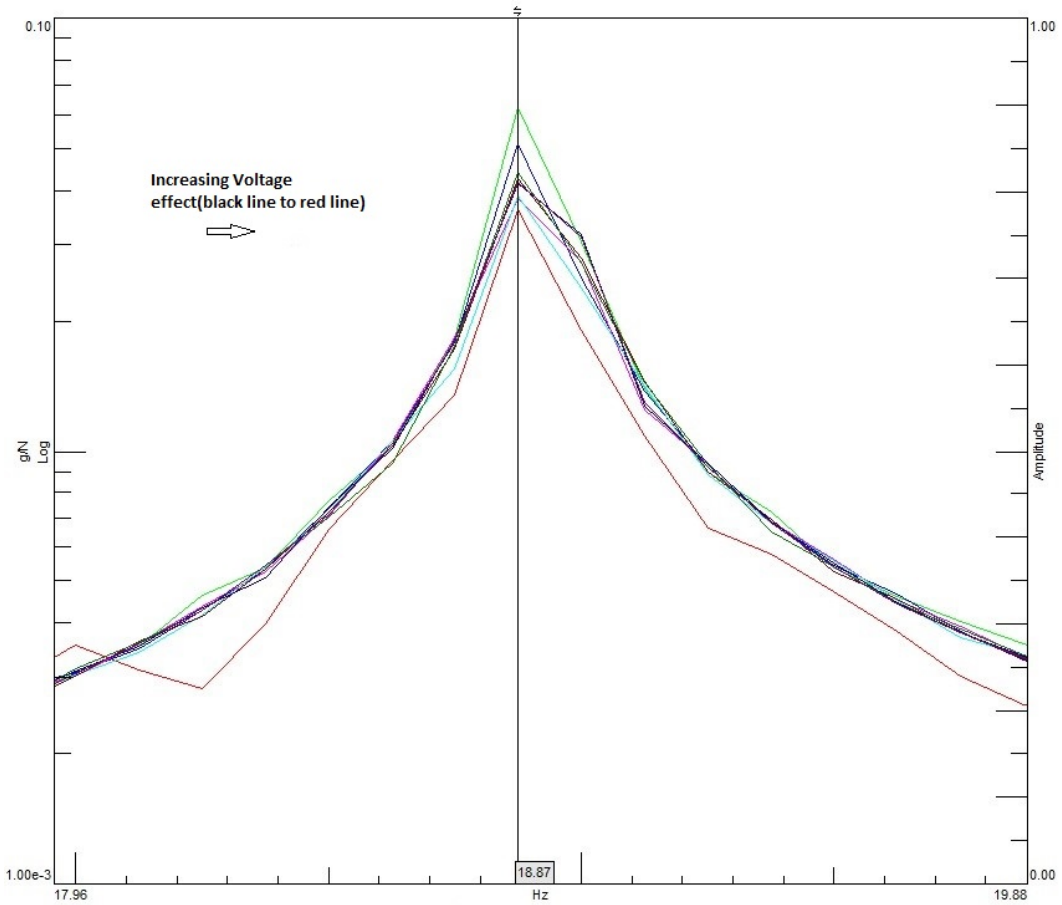

Figure 6.21: Improved rig design, FRF of 2^{nd} level of the rig, 1^{st} mode

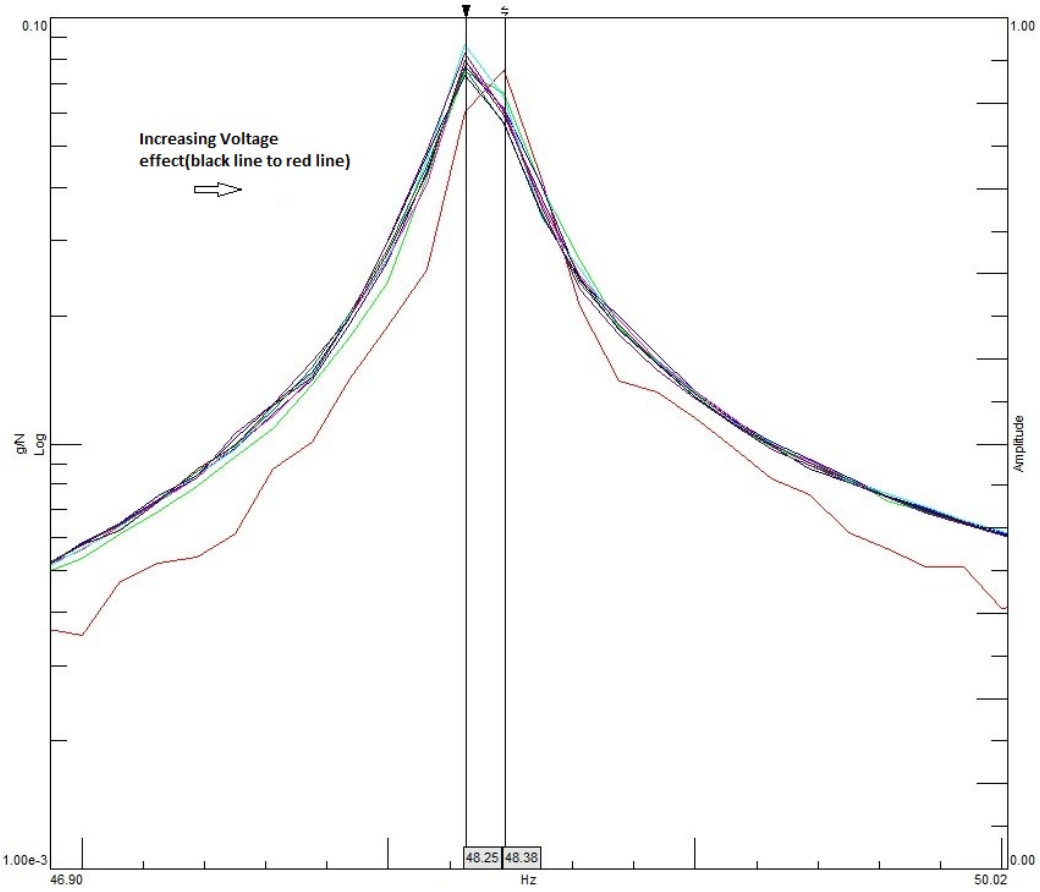

Figure 6.22: Improved rig design, FRF of 2^{nd} level of the rig, 2^{nd} mode

In **Figure 6.18** one can observe the FRF for the chosen 2^{nd} level of the improved rig design. This rig was designed to be bolted onto the testing table, eliminating the railing system element. This is plotted in order of increasing excitation level, from 0.1 Volts to 0.9 Volts as before. The improvement can be immediately observed. Looking closely, in **Figure 6.19** the first mode is constant throughout the increase in excitation level, while the second one sees a difference of 0.27% and the third mode had a difference error of only 0.16%.

After the rig was tested for linearity, a more rigourous approach was taken and the initial idea of identifying noise variance using the RJMCMC method was put aside. With this in mine, the improved rig was tested and a linear and a nonlinear model were built so that the parameters could be identified before applying the RJMCMC algorithm.

6.3 RJMCMC on an Experimental Case Study in Nonlinear System Identification - MDOF system

This present experimental case study is concerned with applying the RJMCMC algorithm on experimental time data from an MDOF structure, in order to do system identification of a nonlinear dynamical system.

The rig used in the second experimental case study is introduced. The aims and objectives for this case study were set as follows:

- Gather experimental data from structure;
- Construct competing models for the studied structure based on linear and nonlinear behaviour of bilinear stiffness type;
- Identify parameters for each considered model by employing the MH algorithm;
- Apply the RJMCMC on competing models under linear and nonlinear dynamic behaviour to do system identification;
- Analyse results;

Experimental time series data was gathered for the proposed structure at a single excitation level, at four different outputs. The simplified model of the structure was chosen as a base excitation, 2-DOF system for which the parameters were identified using the MH algorithm. Competing models were considered: the first model was characterised by two stiffness parameters for each DOF, $k_1^{(1)}$ and $k_2^{(1)}$ and two damping coefficients, $c_1^{(1)}$ and $c_2^{(1)}$ for each DOF, respectively ; two nonlinear models were considered: the first characterised by an extra stiffness parameter, compared to $M^{(1)}$, in order to account for the bilinear stiffness behaviour and the second having two extra parameters, the added stiffness and the distance parameter d, which represents the distance between the two blocks.Once again, a Bayesian framework has the advantage of penalising model complexity.

The expectation of this study was to see how the RJMCMC algorithm performs in doing system identification on a real structure. It was expected for the RJMCMC

method to choose the linear model, $M^{(1)}$ when linear behaviour was observed and the nonlinear model, $M^{(2)}$ when nonlinear behaviour was present, according to the data set used.

Section 6.3.1 covers details on the structure used and the procedure of gathering the data. This is followed in **Section 6.3.4** by a simplified mathematical modelling of the structure. In **Section 6.3.5** and **Section 6.3.6** the MH algorithm is applied in order to estimate the model parameters and the results are presented. **Section 6.3.11** covers the application of the RJMCMC algorithm on the two competing models. The case study is concluded with analysis of results and discussion.

6.3.1 Experimental structure

The structure used is a 3-DOF rig pictured in **Figure 6.22**.

Figure 6.23: Experimental 3-DOFs Rig with nonlinear element in place

The re-designed structure has 3 levels, with the lower level being considered the base as before. Each main plate/level has a weighted mass of 6.1082 kg and dimensions 35

x 25.5 x 2.5cm (LxWxh). Each upright beam connecting the 3 levels has a weighted mass of 238 g and dimensions 55.5 x 2.5 x 0.6cm. Each square block used to connect the main plates and the upright beam has a weighted mass of 18 g and dimensions 2.5 x 2.5 x 1.3cm. For each block, 4 bolts were used with a weighted mass of 10 g and of Viraj A2-70 grade. The structure was bolted in and then clamped onto a testing table. In order to introduce the excitation into the structure, a shaker with a force transducer was connected to the base. The output of the structure was recorded as acceleration, using 3 accelerometers positioned, as in **Figure 6.22** right hand side(blue cables), at the middle of each main plate/level.

The data was gathered once again using a SCADAS-3 interface connected to a PC running LMS Test.Lab software. 186368 data points were recorded at a sampling frequency of 1024 Hz in the form of time domain acceleration readings. Random excitation was used in this particular case study as well.

6.3.2 Test sequence

Three different sets of data were acquired in order to conduct system identification on this case study. The first set of data was gathered,186368 data points, under random excitation, with the two blocks apart from each other,i.e. with no contact. A sampling frequency of $f_s = 1024$ Hz was decided upon.

The second set of data was gathered under random excitation, with the two blocks in full contact. The same number of data points was gathered, under the same sampling conditions.

The third set of data was gather under random excitation, with the two blocks having an offset of $d = 0.1$ mm. This allowed for the blocks to be in and out of contact.

All three different cases were modelled differently, according to the dynamics displayed:

- For the first set of data, the linear base acceleration 2-DOF model,$M^{(1)}$, was

used:

$$\begin{cases} m_1\ddot{z}_1 + (c_1+c_2)\dot{z}_1 - c_2\dot{z}_2 + (k_1+k_2)z_1 - k_2 z_2 = -m_1\ddot{y}_b \\ m_2\ddot{z}_2 - c_2\dot{z}_1 + c_2\dot{z}_2 - k_2 z_1 + k_2 z_2 = -m_2\ddot{y}_b \end{cases} \tag{6.2}$$

- For the second set of data, the linear base acceleration 2-DOF model,$M'^{(1)}$, was used:

$$\begin{cases} m_1\ddot{z}_1 + (c_1+c_2)\dot{z}_1 - c_2\dot{z}_2 + (k_1+k_2')z_1 - k_2' z_2 = -m_1\ddot{y}_b \\ m_2\ddot{z}_2 - c_2\dot{z}_1 + c_2\dot{z}_2 - k_2' z_1 + k_2' z_2 = -m_2\ddot{y}_b \end{cases} \tag{6.3}$$

  where $k_2' = k_2 + k_L$ is the new stiffness that represents the addition of k_2 and the additional stiffness inflicted by the contacting blocks, k_L.

- For the third set of data, a bilinear stiffness model is proposed,$M^{(2)}$, to account for the in and out of contact behaviour of the blocks. The model is dependent on the set offset, d:

$$\begin{cases} m_1\ddot{z}_1 + (c_1+c_2)\dot{z}_1 - c_2\dot{z}_2 + (k_1+k_x)z_1 - k_x z_2 = -m_1\ddot{y}_b \\ m_2\ddot{z}_2 - c_2\dot{z}_1 + c_2\dot{z}_2 - k_x z_1 + k_x z_2 = -m_2\ddot{y}_b \end{cases} \tag{6.4}$$

  where $k_x = k_2$, for $z_2 - z_1 > d$ and $k_x = k_2'$, for $z_2 - z_1 \leq d$.

Figure 6.24 displays the bilinear stiffness behaviour for force, F against displacement, z.

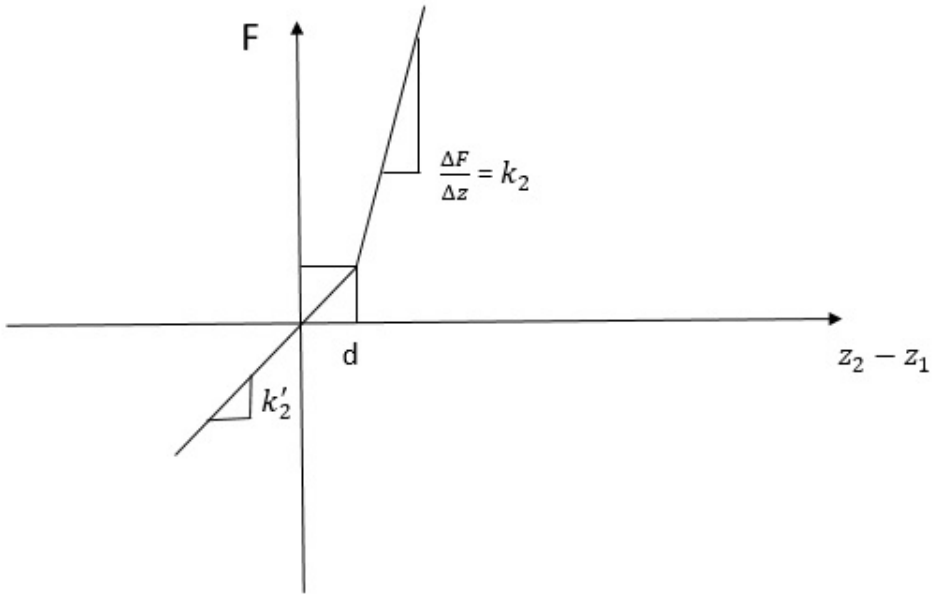

Figure 6.24: Bilinear stiffness behaviour

6.3.3 Data points considerations

The first step was to apply the algorithm on the data gathered under linear behaviour. Due to the fact that the amount of data was too high, a smaller amount of 6000 data points was decided upon such that they represent the dynamic behaviour. The power spectral density was generated for the case of using the whole data set of 186368 data points and only 6000 data points, between data points 180000 and 186000 and they were plotted against each other for the 2-DOF. The range of data points were chosen so that it accounts for the transient response. The PSD plots for the nonlinear behaviour data set are shown in **Figure 6.25** and **Figure 6.26**.

