Thea Martine Marstrander

Investigation of the Burridge-Knopoff-Pad Model with a Focus on Eigenmodes and Rayleigh Damping

Master's thesis in Engineering and ICT
Supervisor: Bjørn Haugen and Astrid de Wijn
March 2021

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

NTNU
Norwegian University of
Science and Technology

Thea Martine Marstrander

Investigation of the Burridge-Knopoff-Pad Model with a Focus on Eigenmodes and Rayleigh Damping

Master's thesis in Engineering and ICT
Supervisor: Bjørn Haugen and Astrid de Wijn
March 2021

Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

NTNU
Norwegian University of
Science and Technology

Preface

This thesis was carried out during fall 2020 and winter 2021 and concludes my degree Master of Science in Engineering and ICT at the NTNU. I would like to express my gratitude to my two supervisors Astrid de Wijn and Bjørn Haugen for their valuable guidance. I am grateful for their engagement towards the project, their commitment to meetings, and that they have freely shared relevant knowledge and new ideas throughout the project. I would also like to give a special thanks to Sander Sundt and Ådne Viga for helping me with the thesis.

Thea Martine Marstrander
Trondheim, February 2021

Abstract

The Burridge-Knopoff-Pad model is a novel and largely unstudied model made to represent the squealing noise from car brakes due to friction-induced vibrations. The model combines the Burridge Knopoff model used to study earthquakes and a single-degree-of-freedom model. The Burridge-Knopoff-Pad model consists of a pad and several blocks, where the blocks are used as a discretization of the elastic brake discs in a physical braking system to represent their natural vibrations. This master's thesis aims to investigate the Burridge-Knopoff-Pad model with a special focus on eigenmodes and Rayleigh damping. This is done by simulating the model using C++ and visualizing the simulation results through plots made in Python. A method has been developed to visualize which normal modes of the model are activated throughout simulations. The thesis also studies the number of degrees of freedom in the model and compares it to the Burridge-Knopoff and single-degree-of-freedom model to get a better understanding of what the extra degree of freedom does to the Burridge-Knopoff-Pad model.

The results show that by changing the external slider speed that acts on the system, the model transitions from periodic to chaotic behaviour, which is consistent with research suggesting that the response from brake squeal can be treated as chaotic. Comparing the Burridge-Knopoff-Pad and the Burridge-Knopoff model with 100 blocks, the models show the same general behaviour, but when decreasing the degrees of freedom, the models show considerable differences and it becomes evident that the pad in the Burridge-Knopoff-Pad model damps the system. The thesis concludes that more than two degrees of freedom are required in the Burridge-Knopoff-Pad model to represent brake squeal and that convergence of the solution is achieved for approximately 40 degrees of freedom. The introducing of Rayleigh damping is found to lead to both theoretical explainable and several unexplained behaviours, verifying other research stating the importance of damping in the modeling of friction-induced vibration systems. This thesis study shows many interesting behaviours in the model, concluding that the model has the potential to represent brake squeal, indicating that further work is needed.

Sammendrag

Burridge-Knopoff-Pad-modellen er en nyutviklet og lite studert modell laget for å representere den skjærende lyden fra bilbremser grunnet friksjonsinduserte vibrasjoner. Modellen kombinerer Burridge-Knopoff-modellen som brukes til å studere jordskjelv med et enkeltfrihetgradsystem. Burridge-Knopoff-Pad-modellen består av en stor masse og flere blokker, hvorav blokkene er brukt som en diskretisering av de elastiske bremseskivene i et fysisk bremsesystem for å representere deres naturlige vibrasjoner. Denne masteroppgaven tar sikte på å undersøke Burridge-Knopoff-Pad-modellen med spesielt fokus på egenmoder og Rayleigh-demping. Dette er gjort ved å simulere modellen i C++ og visualisere simuleringsresultatene gjennom plott laget i Python. En metode er utviklet for å visualisere hvilke normalmoder i modellen som er aktive gjennom simuleringer. Oppgaven studerer også antall frihetsgrader i modellen og sammenligner den med Burridge-Knopoff modellen og enkeltfrihetgradsystemet for å få en bedre forståelse av hva den ekstra frihetsgraden gjør med Burridge-Knopoff-Pad-modellen.

Resultatene viser at ved å endre den eksterne glideshastigheten som virker på systemet, beveger modellen seg fra periodisk til kaotisk oppførsel, noe som verifiserer annen forskning som indikerer at responsen til skjærende bremselyder kan betraktes som kaotisk. Oppgaven konkluderer med at mer enn to grader av frihet kreves i Burridge-Knopoff-Pad-modellen for å representere skjærende bremselyder, og at modellen konvergerer ved omtrent 40 frihetsgrader. Når man sammenligner Burridge-Knopoff-Pad og Burridge-Knopoff-modellen med 100 blokker, vises den samme generelle oppførselen i begge modellene, men når man reduserer antall frihetsgrader, viser modellene betydelige forskjeller, og det blir tydelig at den store massen i Burridge-Knopoff-Pad-modellen demper systemet. Innføringen av Rayleigh demping viser seg å føre til både teoretisk forklarlige og flere uforklarte atferder, hvilket verifiserer annen forskning som diskuterer viktigheten av demping i friksjonsinduserte bremsesystemer. Oppgaven viser flere interessante oppførelser i modellen og konkluderer med at modellen har potensial til å representere skjærende bremselyder, noe som indikerer at det er behov for videre arbeid.

Contents

List of Tables

List of Figures

List of Symbols

Nomenclature

μ	Kinetic friction coefficient
μ_s	Static friction coefficient
ω	Angular frequency
ω^2	Eigenvalue
E	Energy
F	Force by friction
f	Frequency
v	Slider velocity
$\dot{\boldsymbol{e}}$	Modal contribution to velocity
$\boldsymbol{\Phi}$	Mode shape matrix
$\boldsymbol{\phi}$	Mode shape
$\boldsymbol{e}$	Modal contribution to displacement
$\boldsymbol{I}$	Identity Matrix
$\boldsymbol{K}$	Stiffness Matrix
$\boldsymbol{M}$	Mass Matrix
Δt	Time Step

δ	Stiffness proportional damping factor
η	Mass proportional damping factor
λ	Eigenvalue, general
h	Step size, general
i	Block number
j	Step number, numerical methods
t	Time
a	Initial distance between blocks
m_u	Block mass
m_x	Pad mass
u	Block position
x	Pad position
σ	Friction law scaling factor
ζ	Damping ratio
c	Damping coefficient, general
c_c	Critical damping coefficient
c_p	Pad damping coefficient
F_0	Maximum static friction force
k_c	Neighbouring spring constant
k_{p0}	Top plate-pad spring constant
k_p	Pulling spring constant
N	Number of blocks
n	Degrees of freedom in a general MDOF system
p	Applied load
s_c	Neighbouring spring scaling factor

s_m	Block mass scaling factor

Abbreviation

NTNU	Norwegian University of Science and Technology
BK	Burridge-Knopoff
BKP	Burridge-Knopoff-Pad
DOF	Degree-of-freedom
MDOF	Multi degree-of-freedom
SDOF	Single degree-of-freedom
CSV	Comma-separated values
MD	Markedown
VS	Visual Studio
YAML	YAML Ain't Markup Language

Chapter I

Introduction

1.1 Motivation

Many sound waves are produced by frictional motion. An example is the squealing noise from car brakes, which occurs due to the vibrations induced by the frictional braking force. Friction induced vibrations in braking systems is a problem in the industry, as it can cause severe damages on machinery (Jean-Jacques Sinou (2007)). Although friction and friction-induced vibrations is a well-studied field, it is complex, and to this date, there is no accurate model to represent friction-induced vibrations.

In 2017, Astrid De Wijn and Bjørn Haugen at NTNU suggested merging a model frequently used in earthquake analysis, the *Burridge-Knopoff model*, with a single-degree-of-freedom model often used to represent car brakes, to make an improved model to represent the squealing noise from car brakes. The result is a set of second-order differential equations with the received name the *Burridge-Knopoff-Pad model*. The model consists of a pad and several blocks acted upon by a moving substrate. The blocks are used to discretize the elastic brake discs in a physical braking system to represent their natural vibrations.