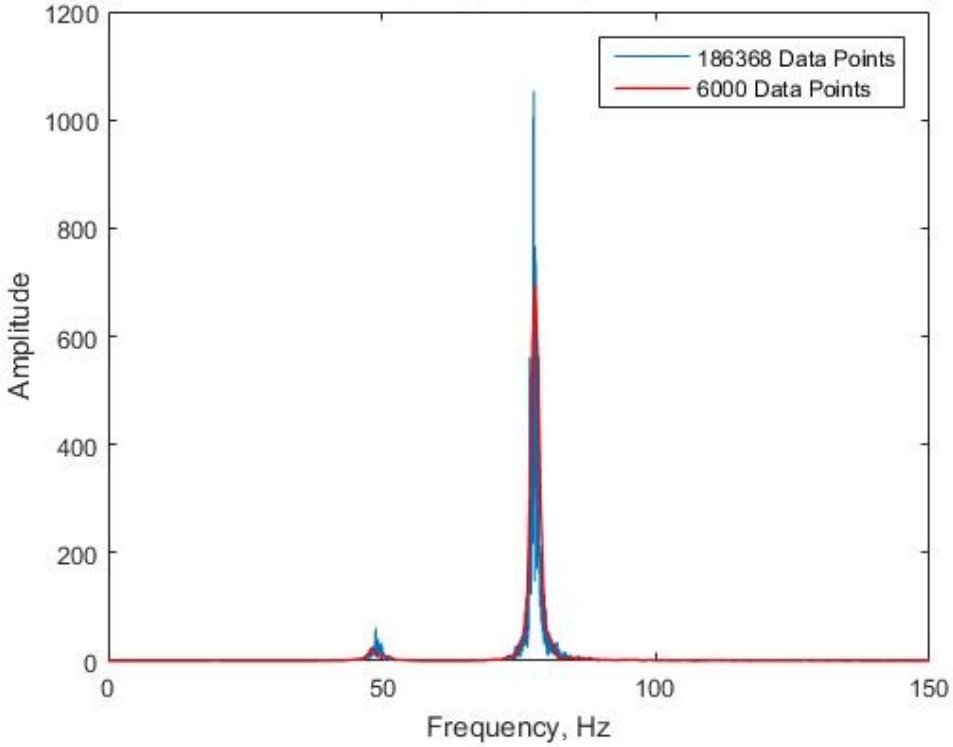

Figure 6.25: PSD First DOF Nonlinear Data

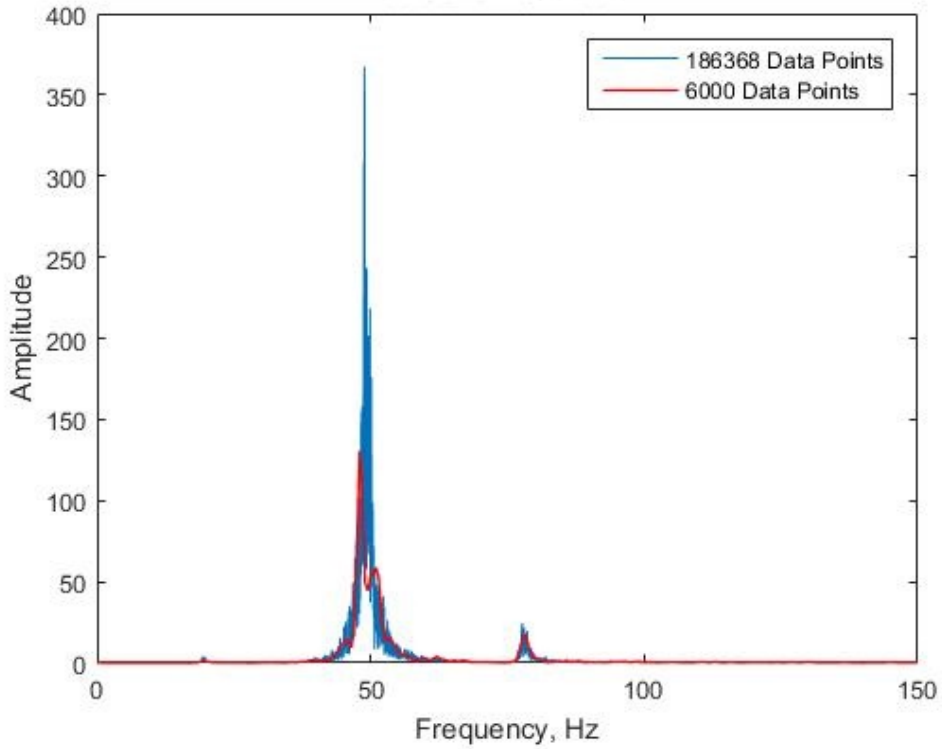

Figure 6.26: PSD Second DOF Nonlinear Data

The PSD for the linear behaviour data set are shown in **Figure 6.27** and **Figure 6.28**. It can be seen that even though the PSD of the smaller number of data points is lower in amplitude, it does follow the same natural frequencies, i.e. the same dynamic trend.

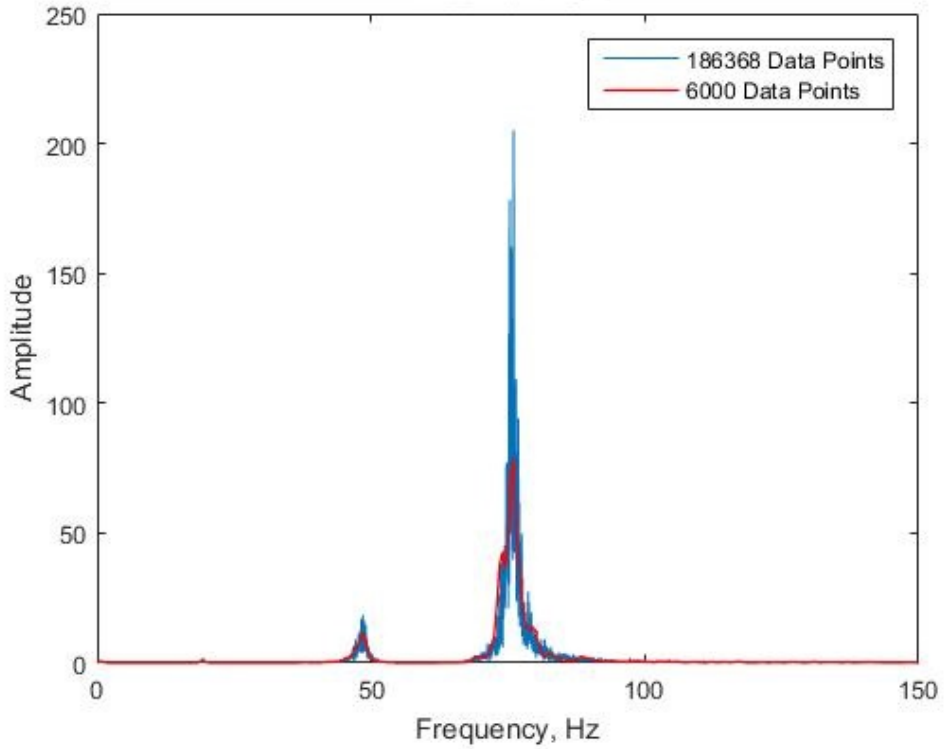

Figure 6.27: PSD Second DOF Linear Data

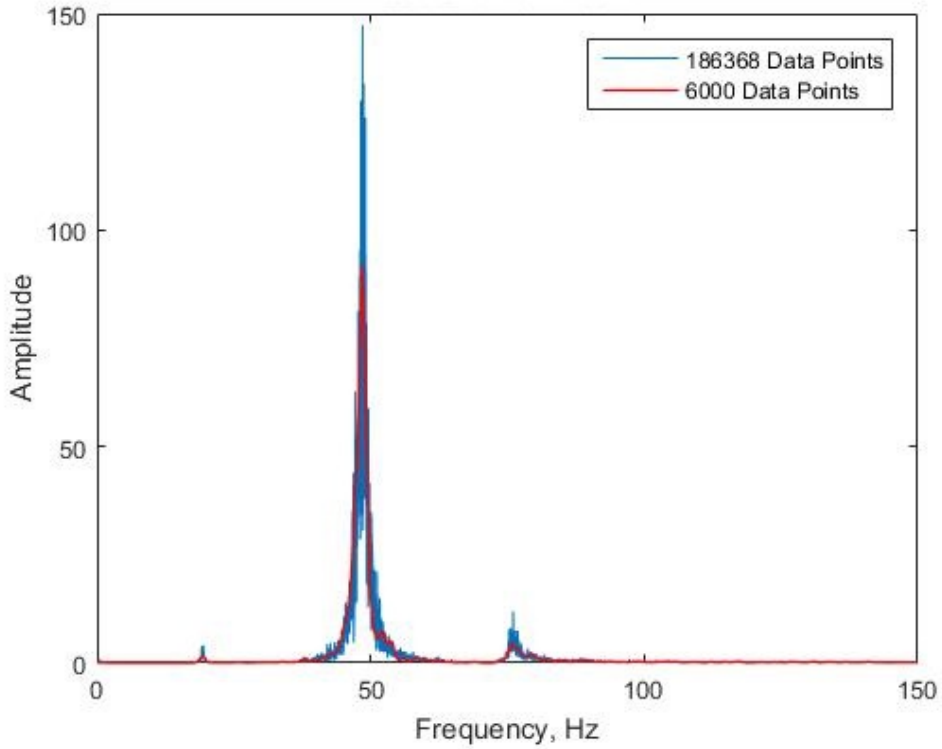

Figure 6.28: PSD Second DOF Linear Data

The role of the PSDs was to check how many data points are needed such that the dynamic behaviour is captured but the computational time of running the algorithm is reduced. This was due to the large amount of data available. A trade off was reached with using 6000 data points out of the 186368 data points available; the 6000 data points were chosen between 180000 and 186000 in order to account for the transient response as well.

6.3.4 Mathematical model for the linear system

The following mathematical model was proposed for the linear 2-DOF system represented in **Figure 6.29**:

$$\begin{cases} m_1\ddot{z}_1 + (c_1+c_2)\dot{z}_1 - c_2\dot{z}_2 + (k_1+k_2)z_1 - k_2 z_2 = -m_1\ddot{y}_b \\ m_2\ddot{z}_2 - c_2\dot{z}_1 + c_2\dot{z}_2 - k_2 z_1 + k_2 z_2 = -m_2\ddot{y}_b \end{cases} \tag{6.5}$$

where m_1, m_2 are the masses of levels 2, 3 (counting from the lowest level to the highest level), $\ddot{z}_1$, $\ddot{z}_2$ are the relative accelerations of each corresponding level with respect to the base (as before), it follows that $\dot{z}_1$, $\dot{z}_2$ are the relative velocities of each corresponding level with respect to the base, z_1, z_2 are the relative displacements of each corresponding level with respect to the base and finally $\ddot{y}_b$ is the acceleration of the base.

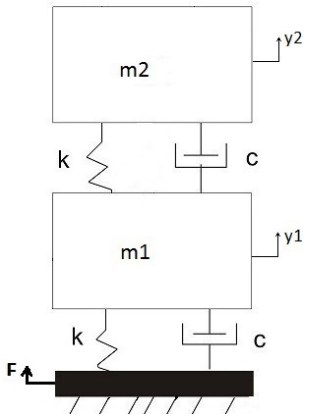

Figure 6.29: 2-DOFs Linear system

The following section covers parameter estimation for the linear model using the MH algorithm.

6.3.5 MH parameter estimation of the linear system

The first step in identifying the parameters by using MCMC sampling algorithms was to find an approximation of the starting point for the sampling procedure. It is straightforward to get approximations of the masses of the three levels. The masses were simply weighed with a precision scale that gave values of $m_1 = 6.6450kg$ and $m_2 = 6.7713kg$. Bear in mind that the model used was base excited so the mass of the lowest level did not form part of the identification process. Knowing the material of the rig, which was aluminium, and assuming that the stiffness was mainly provided by the vertical beams made it possible to calculate the stiffness coefficient by employing cantilever beams linear springs formulas. As there are so many con-

ditions of loading, the approach did not seem feasible and instead SADE software was ran on the gathered data to get an approximation of the parameters, including damping coefficients. SADE stands for Self-Adaptive Differential Evolution algorithm and is a method that applies stochastic search techniques in order to identify or adapt parameters in the case of system identification [81]. Running SADE on a big data set can be computationally time consuming as it can sometimes take days to reach convergence to a solution. In this particular work, due to the problems encountered previously with the rig, the purpose of using SADE was robustness and also to provide starting points for the chains of parameters for the MH algorithm.

The same base acceleration model described in the above equations of motion was used to get these results that can be observed in the table below. It made sense for the masses not to be identified as it saved computational time. The model was checked with these new parameters by doing a forward run.

Model	$k_1(N/m)$	$k_2(N/m)$	$c_1(Ns/m)$	$c_2(Ns/m)$
$M^{(1)}$	477330	442600	62.9771	16.1845

Table 6.7: SADE parameter values for $M^{(1)}$

6.3.6 Results

In **Figure 6.30** and **Figure 6.31** one can see the results of the forward run of the base acceleration model with parameters identified through SADE method for the first and second levels respectively. A reduced range of data points is presented due to the large set of data available. However, this behaviour is representative of that seen across the validation data set.