Three master students, Ferre (2018), Standnes (2019) and Høgberg (2019), have previously studied the Burridge-Knopoff-Pad model. They have simulated the model and analyzed the results, mostly regarding how position and velocity change over time with different initial conditions. The previous students have found that the model has the potential to represent friction-induced vibrations and concludes that more work is needed to further assess the model, which is the starting point for this thesis.

1.2 Goal and Objectives

The project's overall goal is to explore if it is possible to use the Burridge-Knopoff-Pad model to obtain an improved way to represent friction and friction-induced vibrations. This thesis combines previous work with new analysis to get more familiar with the model

and get closer to the project's overall goal. The main goal of working with this thesis has been to bring two new perspectives to the study of the model. The first is to look at which normal modes are present in the model at different behavioural regions. The second new perspective is to introduce *Rayleigh damping*, to see how it affects the system. Additionally, a study of the number of blocks and a comparison to the Burridge-Knopoff and SDOF model will be used to get more familiar with the model and thus get closer to the overall goal.

As three other students have previously worked on the project, the first objective is to grasp the state of the project. Additionally, many researchers have worked with the Burridge-Knopoff model. To look into their work is also one of the objectives. In order to bring in the perspective of normal modes and damping, knowledge and theory in structural dynamics need to be adapted, and it needs to be applied to the Burridge-Knopoff-Pad-model.

The previous students have written code used for simulations in the programming language C++. Learning the basis of C++ and use it to run different simulations with different initial conditions will be a large part of the project. Using C++ to implement new functionalities that allow studying normal modes and damping is also necessary. An objective is also to create a well-documented, easy-to-read, and easy-to-run code, allowing for further work with the project. As the C++ simulation outputs raw data, Python code will be written to analyze the data from the simulations through plots.

1.3 Outline

The thesis starts with providing the theoretical framework used in this thesis. It will then describe relevant previous work done by the master students Ferre (2018) and Standnes (2019). This includes introducing the Burridge-Knopoff-Pad model and summarizing useful results obtained by the previous students, ensuring that the reader does not need to read the former theses to understand the content of thesis. The methods used are split into two chapters - Implementation Tools and Analysis Methods. The chapter Implementation Tools describes the programming languages used, the structure of the code, documentation, and how the model is built up in terms of inputs and outputs. The chapter Analysis Methods starts by introducing the general behaviour of the model before describing how the theory is used to implement an eigenmode analysis and Rayleigh damping. Further, Chapter 6 will present the results, Chapter 7 will discuss the results, and Chapter 8 provides a conclusion and suggestions for further work.

1.4 Related Work

As will be shown, the response of the Burridge-Knopoff and the Burridge-Knopoff-Pad models share similarities. Standnes (2019) demonstrated several common behavioural patterns, and it can be assumed that interesting aspects and papers on the Burridge-Knopoff

model can be used as inspiration for analysis of the Burridge-Knopoff-Pad model.

This thesis is a continuation of work done by Ferre (2018) and Standnes (2019), and their work has been actively used throughout this paper. Their methods in terms of programming were adopted and extended to allow for additional functionalities. The work done by the master student Høgberg (2019) dealt with fretting fatigue, which is less relevant for this thesis, and is thus not used extensively in this paper.

The Burridge-Knopoff model was originally constructed by Burridge and Knopoff (1967). The model was further analyzed and modified by Carlson and Langer (1989), in which version is used in this thesis. The model has later been analyzed in a large variety of articles, desertions, and books. Papers by Mascia and Moschetta (2020) and Ferguson, Klein and Rundle (1998) are used as they provide a more straightforward understanding of the model and how it relates to earthquakes.

Many researchers have come up with models to represent friction-induced vibration. Papangelo et al. (2016) and (2017) wrote two papers on snaking bifurcation where they presented a model with apparent similarities to the Burridge-Knopoff model. Despite the similarities, the BK model is not mentioned. This might be because snaking bifurcation is a different science area, and the authors were not aware of the Burridge-Knockoff model.

A common characteristic of many suggested models to represent friction-induced vibrations is that there are instabilities in the models (Jean-Jacques Sinou (2007)). Sinou published the paper *Mode coupling instability in friction-induced vibrations and its dependency on system parameters including damping*. Here, Jean-Jacques Sinou (2007) states that even though several researchers over the years have studied friction-induced vibrations, no method reduces instabilities. The paper further explains a stability analysis (Hopf bifurcation point) using a model to represent friction-induced vibrations. It concludes by emphasizing the considerable importance of damping in stability analysis and that it can lead to a misunderstanding of the mode coupling instability of mechanical systems. Wernitza and Hoffmannb (2016) also shows that intermittency and multiscale behaviour is dominant in friction-induced vibration for braking systems.

In the paper *Nonlinear Dynamics and Control in an Automotive Brake System*, Chang and Hu (2016) use an 2-DOF model to study brake squeal. As Jean-Jacques Sinou (2007), they use different methods to look into the transition to chaotic behaviour for the model they use, concluding that their experiments have indicated that the response of brake squeal can be treated as chaotic. In order to study chaotic behaviour they use a bifurcation diagram, phase portraits, a Poincaré map, frequency spectra, and Lyapunov exponents. The latter concepts are different ways to study chaotic motion (Artuso et al. (2020)). As Jean-Jacques Sinou (2007), Chang and Hu (2016) also notes the importance of damping in their brake squeal model.

Carlson and Langer (1989) argue that the velocity weakening friction law they use in the Burridge-Knopoff model is responsible for amplifying small heterogeneities in the initial spatial distribution, which leads to chaotic motion. Erickson, Birnir and Lavallée (2010) simulates the Burridge-Knopoff model with a different number of blocks and concludes that, with the parameters used in the study, the model has a transition to chaotic behaviour at $N > 21$ blocks. In the same paper, Erickson, Birnir and Lavallée (2010) argue that the friction law may be responsible for causing small instabilities to grow large finite events, similar to the conclusion of Carlson and Langer (1989).

Chapter II

Theoretical Framework

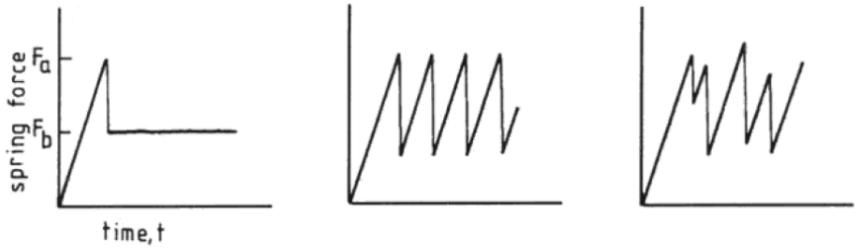
This chapter will provide the theoretical framework necessary to understand the model itself, its implementation for numerical simulations, and the methods used to analyze the model. The first four sections, Sliding Friction, The Burridge-Knopoff model, A Single-Degree-of-Freedom system, and Numerical Simulations, are relevant to understand the development and numerical implementation of the new model. The following section, Structural Dynamics, will provide theory in the area of structural dynamics needed to understand the study of normal modes and Rayleigh damping.

2.1 Sliding Friction

The study of friction is one of the oldest, most common, and most fundamental problems in physics (Persson (1998)). It is usually introduced and studied in a simplified manner in an early stage of the first physics classes. However, Persson (1998) states that the behaviours of friction are complicated and not well understood.

One can define sliding friction as a force between surfaces in contact that opposes relative motion between the surfaces (Persson (1998)). The forces between the two bodies are ultimately due to electromagnetic forces between the solids' particles. Thus, to describe the exact interaction between two solids, one would need to incorporate all electrons and nuclei's coupling using microscopic equations. Even though practically all macroscopic solids are asperate from a microscopic point of view, microscopic studies of a contact surface of macroscopic size would include a large number of atoms which is not feasible, nor meaningful, to simulate. *Friction* is a substitute for such a microscopic analysis.