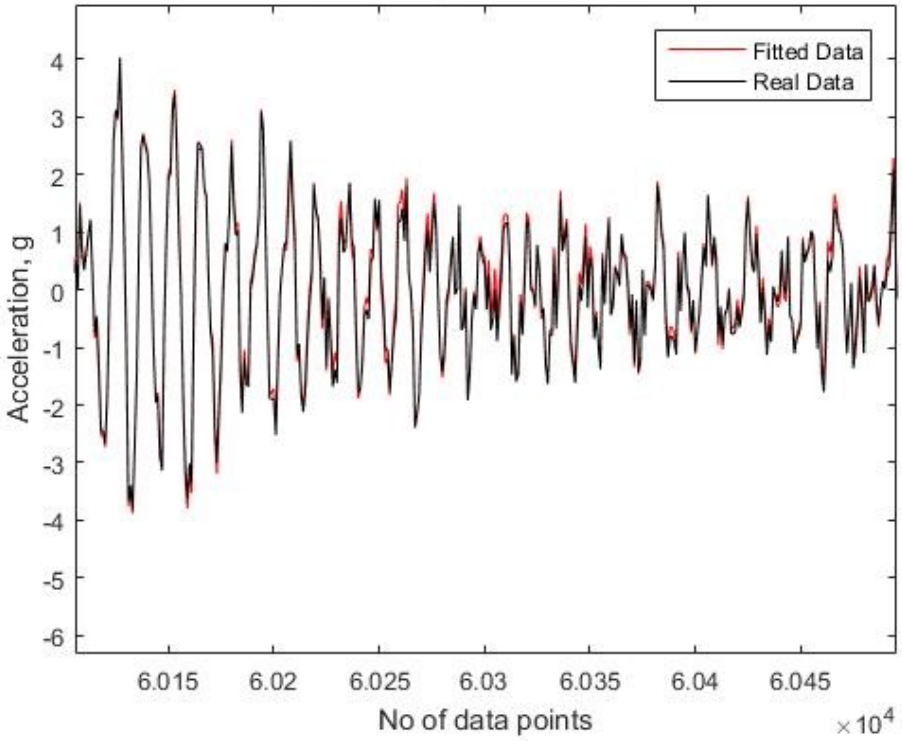

Figure 6.30: Forward Run Results of SADE simulations, 1^{st} level

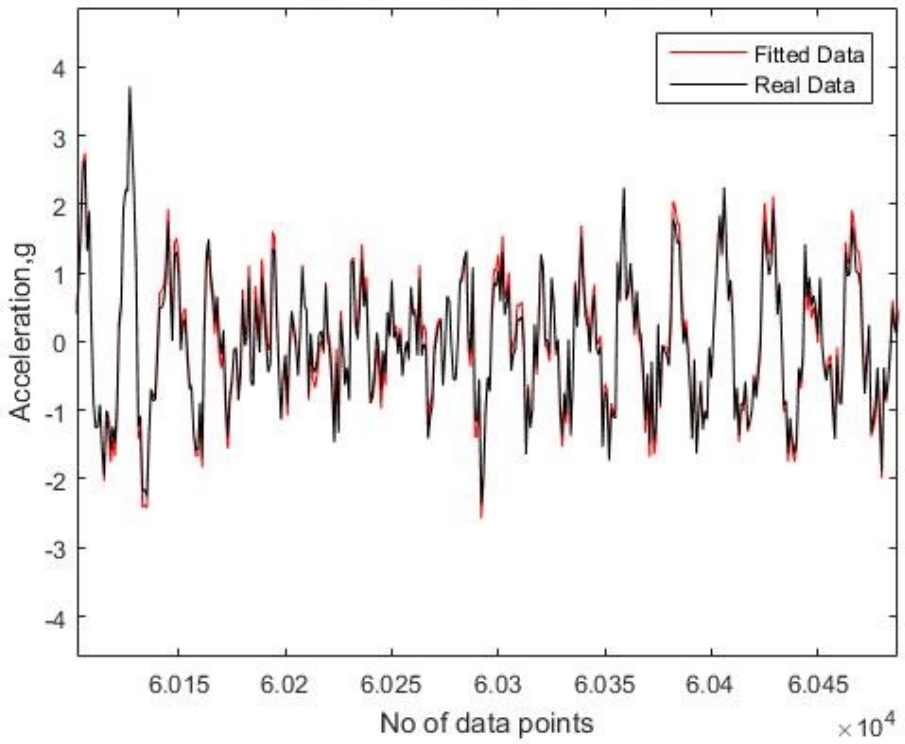

Figure 6.31: Forward Run Results of SADE simulations, 2^{nd} level

The mean square error(MSE) was calculated using the following equation, in order to study the fit of the model to the data.

$$MSE = \frac{\sum (D - \bar{D})^2}{N} \tag{6.6}$$

where D is the experimental data, $\bar{D}$ is the simulated data and N is the number of data points.

For the first level fitting, $MSE = 0.0167$ while for the second level data fitting, $MSE = 0.0216$. These results indicate an excellent fit of the model with the identified parameters to the experimental data.

The results obtained through SADE simulations were further used in MH simulations as Markov chains starting points in order to obtain estimates of the parameters. The MCMC algorithms provide probability distributions over the parameters of interest, covering a bigger area of possible solutions, with the extra benefit of accounting uncertainty in the process of parameter estimation.

In order to simplify the process of finding good proposals for all four parameters, the MH algorithm was applied individually to each parameter first. This made the process of finding appropriate proposal distributions considerably more efficient. By identifying only one parameter at a time while the rest are kept as constants, the appropriate proposal widths could be identified such that the Markov chains do not spend too much time in areas of low acceptance.

In **Figure 6.32** one can see the results of the final MH run on all four parameters simultaneously. The number of necessary iterations was drastically reduced to only 1000 iterations in order to get convergence due to applying it first to each parameter in turn.

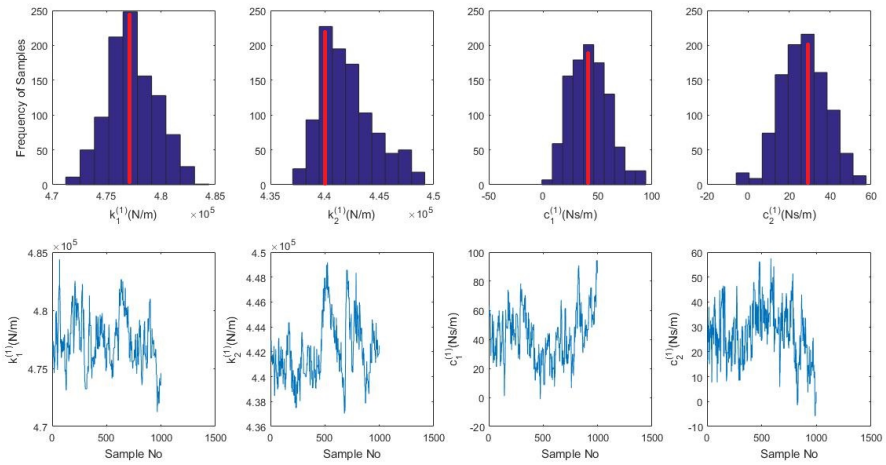

Figure 6.32: Parameter Estimation - MH method results(blue), most frequent values(red)

The same approach of calculating the MSE was followed after getting the estimate of the parameters by using the mean value for each distribution. The mean values of the parameters identified with the MH algorithm are displayed in the table below. The mean squared error was calculated to be $MSE = 0.0164$ for the first level and $MSE = 0.0233$ for the second level. This analysis showed that the parameters estimated by the MH algorithm for the linear behaviour of the rig were appropriate with the used model.

Model	$k_1(N/m)$	$k_2(N/m)$	$c_1(Ns/m)$	$c_2(Ns/m)$
$M^{(1)}$	477500	442600	41.7	29

Table 6.8: MH parameter values for $M^{(1)}$

For comparison considerations, **Figure 6.33** and **Figure 6.34** are included of a chosen range of data points for running the model forward with identified parameters using the MH algorithm; approximately the same range of data points that were captioned for the SADE simulations.

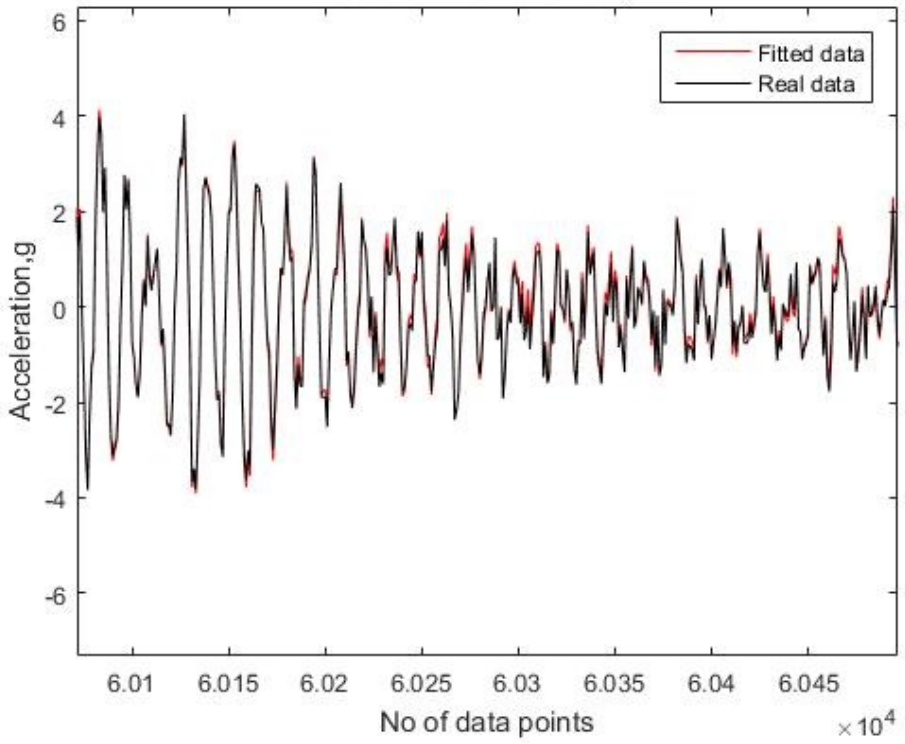

Figure 6.33: Forward Run Results of MH simulations, 1^{st} level

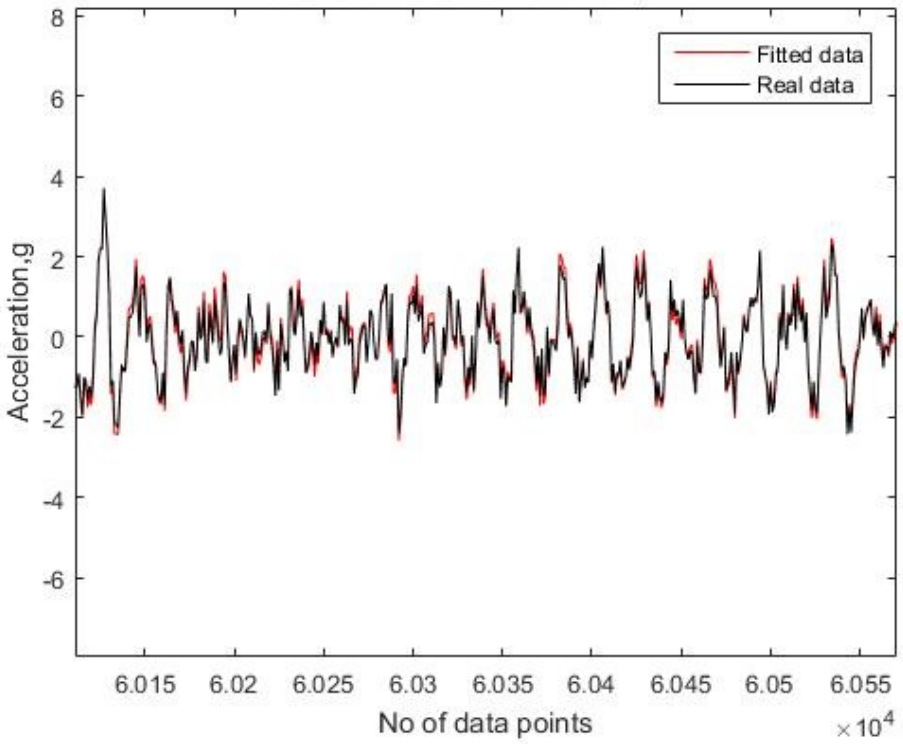

Figure 6.34: Forward Run Results of MH simulations, 2^{nd} level

The SADE algorithm and the MH method gave different results for the damping coefficients. This is not considered to have significant relevance for the studied rig and for the modelling approach chosen, which can be observed by looking at the forward model runs plots when using the parameters identified with SADE algorithm and the parameters identified with the MH method. Even though there is quite a difference in the damping coefficients values, both models fit the data with low error.

6.3.7 Mathematical model for the nonlinear system

The nonlinear system is graphically represented in **Figure 6.35**:

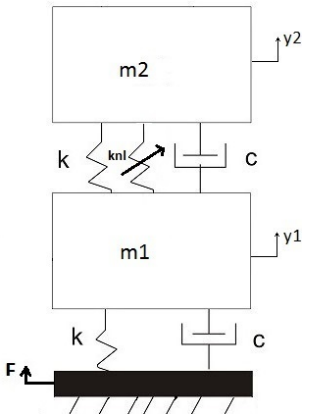

Figure 6.35: 2-DOFs Nonlinear system

The bilinear stiffness model is considered,$M^{(2)}$, to account for the in and out of contact behaviour of the blocks. The model is dependent on the set offset, d:

$$\begin{cases} m_1\ddot{z}_1 + (c_1+c_2)\dot{z}_1 - c_2\dot{z}_2 + (k_1+k_x)z_1 - k_x z_2 = -m_1\ddot{y}_b \\ m_2\ddot{z}_2 - c_2\dot{z}_1 + c_2\dot{z}_2 - k_x z_1 + k_x z_2 = -m_2\ddot{y}_b \end{cases} \tag{6.7}$$

where $k_x = k_2$, for $z_2 - z_1 > d$ and $k_x = k_2'$, for $z_2 - z_1 \leq d$.

6.3.8 MH parameters estimation on the nonlinear system

The MH algorithm was then applied to the nonlinear system using the experimentally gathered data. The number of data points used was, as decided, 6000 data points, between 180000 and 186000. The second model is characterised by 6 parameters,$k_1^{(2)}(N/m)$, $k_2^{(2)}(N/m)$, $k_2^{\prime(2)}(N/m)$, $c_1^{(2)}(Ns/m)$, $c_2^{(2)}(Ns/m)$, $d(m)$, for which the proposal widths are tabulated below. 10000 iterations for the MH algorithm were used in this study.

Parameters	$k_1^{(2)}(N/m)$	$k_2^{(2)}(N/m)$	$k_2'^{(2)}(N/m)$	$c_1^{(2)}(Ns/m)$	$c_2^{(2)}(Ns/m)$	$d(m)$
Proposal widths	1000	1000	10000	1	1	10^{-6}

Table 6.9: Proposal widths for the parameters to be identified, $M^{(2)}$

The next section covers the results of doing parameter estimation on $M^{(2)}$, using the MH algorithm.

6.3.9 Results

In **Figure 6.36** and **Figure 6.37** the histogram and the sampling history for the parameters of $M^{(2)}$ are displayed.

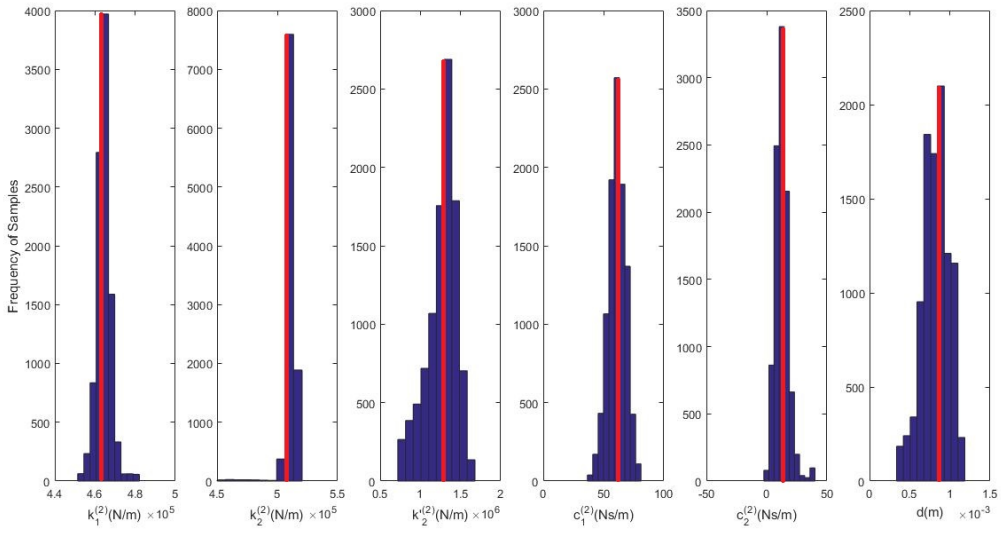

Figure 6.36: MH histograms of parameter estimates, $M^{(2)}$(blue) and most frequent values(red)

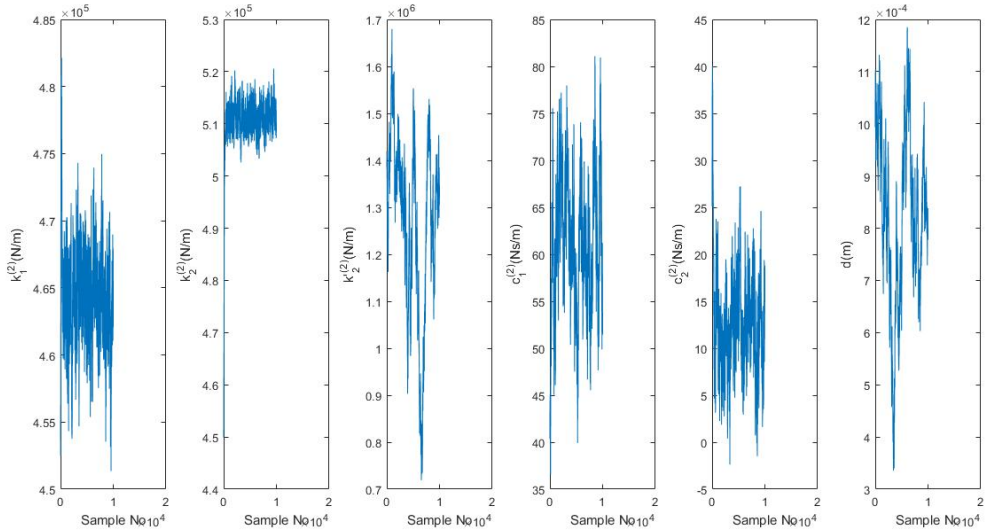

Figure 6.37: MH history of samples, $M^{(2)}$

The means of the parameters value are tabulated below.

Model	$k_1^{(2)}(N/m)$	$k_2^{(2)}(N/m)$	$k_2'^{(2)}(N/m)$	$c_1^{(2)}(Ns/m)$	$c_2^{(2)}(Ns/m)$	$d(m)$
$M^{(2)}$	465000	510000	1340000	61.1	12.5	0.08x10^{-3}

Table 6.10: MH parameter values for $M^{(2)}$

The mean squared error was calculated to be $MSE = 0.2082$ for the first level and $MSE = 0.1045$ for the second level. Comparison of data from the forward run of the second model with the identified parameters can be observed in **Figure 6.38** for the first level and **Figure 6.39** for the second level.

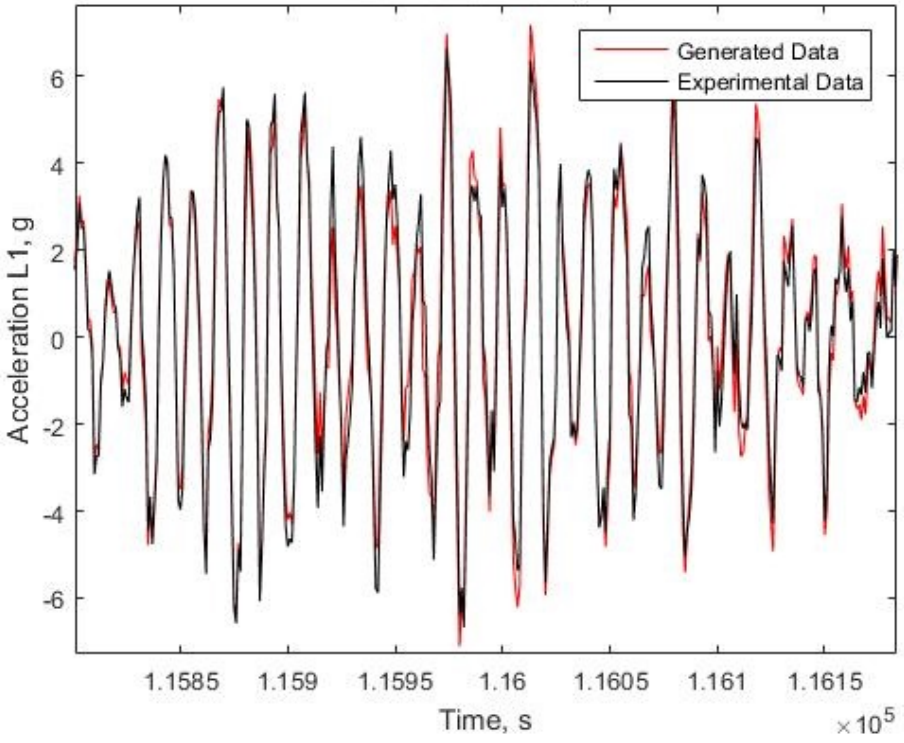

Figure 6.38: Forward Run Results of MH simulations, 1^{st} level

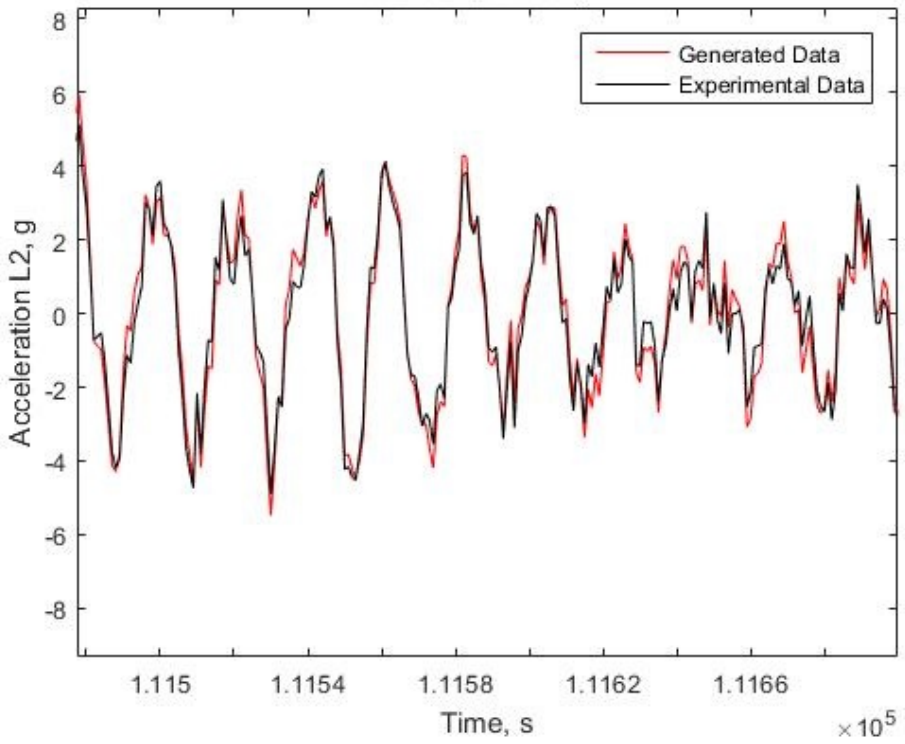

Figure 6.39: Forward Run Results of MH simulations, 2^{nd} level

By firstly applying the MH algorithm on the linear and nonlinear systems considered, the parameters of each model were identified. This provided a starting point for applying the RJMCMC algorithm. The mean of the estimated parameters were used as starting points for the Markov chains in the RJMCMC application.

6.3.10 Experimental Rig with nonlinear element

The nonlinear element was included in the rig by adding two impacting blocks. Looking back at **Figure 6.18**, it can be noticed that there are two 'L'-shaped blocks included between the two top floors of the structure, as seen in the closed-up caption in **Figure 6.40**.

Figure 6.40: Source of nonlinearity

The idea behind this particular design is that once the excitation is increased, the 'L' blocks will start impacting each other. In order to control the distance between the blocks, a micrometer was attached, as in **Figure 6.41**.

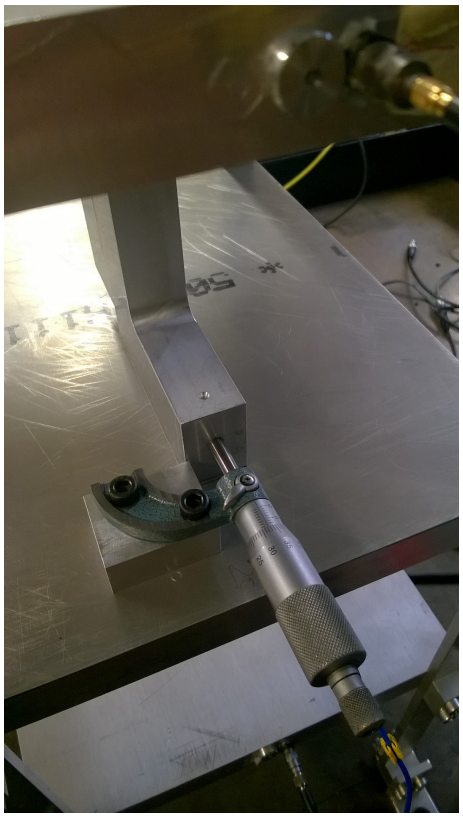

Figure 6.41: Attached micrometer device

Due to the large area of the two blocks touching, under higher excitation the top two floors coupled. In itself, an interesting case but beyond the purpose of this work. The rig was further modified for added flexibility in the 'L' blocks by thinning the top 'L' block and for less impact area by cutting down the bottom 'L' block to a rectangular one as seen in **Figure 6.42**. This provided a rig in which the nonlinearity is controllable and is characterised as a bilinear stiffness effect.

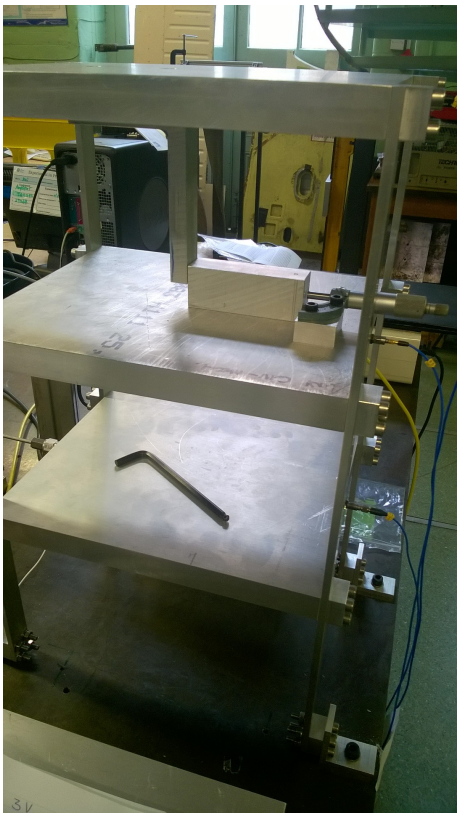

Figure 6.42: Redesigned 'L'-shaped blocks

6.3.11 RJMCMC on the 2-DOF experimental structure - first scenario

The parameters of the linear and nonlinear systems were identified by using the MH algorithm in **Section 6.3.5** and **Section 6.3.8**. The RJMCMC algorithm was applied for two different scenarios. The first one included the linear model,$M^{(1)}$, and the bilinear stiffness model,$M^{(2)}$. For this case, $M^{(1)}$ depends on 4 parameters to be identified ($M^{(1)} = \{k_1^{(1)}, k_2^{(1)}, c_1^{(1)}, c_2^{(1)}\}$), while the two masses, m_1 and m_2 are considered approximately known as they were weighed. The bilinear stiffness model, $M^{(2)}$, is dependent on 5 parameters, again with the masses being considered known($M^{(2)} = \{k_1^{(2)}, k_2^{(2)}, k_2'^{(2)}, c_1^{(2)}, c_2^{(2)}\}$).