More than 500 years ago, Leonardo Da Vinci defined friction to be proportional to the normal force (Freedman (1998)). His results are still used today, and sliding friction is often defined in terms of the coefficient of friction μ, which is given as F/N, where F is the force by friction working opposite to the direction of motion, and N is the force acting normal to the surface (Freedman (1998)). One also often defines static friction μ_s, as the



(a) Steady sliding. **(b)** Periodic stick-slip motion **(c)** Chaotic motion

Figure 2.1: The force acting on a block in three different sliding motion scenarios (Persson (1998)).

friction coefficient in the case where the objects have a static relative motion.

2.1.1 The Stick-Slip Phenomenon

The stick-slip phenomenon describes a motion that can occur when two objects are sliding opposite to each other with a nonzero relative motion (Persson (1998)). One of the fundamental problems in the study of friction is discovering the origin of the microscopic stick-slip motion. It is believed that the phenomenon is caused by the fact that the static friction coefficient is usually higher than the kinetic friction between two surfaces.

Imagine an object laying on a plate and acted on by an external force in the direction parallel to the plate. When the force by motion overcomes the force by static friction, there will be an instant reduction of the absolute value of the total force by friction, and the object will start to move (Persson (1998)). Suppose then that the force from motion becomes insufficient to overcome the kinetic friction. In that case, the object's motion will decelerate to a state of rest by the static friction, and the phenomenon can repeat.

Figure 2.1 from Persson (1998) illustrates different developments of friction over a period of time for an object laying on a frictional plate and acted upon by a spring force. Figure 2.1(b) illustrates how the friction force changes during periodic occurrences of the stick-slip motion. As a comparison, Figure 2.1(a) and (c) show how the force changes during steady sliding and chaotic motion, respectively.

2.1.2 Stick-Slip Friction Law

Coulomb's law states that kinetic friction is almost independent of velocity (Freedman (1998)). However, when considering stick-slip, the stick-slip friction law can be used (Carlson and Langer (1989)). This law describes how the force F on a sliding element is dependent on the velocity $\dot{x}$ and decreases towards zero as $|\dot{x}|$ increases. This is illustrated in Figure 2.2, where it is also illustrated how $F(\dot{x})$ varies between $\pm F_0$ as $\dot{x}$ is equal to zero.

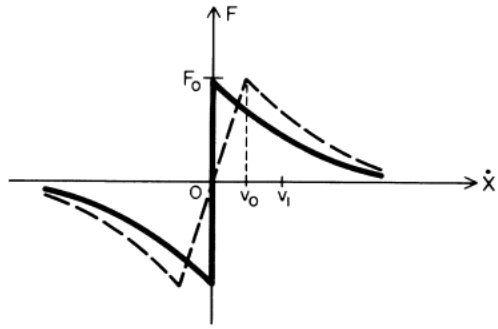


Figure 2.2: Illustration of the stick-slip friction law described by Carlson and Langer (1989).

2.2 The Burridge-Knopoff Model

The Burridge-Knopoff (BK) model is a set of differential equations originally used to model the physics of earthquakes, suggesting that the tectonic plates are in contact only at discrete points (Burridge and Knopoff (1967)). Earthquakes can happen at different kinds of faults, but this model focuses on earthquakes occurring due to stick-slip faults. These earthquakes have their initial rapture close to the earth's surface, a relatively rigid layer consisting of the crust and upper mantle (Mascia and Moschetta (2020)). The BK model approaches the problem of stick-slip faults between two surfaces by letting two parallel surfaces move relative to one another with a set of blocks in between, representing the contact area between the surfaces.

The version of the model used in this project is developed by Carlson and Langer (1989) and has slight adjustments from the original model. The model is illustrated in Figure 2.3. It consists of n massive blocks of mass m all connected via springs with a spring constant k_c following Hooke's law. The blocks are in contact with a straight, frictional, moving substrate, a *slider plate*, on one side and connected to a stationary plate via springs of spring constant k_p on the opposite side. "The spring constants k_c and k_p describe the linear elastic response of the contact region to compression and shear, respectively (Carlson and Langer (1989))." As the slider plate moves with velocity, v, the blocks experience a frictional force F, in the opposite direction of the slider plate motion.