6.3.12 Results

This time the RJMCMC method was applied for 25000 iterations, the starting parameters were updated after several runs of the MH algorithm. The proposals for both models are presented in the table below.

Parameters	$k_1(N/m)$	$k_2(N/m)$	$k_2'(N/m)$	$c_1(Ns/m)$	$c_2(Ns/m)$
Proposal widths, $M^{(1)}$	1000	1000	-	1	1
Proposal widths, $M^{(2)}$	1000	1000	10000	1	1

Table 6.11: Proposal widths for the parameters to be identified, $M^{(1)}$ and $M^{(2)}$

Once the RJMCMC algorithm was applied on the linear data set, it was clear that the confidence was in the linear model. As such, the algorithm spent no time in the second model, the bilinear stiffness one. In **Figure 6.43** it can be observed that the parameters were estimated, with the stiffness parameters being more certain. The same aspect can be observed in **Figure 6.44**. The Markov chains for the stiffness parameters settled after only approximately 1000 iterations, while the Markov chains for the damping parameters took more time to become stationary.

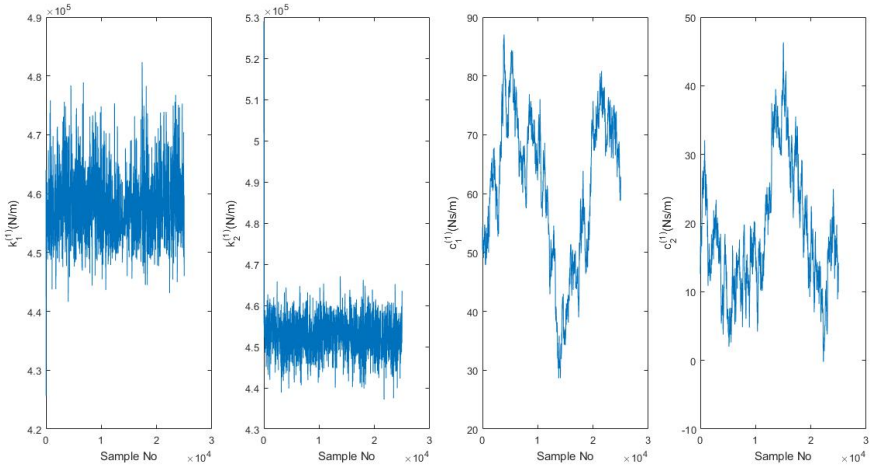

Figure 6.44: History of Samples for $M^{(1)}$

As used so far, in **Figure 6.45** the bar chart for the number of samples spent in the two models is plotted.

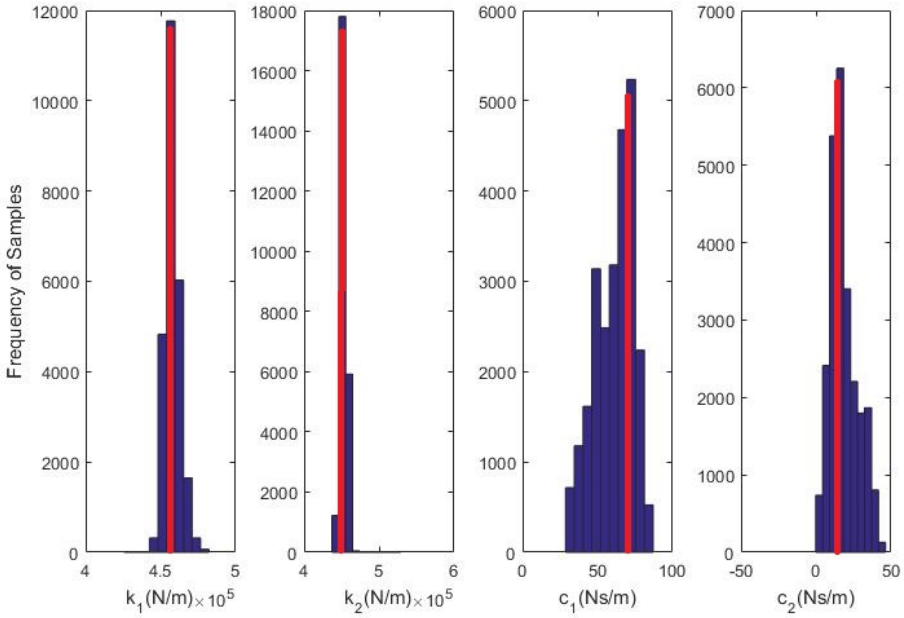

Figure 6.43: Histogram of Samples for $M^{(1)}$(blue) and most frequent values(red)

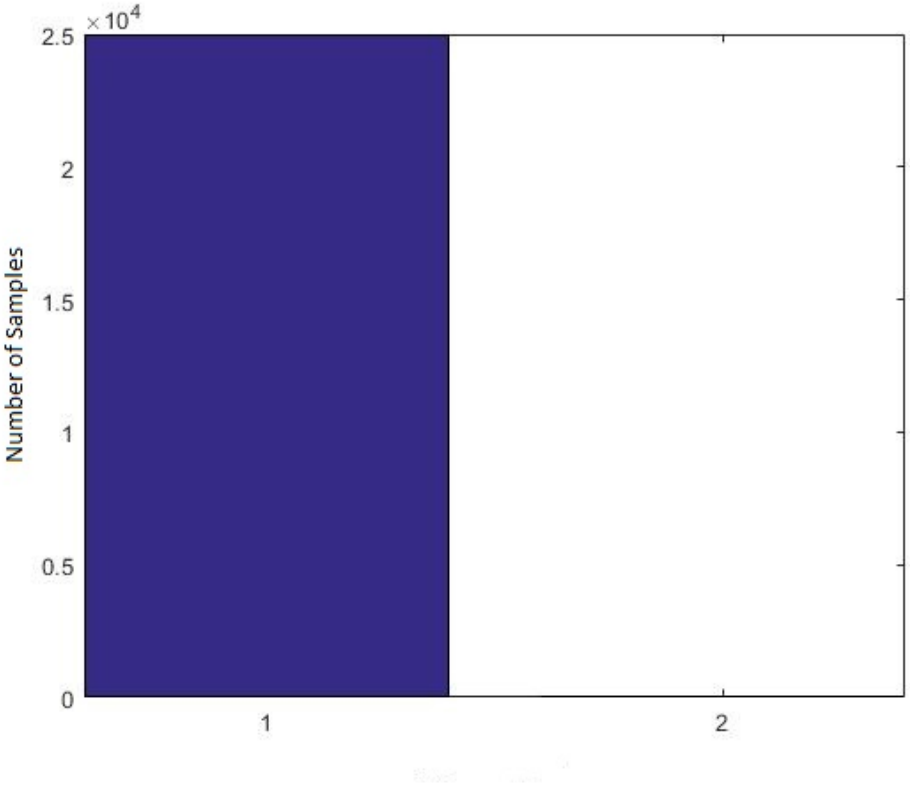

Figure 6.45: Acceptance frequency of $M^{(1)}$ vs $M^{(2)}$

The second step was to input the nonlinear data into the algorithm and identify it. Unfortunately the RJMCMC method was choosing the first model as being the best fit for the nonlinear data as well. The mean squared error was then calculated for the both cases of using the nonlinear data with the linear model and the bilinear stiffness model and the values were found to be the same: $MSE = 0.4350$ for the first level and $MSE = 0.2408$ for the second level. While these low values indicate a good fit, they prove the linear model to be a better fit for the nonlinear gathered data. One of the advantages of using a Bayesian framework is that it penalises complexity, which was the case in this scenario. This can mean that the nonlinearity introduced was not strong enough or that the rig needs to be driven at higher excitation levels where the nonlinearity can be more influential.

Model	$k_1(N/m)$	$k_2(N/m)$	$c_1(Ns/m)$	$c_2(Ns/m)$
$M^{(1)}$	457000	451000	72.4	16.1

Table 6.12: RJMCMC parameter values for $M^{(1)}$

In order to check the results, the BIC was calculated for using both the linear and nonlinear model with the nonlinear data according to the formula:

$$BIC = -2\ln(L) + k\ln(n) \tag{6.8}$$

where L is the maximized likelihood function, k is the number of parameters to be identified and n is the number of data points. The BIC is calculated for all the models and then the lowest BIC value determines the preferred model. For the nonlinear model the $BIC = 1.3392e+06$ while for the linear model the $BIC = 1.3392e+06$ which once again proves that the RJMCMC algorithm did the right thing by penalising the complexity and choosing the linear model for the nonlinear data set. According to the AIC measure, $AIC = 1.3391e+06$ for the nonlinear model and $AIC = 1.3391e+06$ for the linear one where the AIC is calculated as:

$$AIC = 2k - 2\ln(L) \tag{6.9}$$

with k and L being defined as before. This time, the preferred model is the one with the lowest AIC, again, the linear model.

6.3.13 RJMCMC on the 2-DOF experimental structure - second scenario

The second scenario involved applying the RJMCMC algorithm on the linear model,$M^{(1)}$ and the bilinear stiffness model,$M^{(2)}$. For this case study the parameters to be identified were slightly changed such that the linear model $M^{(1)}$ stayed the same, while the nonlinear model, $M^{(2)}$ had an additional parameter to be identified, d, the offset for the two middle blocks so that $M^{(2)} = \{k_1^{(2)}, k_2^{(2)}, k_2'^{(2)}, c_1^{(2)}, c_2^{(2)}, d\}$.

As before, firstly the RJMCMC algorithm is used to identify the linear structure, so that by being applied to the linear set of gathered data it provides estimates over parameters and models. The method was employed with 25000 iterations once again. The proposal densities were tuned accordingly. In **Figure 6.46** the histogram for the considered samples is plotted. For the two stiffness, $k_1^{(1)}$ and $k_2^{(1)}$ the maximum peaks are very clearly marked, which can be observed in **Figure 6.47**, in the history of the samples. The Markov chains are stationary almost throughout the 25000 iterations. For the damping coefficients, $c_1^{(1)}$ and $c_2^{(1)}$, more time was required in order for the chains to become stationary. For $c_1^{(1)}$ an increased number of iterations would provide even better results.

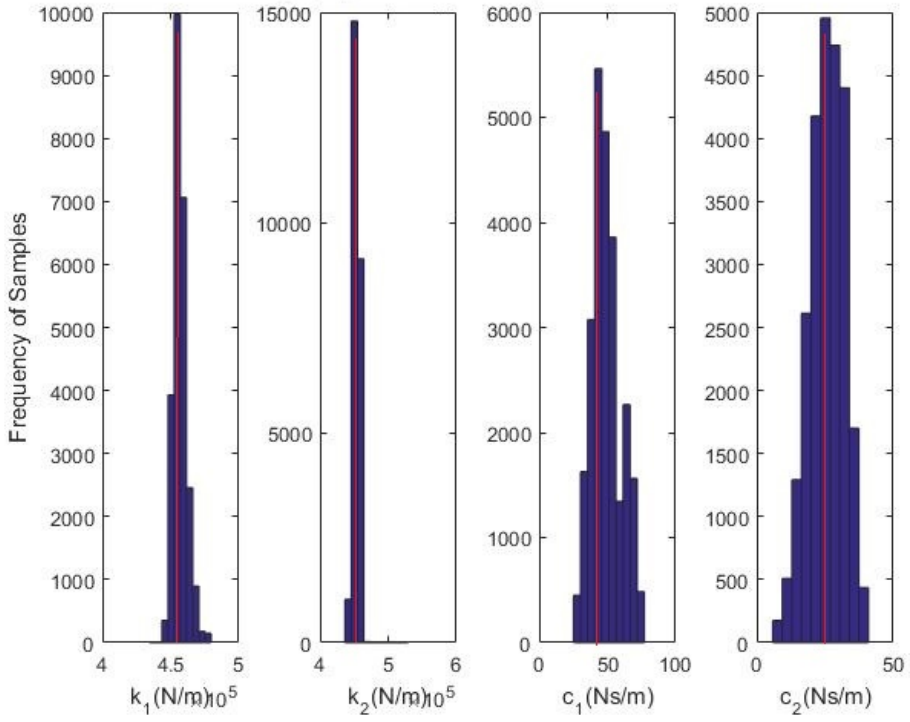

Figure 6.46: Histogram of Samples for $M^{(1)}$(blue) and most frequent values(red)

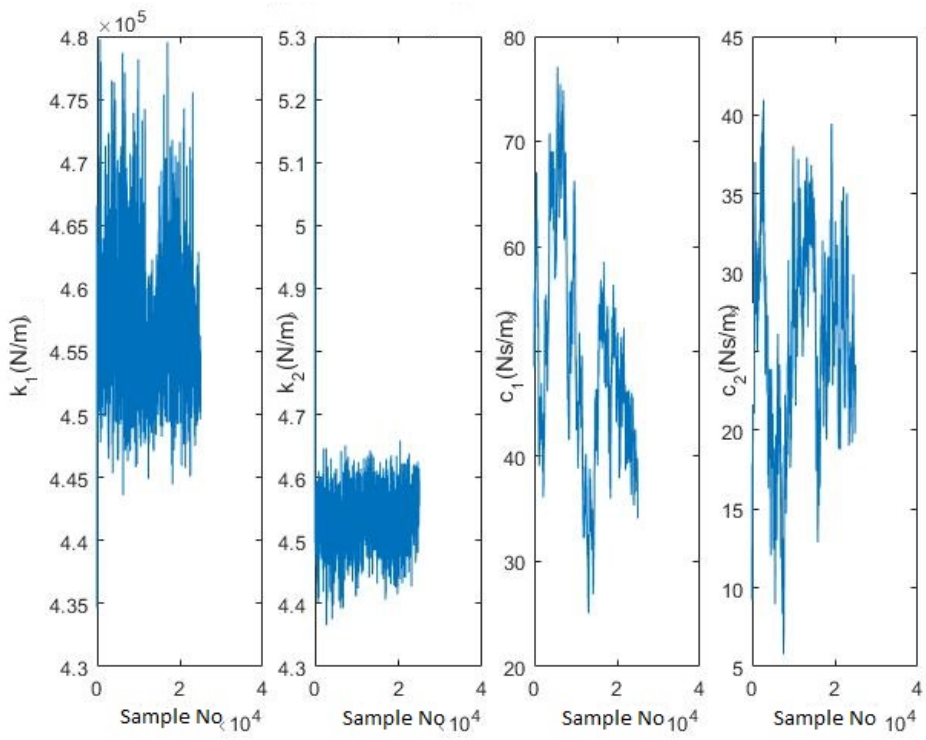

Figure 6.47: History of Samples for $M^{(1)}$

Model	$k_1(N/m)$	$k_2(N/m)$	$c_1(Ns/m)$	$c_2(Ns/m)$
$M^{(1)}$	455000	450000	43.3	25.1

Table 6.13: RJMCMC parameter values for $M^{(1)}$

A forward run of the model by using the estimated parameters, gave mean squared errors between the generated data set and the gathered data set of $MSE = \{0.0174, 0.0200\}$, for each DOF respectively. **Figure 6.48** and **Figure 6.49** show a small sample of the generated data points against the experimental data points for each level respectively.

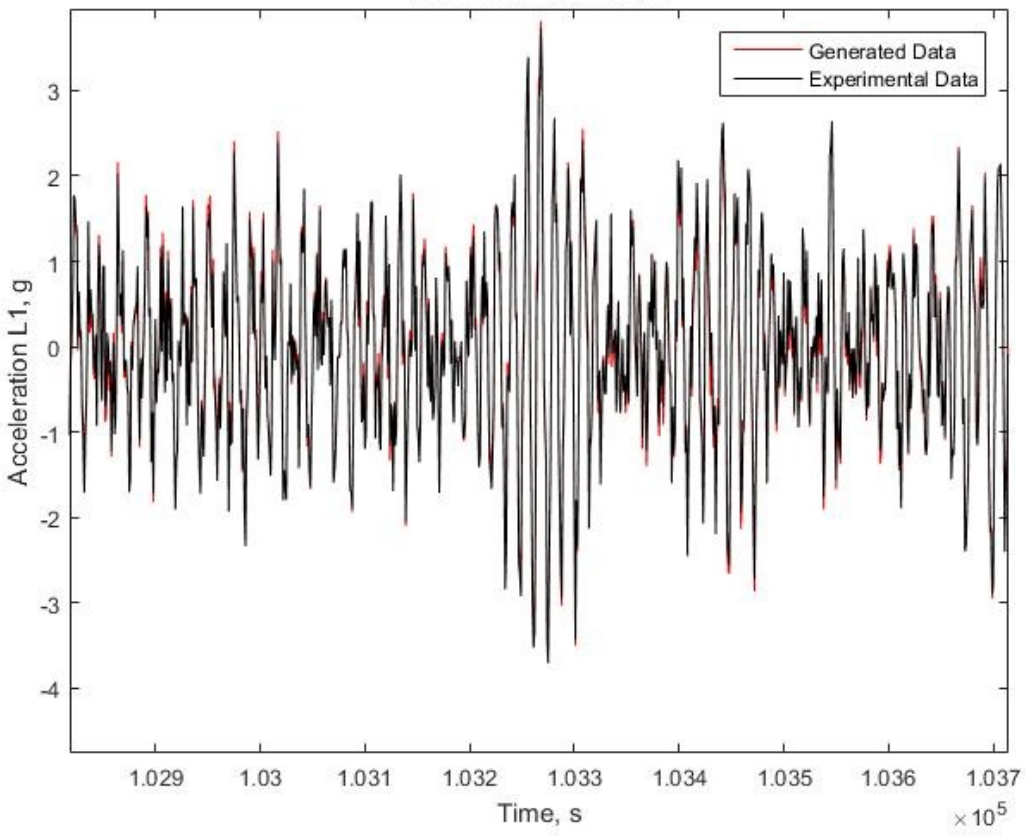

Figure 6.48: Data comparison plot, Level 1

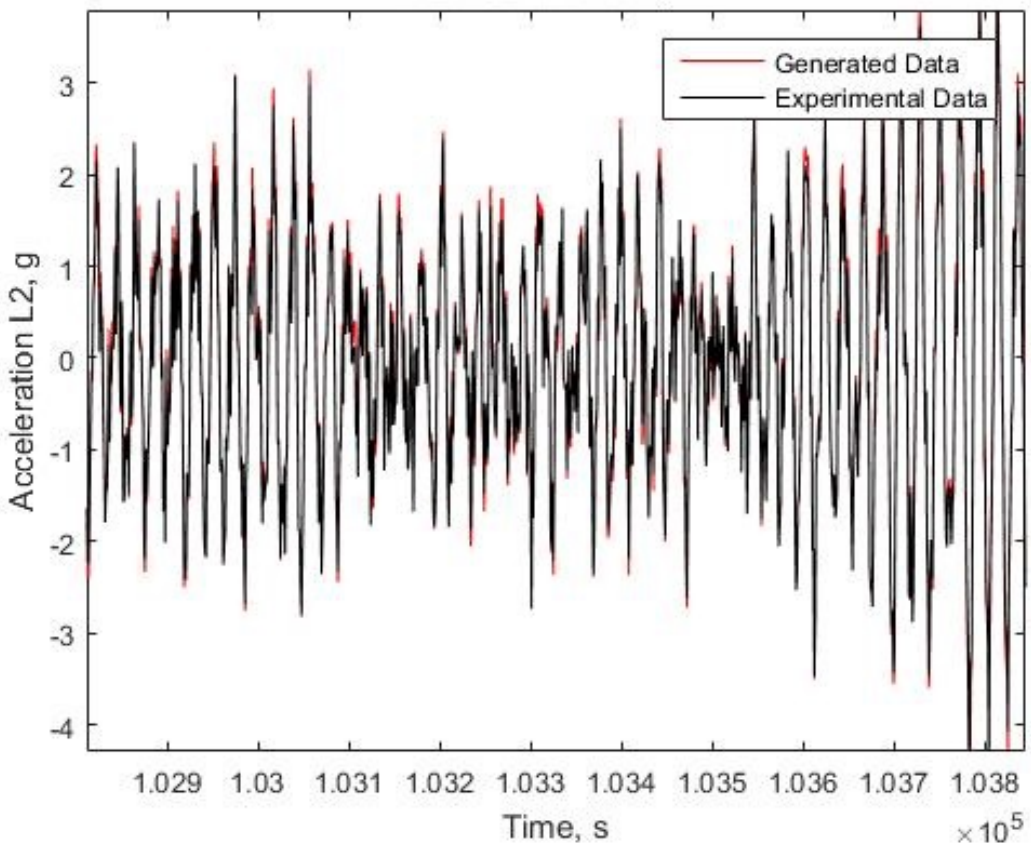

Figure 6.49: Data comparison plot, Level 2

The next step was to apply the RJMCMC on the second data set of gathered data, the one representing the nonlinear behaviour. The situation stayed the same as with the first approached scenario. The RJMCMC method chose the linear model, $M^{(1)}$ to best fit the nonlinear data. **Figure 6.50** shows the bar chart of number of iterations spent in $M^{(1)}$ against number of iterations spent in $M^{(2)}$.

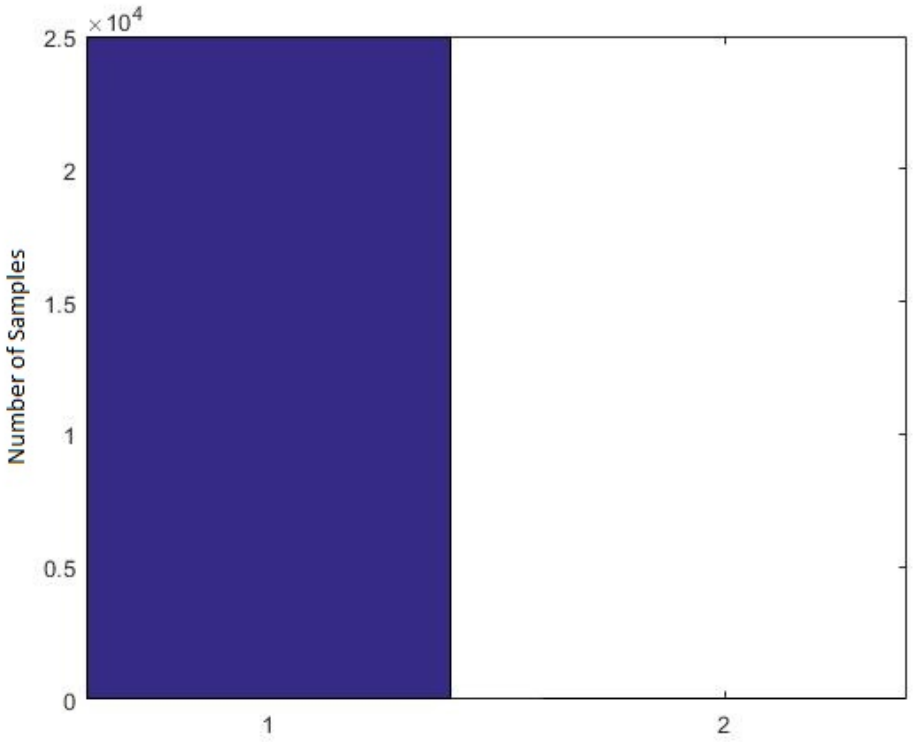

Figure 6.50: Acceptance frequency of $M^{(1)}$ vs $M^{(2)}$

As with the first scenario, when the RJMCMC was applied on what was believed to be nonlinear data, the expectation was that the method would choose the second model, $M^{(2)}$, to be the better fit. However, the algorithm preferred $M^{(1)}$ to best fit the data. Firstly, this raises the issues of having appropriate models to represent the dynamics in a structure. It is believed that the nonlinear model chosen for this particular study did not best represent the present nonlinearity, at least not better than the linear model. Secondly, when dealing with random excitation and a harsh nonlinearity, such as the impacting beams considered, the nonlinearity is not always apparent. Sometimes, periodic signals can be more appropriate in order to observe the effect of the nonlinear element.

6.4 Summary

Chapter 6 covered the application of the RJMCMC algorithm on experimental data in order to do system identification. The work was split into two main sec-

tions. **Section 6.2** concentrated on an initial design of a 4-DOF structure, modelled as a 3-DOF system with base acceleration. Two competing models were developed for the 3-DOF system, based on the noise variance of each DOF such that the first model considered modelling the noise variance equally for each of the 3-DOFs and the second model took into account different noise variance for each 3-DOFs. Each value of the noise variance was identified together with the stiffness and damping coefficient parameters. The MH algorithm was applied individually on each of the considered models in order to identify the parameters. The limitations of the MH algorithm means that it cannot do model selection. This was followed by the application of the RJMCMC algorithm on both models in order to do system identification, making use of the parameters identified with the MH method as a starting point for the Markov chains. At the end of **Section 6.2** the rig was re-designed for the purpose of controlling the nonlinearity present, due to unexpected design issues. **Section 6.3** covered the application of the RJMCMC algorithm on a 3-DOF structure, modelled as a 2-DOF system with base acceleration, with the purpose of doing linear and nonlinear system identification. The MH algorithm was first applied on each of the considered models in order to give estimates for the parameters. These estimates helped in further applying the RJMCMC algorithm. The application of the RJMCMC algorithm followed two scenarios. The RJMCMC method correctly identified the linear model as to best fit the linear data and simultaneously provided parameter estimates in both scenarios. The RJMCMC method did not choose the second model to best fit the nonlinear data due to the fact that the linear model could fit the nonlinear data as well. As the first linear model is less complex, the RJMCMC algorithm chose the first model in this case as well. Further analysis by using model selection criterions such as BIC and AIC followed to support the findings. **Chapter 7** concludes the thesis and provides further discussion on the findings presented in each chapter.

Chapter 7

Conclusions, Discussion and Future Work

Chapter 7 finalises the present thesis. The aim of this thesis was to introduce and apply a not so well known algorithm, the RJMCMC method, in system identification of linear and nonlinear dynamical structures. Motivated by a lack of computational tools that are capable to both probabilistically estimate parameters and select between models, the author proposed working with an MCMC sampling algorithm, within Bayesian inference for advantages such as being capable of sampling from complex distributions and a framework that addresses uncertainty. A set of objectives were decided upon at the beginning of this work:

- Introduce the RJMCMC method to be used in the context of structural dynamics in order to do system identification of linear and nonlinear structures in a probabilistic framework, by simultaneously providing estimates of the parameters of a system and selecting a robust model to best fit the available data set;

- Adapt and apply the RJMCMC algorithm on a series of numerical case studies of SDOF(single degree of freedom) and MDOF(multi-degree of freedom) scenarios in order to show its capabilities in doing system identification of linear and nonlinear structures;

- Adapt and apply the RJMCMC algorithm on experimental case studies in order to emphasize its capabilities of simultaneously do parameter estimation

and model selection for both linear and nonlinear systems;

- Discuss the RJMCMC method in the context of SID of dynamical systems by underlining its advantages and considering its drawbacks;
- Discuss the RJMCMC algorithm in comparison with other MCMC sampling methods;

The objectives and their success will be analysed in the discussion section.

A summary of the contribution of each chapter is given and the objectives set at the beginning of the thesis are discussed. The contribution to knowledge is emphasized and plans for future work are proposed.

7.1 Thesis Summary

The thesis summary is presented in bullet point format as each individual chapter has a thorough summary at the end of it, such that only the main ideas are stated in the following paragraphs.

Chapter 1:

- The motivation for the thesis is stated;
- A thorough introduction to system identification, Bayesian inference and the Markov Chain Monte Carlo sampling algorithms is given;
- The outline of the thesis is set.

Chapter 2:

- Background literature review is done for structural dynamics and using MCMC sampling algorithms within a Bayesian framework;
- Background literature review is done for areas in which the RJMCMC algorithm was employed;
- Literature review is conducted for using MCMC methods and the RJMCMC algorithm in structural health monitoring.

Chapter 3:

- The theory for Bayesian inference is presented;
- The proof for Bayes theorem is derived;
- A number of MCMC sampling algorithms are discussed.

Chapter 4:

- The theory for the RJMCMC algorithm is explained from a structural dynamics point of view;
- Detailed balance is explained;
- Dimension matching is addressed;
- A theoretical example of system identification using the RJMCMC algorithm is provided.

Chapter 5:

- Numerical case studies are considered in SID of dynamical system for applying the RJMCMC algorithm;
- A SDOF structure for which a linear and a nonlinear model are simulated is identified by using the RJMCMC method;
- A MDOF structure for which a linear and a nonlinear model are simulated is identified by using the RJMCMC method;
- A simply supported beam case study in considered for damage identification by using the RJMCMC algorithm.

Chapter 6:

- Experimental case studies are considered for SID of dynamical systems by using the RJMCMC algorithm;
- A 4-DOF and then 3-DOF rig was built and redesigned for the purpose of studying nonlinearity;
- A MDOF structure for which two scenarios for noise variance exist is identified by using the RJMCMC method;
- A MDOF structure is considered within a linear regime and a nonlinear regime

to be identified by emplying the use of the RJMCMC algorithm.

7.2 Discussion

The objectives for this thesis were introduced in **Chapter 1**. The RJMCMC algorithm was introduced in structural dynamics for system identification of both linear and nonlinear systems. It followed that the RJMCMC method was applied on a range of increasingly challenging scenarios through numerical case studies for both SDOF systems and MDOF systems with different types of nonlinearity (cubic stiffness and bilinear stiffness). Up to this point, the objectives were met. The advantages of using the RJMCMC algorithm are that it can do both parameter estimation and model selection of linear and nonlinear systems. The time the algorithm needs to run increases with the number of parameters it needs to estimate as well as with the number of data points employed. It was shown that the number of data points used can be decreased by analysing the data previously to applying the RJMCMC method. It is expected that the RJMCMC algorithm will take more time to run when the number of models it needs to decide from increases. The RJMCMC algorithm can be used in numerical case studies as it was expected. It provided the expected parameter estimates in the numerical case studies considered, it was capable of selecting the expected model to best fit the employed data. The decision to use the method comes down to the problem set up by the user but in the situations presented in the numerical studies considered in the present work it proved to work as desired. The application of the algorithm on experimental data came with some drawbacks. Firstly, there is a clear requirement for an accurate model. The RJMCMC method will keep choosing the least complex model when made to decide between models that represent the collected data in the same way. This was the case when the algorithm was applied on the experimental structure. The hope was to include a nonlinearity so that the nonlinear model will fit the data better than the linear one. Unfortunately, that was not the case as it is believed that the nonlinearity was not captured accurately by the proposed nonlinear model. This led to the algorithm choosing the linear model for nonlinear data as the fit was similar, the calculated errors were the same and the first model was less complex. It is expected for challenges to increase when trying to do system identification on real structures and this was the case when the experimental case studies were considered. The RJMCMC algorithm should be applied on experimental data only when

the models under consideration are accurate. When nonlinearity is present this issue becomes more pressing.

The broad applicability of the RJMCMC algorithm makes it an attractive method. As it was discussed in **Chapter 2**, RJMCMC algorithm was used in various areas of research, from economics to nuclear physics or biology. Compared to other methods that can do model selection, such as TMCMC algorithm or ABC method, the RJMCMC algorithm can do parameter estimation and model selection simultaneously which is a considerable advantage. If set up properly, the RJMCMC algorithm has an effective computational time run compared with the ABC method for example. The RJMCMC algorithm is flexible in the sense that one is not limited to a certain set of moves(in the present work three types of moves were used, birth move, death move and update move) as it can be adapted to the user's need. Another advantage is that, as long as it is set up to obey the principle of detailed balance and the dimensions are matching, the RJMCMC method can be built around other sampling algorithms, such as Gibbs Sampling; it is not necessary to use at its core the MH algorithm. However, the fact that the width of the proposals for the parameters of interest needs to be tuned according to each case is a drawback as it can potentially be a time consuming task. This is the case with most MCMC sampling algorithm though, which is why the attention right now should be toward automation. Also, the acceptance of move between the considered models needs to improve before the confidence in using the RJMCMC algorithm can increase. These issues need to be addressed before the RJMCMC algorithm can become widely used in industrial situations.

The next section is covering the contribution the work presented thus far has brought to research.

7.3 Contributions to Knowledge

It should be well acknowledged by now that the area of nonlinear system identification of dynamical structures is still lacking in well-developed computational tools that can provide means of doing both parameter estimation and model selection. This thesis aimed to address this lack in computational tools by introducing the RJMCMC method, a powerful MCMC sampling algorithm that to the best of the author's knowledge has not yet been introduced in SID of dynamical systems up

to the point of writing, a method that is capable of simultaneously addressing parameter estimation and model selection for both linear and nonlinear systems. The method had to be broken down into its most important features in order to reassemble it from a structural dynamics point of view. The application of the RJMCMC algorithm was then demonstrated for a sequence of incrementally challenging cases. It followed that the applications of the RJMCMC algorithm had to be built up, starting from a SDOF structure for which two competing models were simulated (a linear model and a nonlinear model with a cubic stiffness), the method successfully fit the appropriate data sets to the corresponding models and provided probability distributions over the parameter sets of each individual model accordingly. After the success of the first numerical case study, a MDOF structure was considered, for which two competing MDOF models were built, one linear and one nonlinear with a nonlinearity of Duffing type. For this particular case study, the author considered both an increase in degrees of freedom as well as an increase in number of parameters to be identified. The RJMCMC algorithm demonstrated its capabilities by fitting the expected data sets to the corresponding models and providing estimates over the set of parameters to be identified. The last numerical case study considered was in SHM, to do damage detection and identification, an area that still needs addressing in future work. Nonetheless, the RJMCMC method is a capable tool to be further explored in SHM. The next step in the process of the presented work was to build a rig that could allow experiments in order to validate the RJMCMC algorithm on real structures. This proved to be an intensive process and several design ideas were considered before arriving at the one presented in **Chapter 6**. The first case study included characterising the dynamic behaviour of the linear structure. What followed was introducing elements into the rig that would allow it to behave nonlinearly under certain excitation regimes. The RJMCMC algorithm is being used in a Bayesian framework, in part to penalise complexity. This advantage of utilising Bayesian inference came through in the last case study where the method favoured the linear model over the nonlinear model due to the nonlinearity not being strong enough.

It is fair to conclude that the RJMCMC algorithm is a newly introduced powerful tool in doing system identification of dynamical structures and the case studies presented in this thesis have demonstrated aspects of its capabilities and its potential applications.

7.4 Plans for future work

The RJMCMC algorithm still has more to offer. With this in mind, the future work into tapping the potential of this computational tool is already in progress. One of the first issues to be addressed is that of exploring its capabilities when faced with a more diverse set of nonlinear behaviour than that introduced in **Chapter 6**. This will be done in a controlled setting and the results of this ongoing work will be presented in future journal papers. Another aspect that needs to be considered is that of more efficient Markov chains. It is a known problem in the Bayesian community that randomly generated Markov chains can take longer to reach stationarity than ones with a controlled moving pattern. For this reason, many MCMC algorithm are combined with Simulated Annealing, a computational method of creating Markov chains that travel the probable space more effectively. The RJMCMC method itself could benefit from the addition of a Simulated Annealing schedule and more thought and work needs to go into this aspect. In the case of the RJMCMC algorithm, a more efficient acceptance of move from one model to another needs to be dealt with as well, such that the method spends enough time in all spaces of possible models. Furthermore, the MCMC sampling algorithms in general are still in need for automation. The proposals are decided by making educated guesses and that can be very time consuming. Yet another idea is to keep pursing using the RJMCMC algorithm in SHM. The built available rig can be used for a number of damage detection and identification purposes. The starting point would be the simply-supported beam case study considered in **Chapter 5**.

The work done so far has shown that the RJMCMC algorithm is a capable new tool to be used in nonlinear system identification of dynamical structures. The numerical case studies and more importantly the experimental work set the RJMCMC method as a powerful and capable algorithm.

Publications

Conference Proceedings

D. Tiboaca, P.L. Green, R.J. Barthorpe and K. Worden, 'Bayesian System Identification of Dynamical Systems using Reversible Jump Markov Chain Monte Carlo' in 'Topics in Modal Analysis II, Proceedings of the 32nd IMAC , A Conference and Exposition on Structural Dynamics', Vol. 8, pp276 - 284 Springer,2014.

D. Tiboaca, P.L. Green, R.J. Barthorpe and K. Worden, 'Bayesian Parameter Estimation and Model Selection of a Nonlinear Dynamical System using Reversible Jump Markov Chain Monte Carlo' in Proceedings of ISMA2014, International Conference on Noise and Vibration Engineering, Leuven, Belgium, 2014.

D. Tiboaca, R.J. Barthorpe and K. Worden, 'Multiple Damage Identification using the Reversible Jump Markov Chain Monte Carlo' in Proceedings of IWSHM2015, International Workshop on Structural Health Monitoring, Stanford, CA, 2015.

D. Tiboaca, P.L. Green, R.J. Barthorpe and K. Worden, 'Bayesian Inference and RJMCMC in structural dynamics on experimental data ' in 'Model Validation and Uncertainty Quantification, Volume 3, Proceedings of the 34th IMAC , A Conference and Exposition on Structural Dynamics, Springer,2016.

M. Scott, D. Tiboaca, R.J. Barthorpe, K. Worden and D. Wagg, 'On the Validation of Nonlinear MDOF System Models' in Proceedings of ISMA2016 Conference on Noise and Vibration Engineering, Leuven, Belgium, 2016.

Book Chapters

K. Worden, O.D. Tiboaca, I. Antoniadou, R.J. Barthorpe, 'System Identification of an MDOF Experimental Structure with a View Towards Validation and Verification' in 'Topics in Modal Analysis, Volume 10', pp. 57-65, Springer, 2015.

K. Worden, I. Antoniadou, O.D. Tiboaca, G. Manson, R.J. Barthorpe, 'Linear and Nonlinear System Identification using Evolutionary Optimisation' in 'Simulation-Driven Modelling and Optimization', Vol 153, Proceedings in Mathematics and Statistics, pp 325-345, Springer, February 2016.

Appendix

In this section, the derivation of the Laplace method is presented, as it was introduced in **Chapter 3**.

Laplace's Method extended:

If there exists a probability density, $P(x)$, over variables x, its un-normalised density function will be denoted $P^*(x)$. The normalising constant for the un-normalised density function is defined as:

$$Z_P \equiv \int P^*(x)dx \tag{7.1}$$

Let one assume that the probability density has a maximum value or peak at value x_0. Taylor-expanding the natural logarithm of the un-normalised density function around this maximum values gives:

$$\ln(P^*(x)) \simeq \ln(P^*(x_0)) - \frac{c(x-x_0)^2}{2} + ... \tag{7.2}$$

where, $c = -\frac{\partial^2 \ln(P^*(x))}{\partial x^2}|x = x_0$.

Approximating $P^*(x)$ by an un-normalised Gaussian, one gets:

$$Q^*(x) \equiv P^*(x_0)exp(-\frac{c(x-x_0)^2}{2}) \tag{7.3}$$

For the un-normalised Gaussian $Q^*(x)$, the normalising constant will be calculated by:

$$Z_Q = \int Q^*(x)dx \tag{7.4}$$

Replacing $Q^*(x)$ from 7.3 into 7.4 gives:

$$Z_Q = \int P^*(x_0)exp(-\frac{c(x-x_0)^2}{2})dx \tag{7.5}$$

Calculating the integral gives:

$$Z_Q = P^*(x_0)\sqrt{\frac{2\pi}{c}} \tag{7.6}$$

For a space x of a fixed dimension k, the matrix of second order derivatives at the maximum value is of the form:

$$A_{ij} = -\frac{\partial^2 \ln(P^*(x))}{\partial x_i x_j}|x = x_0 \tag{7.7}$$

By Taylor expansion,

$$\ln(P^*(x)) \simeq \ln(P^*(x_0)) - \frac{(x - x_0)^T}{2} A(x - x_0) + ... \tag{7.8}$$

In this case, Z_P becomes:

$$Z_P \simeq Z_Q = P^*(x_0)\frac{1}{\sqrt{\det \frac{A}{2\pi}}} = P^*(x_0)\sqrt{\frac{(2\pi)^k}{\det A}} \tag{7.9}$$

The above approximation is also known in literature as the Saddle-Point Approximation.

***Note: The derivation was followed as explained in [12].**

Bibliography

[1] R. Isermann, M. Munchhof, *Identification of Dynamic Systems - An Introduction with Applications*, Springer, 2011.

[2] J.-N. Juang, *Applied System Identification*, Prentice Hall, 1994.

[3] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann Publishers, 1988.

[4] A. Der Kiureghian, O. Ditlevsen, *Aleatory or epistemic? Does it matter?*, Structural Safety, Vol. 31, Issue 2, pp. 105-112, 2009.

[5] J. Sacks, W.J. Welch, T.J. Mitchell, H.P. Wynn, *Design and Analysis of Computer Experiments*, Statistical Science, Vol. 4, Issue 4, pp. 409-423, 1989.

[6] M.C. Kennedy, A. O'Hagan, *Bayesian Calibration of Computer Models*, Journal of Royal Statistical Society, Vol. 63, Part 3, pp. 425-464, 2001.

[7] W.E. Walker, P. Harremoes, J. Rotmans, J.P. Van Der Sluijs, M.B.A. Van Asselt, P. Janssen, M.P. Krayer Von Krauss, *Defining Uncertainty A Conceptual Basis for Uncertainty Management in Model-Based Decision Support*, Integrated Assessment, Vol. 4, No. 1, pp. 5-17, 2003.

[8] K. Worden, G.R. Tomlinson, *Nonlinearity in Structural Dynamics - Detection, Identification and Modelling*, Institute of Physics Publishing, 2001.

[9] K.Y. Billah, R.H. Scanlan, *Resonance, Tacoma Narrows bridge failure, and undergraduate physics text books* Am. J. Phys., No. 2, Vol. 59, 1991.

[10] NASA/Dryden Flight Research Center, *Helios Prototype So-*

lar Aircraft Lost In Flight Mishap, ScienceDaily, 1 July 2003, (www.sciencedaily.com/releases/2003/06/030630111917.htm).

[11] Z.Q. Chen, X.Y. Wang,J.M. Ko, Y.Q. Ni, B.F. Spencer, G. Yang, *MR damping system on Dongting Lake cable-stayed bridge*, Smart Structures and Materials, 0001, pp.229-235, 2003.

[12] D.J.C. MacKay, *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, 2003.

[13] Xiao-Li Meng, G.L. Jones, A. Gelman, S. Brooks, *Handbook of Markov Chain Monte Carlo*, Chapman and Hall/CRC, 2011.

[14] J.L. Beck, S-K. Au, *Bayesian Updating of Structural Models and Reliability using Markov Chain Monte Carlo Simulation*, Journal of Engineering Mechanics, Vol. 391, pp. 128-380, 2002.

[15] K. Worden, J.J. Hensman, *Parameter estimation and model selection for a class of hysteretic systems using Bayesian inference*, Mechanical Systems and Signal Processing, Vol. 32, pp. 153-169, 2012.

[16] J.L. Beck, *Bayesian updating, model class selection and robust stochastic predictions of structural response*, Proceedings of the 8th International Conference on Structural Dynamics, EURODYN2011, Belgium, July 2011.

[17] M. Muto,J.L. Beck, *Bayesian updating and model class selection for Hysteretic structural models using stochastic simulation*, Journal of Vibration and Control, Vol. 14, pp. 7-34, 2008.

[18] J. Ching, Y. Chen, *Transitional Markov Chain Monte Carlo method for Bayesian model updating, model class selection, and model averaging*, Journal of Engineering Mechanics, Vol. 133, pp. 816-832, 2007.

[19] B.P. Carlin, S. Chib, *Bayesian Model Choice via Markov Chain Monte Carlo Methods*, Journal of the Royal Statistical Society, Vol. 57, Issue 3, pp. 473-484, 1995.

[20] S. Chib, E. Greenberg, *Understanding the Metropolis-Hastings Algorithm*, The American Statistician, Vol. 49, No. 4, November 1995.

[21] M. Banterle, C. Grazian, A. Lee, C.P. Robert, *Accelerating Metropolis-Hastings*

algorithms by Delayed Acceptance, March 2015.

[22] P.L. Green, *Model Selection of a Nonlinear System via Bayesian Inference: An evaluation of Markov Chain Monte Carlo methods*, Proceedings of ICEDyn 2013, International Conference on Structural Dynamics - Sesimbra, Portugal, 17 - 19 June, 2013.

[23] P.L. Green, K. Worden, *Modelling Friction in a Nonlinear Dynamic System via Bayesian Inference*, IMAC XXXI Proceedings, 2013.

[24] P.L. Green, *Bayesian system identification of nonlinear dynamical systems using a fast MCMC algorithm*, Proceedings of 8th European Nonlinear Dynamics Conference, ENOC, Vienna, Austria, 2014.

[25] P.L. Green, *Bayesian system identification of a nonlinear dynamical system using a novel variant of Simulated Annealing*, Journal of Mechanical Systems and Signal Processing, Vol. 52-53, pp. 133-146, 2015.

[26] P.L. Green, E.J. Cross, K. Worden, *Bayesian system identification of dynamical systems using highly informative training data*, Journal of Mechanical Systems and Signal Processing,Vol. 56-57,pp. 109-122, 2014.

[27] P.L. Green, K. Worden, *Bayesian and Markov Chain Monte Carlo methods for identifying nonlinear systems in the presence of uncertainty*, The Royal Society A, Philosophical transactions, Vol. 373, Issue 2051, September 2015.

[28] J.P. Noel, G. Kerschen, *Nonlinear system identification in structural dynamics: 10 more years of progress*, Mechanical Systems and Signal Processing, Vol.83, pp. 2-35, August 2016.

[29] D. Barber, *Bayesian Reasoning and Machine Learning*, Cambridge University Press, 2012.

[30] D. Tiboaca, P.L. Green, R.J. Barthorpe, *Bayesian System Identification of Dynamical Systems using Reversible Jump Markov Chain Monte Carlo*, Proceedings of IMAC XXXII Conference, Orlando, FL, 2014.

[31] S.G. Razul, W.J. Fitzgerald, C. Andrieu, *Bayesian model selection and parameter estimation of nuclear emission spectra using RJMCMC*, Journal Nuclear Instruments and Methods in Physics Research(2002).

[32] J. Guo, R. Jain, P. Yang, R. Fan, C.-K. Kwoh, J. Zheng, *Reliable and Fast Estimation of Recombination Rates by Convergence Diagnosis and Parallel Markov Chain Monte Carlo*, IEEE Transactions on Computational Biology and Bioinformatics, Vol. 13, pp. 1545-5963, 2013.

[33] B. Schutz, *Geometrical Methods of Mathematical Physics*, Cambridge University Press, 1980.

[34] P.J. Green, *Reversible jump Markov Chain Monte Carlo computation and Bayesian model determination*, Biometrika, Vol. 82, pp. 711-732, 1995.

[35] P.J. Green, D.I. Hastie, *Model Choice using Reversible Jump Markov Chain Monte Carlo*, Statistica Neerlandica, pp. 309-338, 2012.

[36] T. Baldacchino, S.R. Anderson, V. Kadirkamanathan, *Computational system identification for Bayesian NARMAX modelling*, Automatica, Vol. 49, pp. 2641-2651, 2013.

[37] I. Ntzoufras, J.J. Forster, P. Dellaportas, *On Bayesian model and variable selection using MCMC*, Statistics and Computing, Vol. 36, pp. 12-27, 2002.

[38] A. Doucet, P.M. Djuric, C. Andrieu, *Model selection by MCMC computation*, Journal of Signal Processing, Vol. 81, pp. 19-37.

[39] A. Doucet, C. Andrieu, *Joint Bayesian Model Selection and Estimation of Noisy Sinusoids via Reversible Jump MCMC*, IEEE Transactions on Signal Processing, 1999.

[40] D. Damen, D. Hogg, *Recognizing Linked Events: Searching the Space of Feasible Explanations*, IEEE Conference on Computer Vision and Pattern Recognition , June 2009.

[41] D. Ustundag, *Recovering Sinusoids from Data using Bayesian Inference with RJMCMC*, IEEE Seventh International Conference on Natural Computation , 2011.

[42] Y.-T. Yeh, L. Yang, M. Watson, N. D. Goodman, P. Hanrahan, *Synthesizing Open Worlds with Contraints using Locally Annealed Reversible Jump MCMC*, Journal ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH, Vol. 31, Issue 4, No.56, 2012.

[43] S. A. Scisson, *Transdimensional Markov Chains: A Decade of Progress and Future Perspectives*, Journal of the American Statistical Association, Vol.100, No. 471, September 2005.

[44] P. J. Green, *Trans-dimensional Markov Chain Monte Carlo*, "Highly Structured Stochastic System", P. J. Green, N. L. Hjort, S. Richardson, Oxford University Press, pp. 179-198, 2003.

[45] S. P. Brooks, P. Giudici, *Convergence Assessment for Reversible Jump MCMC Simulations*, Bayesian Statistics, pp. 000 - 000, Oxford University Press, 1998.

[46] A. Gelman, D. B. Rubin, *Inference from Iterative Simulations Using Multiple Sequences*, Statistical Science, Vol. 7, pp. 457 - 511, 1992.

[47] B. M. Turner, T. V. Zandt, *A tutorial on approximate Bayesian computation*, Journal of Mathematical Psychology, Vol. 56, pp. 69 - 85, 2012.

[48] M. Chiachio, J. L. Beck, J. Chiachio, G. Rus, *Approximate Bayesian Computation by Subset Simulation*, SIAM J. Sci. Comput., Vol. 36, No. 3, pp. A1339 - A1358, 2014.

[49] S.-K. Au, J. L. Beck, *Estimation of small failure probabilities in high dimensions by subset simulation*, Probabilistic Engineering Mechanics, Vol. 16, pp. 263 - 277, 2001.

[50] R.Y. Liang, J. Hu, F. Choy, *Quantitative NDE Techniques for Assessing Damages in Beam Structures*, J. Eng. Mech., Vol. 118, pp. 1468 - 1487, 1992.

[51] Y.-S. Lee, M.-J. Chung, *A study on crack detection using eigenfrequency test data*, Computers and Structures, Vol. 77, pp. 327 - 342, 2000.

[52] S. Dokhe, S. Pimpale, *Effects of crack on modal frequency of cantilever beam*, International Journal of Research in Aeronautical and Mechanical Engineering, Vol. 3, Issue 8, pp. 24 - 38, 2015.

[53] M. Behzad, A. Meghdari, A. Ebrahimi, *A new approach for vibration analysis of a cracked beam*, International Journal of Engineering, Vol. 18, pp. 319 - 330, 2005.

[54] H.-Y. Gao, X.-L. Guo, H.-F. Hu, *Crack identification based on Kriging surrogate model*, Structural Engineering and Mechanics, Vol. 41, No.1, pp. 25 - 41, 2012.

[55] S. Moradi, M. H. Kargozarfard, *On multiple crack detection in beam structures*, Journal of Mechanical Science and Technology, Vol. 27, pp. 47 - 55, 2013.

[56] A. Kumar, J. N. Mahato, *Experimental investigation of crack in aluminum cantilever beam using vibration monitoring technique*, International Journal of Computational Engineering Research, Vol. 4, Issue 4, pp. 39 - 50, 2014.

[57] D. Broda, V. Hiwarkar, V. V. Silberschmidt, W. J. Staszewski, *Effect of crack induced nonlinearity on dynamics of structures: application to structural health monitoring*, Journal of Physics, Series 451, 2013.

[58] E. Simoen, B. Moaveni, J. P. Conte, G. Lombaert, *Uncertainty Quantification in the Assessment of Prigressive Damage in a 7-Story Full-Scale Building Slice*, J. Eng. Mech., Vol. 139, pp. 1818 - 1830, 2013.

[59] I. Behmanesh, B. Moaveni, G. Lombaert, C. Papadimitriou, *Hierarchical Bayesian Model Updating for Probabilistic Damage Identification*, Model Validation and Uncertainty Quantification, Vol. 3, pp. 55 - 66, Springer, 2014.

[60] H. Sun, O. Buyukozturk, *Probabilistic updating of building models using incomplete modal data*, Mechanical Systems and Signal Processing, Vol. 75, pp. 27 - 40, 2016.

[61] Thomas Bayes, *An Essay Towards Solving a Problem in the Doctrine of Chances*, Philosophical Transactions of the Royal Society of London, Vol. 418, pp. 53-370, 1763.

[62] Stephen E. Fienberg, *When did Bayesian Inference become 'Bayesian'*, Bayesian Analysis, Vol. 1, pp. 1-40, 2006.

[63] M. H. Faber, *Statistics and Probability Theory*, Springer, 2012.

[64] M. Das, S. Bhattacharya, *Transdimensional Transformation based Markov Chain Monte Carlo*, arXiv:1403.5207, Cornell University Library, v2, 2014.

[65] S. P. Brooks, P. Giudici, G. O. Roberts, *Efficient construction of reversible jump Markov chain Monte Carlo proposal distributions*, Royal Statistical Society, 65, part 1, pp. 3-55, 2003.

[66] C.R. Farrar, K. Worden, *Structural Health Monitoring: A machine learning perspective*, John Wiley & Sons, Ltd 2013.

[67] P. Congdon, *Bayesian Statistical Modelling*, John Wiley & Sons, Ltd 2001.

[68] M.W. Vanik, J.L. Beck, S.K. Au, *Bayesian Probabilistic Approach to Structural Health Monitoring*, Journal of Engineering Mechanics, Vol. 126, pp. 738-745, July 2000.

[69] A. Coppe, R.T. Haftka, N.H. Kim, *Uncertainty Identification of Damage Growth Parameters using Health Monitoring Data and Nonlinear Regression*, The Annual Conference of the Prognostics and Health Management Society, 2010.

[70] E.Z. Moore, J.M. Nichols, K.D. Murphy, *Model-based SHM: Demonstration of identification of a crack in a thin plate using free vibration data*, Elsevier, Mechanical Systems and Signal Processing, Vol. 29, pp. 284-295, 2011.

[71] E. Zio, A. Zoia, *Parameter Identification in Degradation Modeling by Reversible-Jump Markov Chain Monte Carlo*, Elsevier Journal, Structural Safety, Vol. 58, No. 1, March 2009.

[72] A. Azevedo-Filho, R. D. Shachter, *Laplace's Method Approximations for Probabilistic Inference in Belief Networks with Continuous Variables*, Uncertainty in Artificial Intelligence, San Francisco: Morgan Kaufmann,p.28-36, 1994.

[73] X. Guan, R. Jha, Y. Liu, *Model selection, updating and averaging for probabilistic fatigue damage prognosis*, IEEE Transactions on Reliability Journal, Vol. 33, pp. 242-249, 2011.

[74] H.M. Reed, C.J. Earls, J.M. Nichols, *Stochastic identification of imperfections in a submerged shell structure*, Elsevier Journal, Comput. Methods Appl. Mech. Engrg., Vol. 272, pp. 58-82, 2014.

[75] W.A. Link, M.J. Eaton, *On thinning of chains in MCMC*, Methods in Ecology and Evolution, Vol.3, pp. 112-115, 2012.

[76] R.M. Neal, *Probabilistic Inference using Markov Chain Monte Carlo methods*, Technical Report, CRG-TR-93-1, Dept. of Computer Science, University of Toronto, 1993.

[77] C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer Science + Business Media,Inc, 2006.

[78] J. Berger, M. Clyde, G. Molina, R. Paulo, F. Liang, *Mixtures of g-priors for Bayesian variable selection*, Journal of the American Statistical Association, Vol. 103(481), pp. 410-423, 2008.

[79] *The MathWorks 2012-2015.Matlab R2012a, R2015b.*, 2012-2015.

[80] E.L. Lehmann, G. Casella, *Theory of Point Estimation*, Springer, 2nd Edition, New York, 1998.

[81] A.K. Qin, P.N. Suganthan, *Self-adaptive Differential Evolution Algorithm for Numerical Optimization*, IEEE Xplore Journal, Vol. 3, IEEE Congres on Evolutionary Computation 2005.