Imagine the blocks are initially at rest. As the loader plate starts to move, the static friction increases to the point where the friction force will decrease as the velocity of the blocks deviates from zero (Ferguson, Klein and Rundle (1998)). The force will now depend on the kinetic friction coefficient instead of the static, as explained in Section 2.1.1. The friction force will then increase to the point where the blocks slide to a lower residual force state

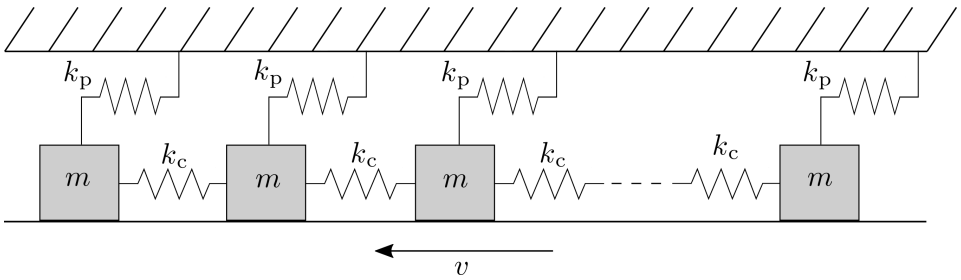


Figure 2.3: Illustration of the BK model. The blocks of mass m are connected to each other via springs with spring constant k_c and in contact with a moving frictional substrate. The blocks are also connected to a stationary plate via springs with spring constant k_p. (Recreation of the model by Carlson and Langer (1989))

and decrease their velocity to the state of rest. To produce this dynamic instability, the BK model uses a velocity-dependent frictional force, such as the stick-slip friction law explained, that will decrease as the sliding block velocity increase (Ferguson, Klein and Rundle (1998)).

The BK model is described using Newton's second law of motion, which states that force is equal to mass times acceleration, and Hooke's law, which says that force is equal to Hooke's constant times the relative displacement. The model for one block is given in Equation 2.1, where the equation is written as in the master thesis of Standnes (2019). The mass of block i is given as m_i, $x_i(t)$ is the displacement at time t, the dot notation represents the time derivative and v is the velocity of the slider plate.

$$m_i\ddot{x}_i(t) = k_c[x_{i+1}(t) - 2x_i(t) + x_{i-1}(t)] - k_p x_i(t) - F(v + \dot{x}_i(t)) \tag{2.1}$$

As seen in the first term of Equation 2.1, a sliding block i is dependent on its two neighboring blocks, $i-1$ and $i+1$. If any of the neighboring blocks receives enough force to cause them to slip, it can result in the slip of block i. Imagining that this happens to enough blocks, a chain reaction of failure can occur. Going back to the origin of the model, this is when an earthquake would occur (Mascia and Moschetta (2020)).

2.3 A Single-Degree-of-Freedom System

A single-degree-of-freedom (SDOF) oscillator system is a system used to simplify many different motion systems (Clough and Penzien (1995)). Single-degree-of-freedom means that only a single variable is needed to describe the system's motion. Figure 2.4 illustrates one of many ways to represent an SDOF system, where a block, or pad, of mass m is connected to a spring, a damper, and an external friction force in terms of a moving plate,

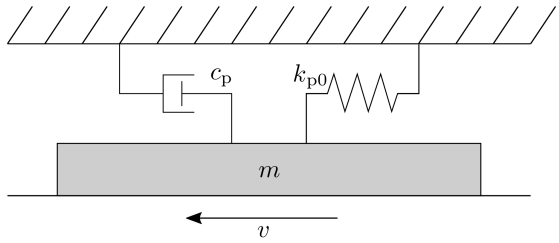


Figure 2.4: Representation of a single-degree-of freedom system, including a massive block, or pad, with mass m, a damper with damping constant c_p, a spring with spring constant k_{p0} and a moving plate with velocity v.

moving with velocity v. The motion of the system can be described by the second order differential equation (Clough and Penzien (1995)):

$$m\ddot{x} = -c_p\dot{x} - k_{p0}x - f(\dot{x} + v), \tag{2.2}$$

with initial conditions

$$\dot{x}(0) = \dot{x}_0, \quad x(0) = x_0, \tag{2.3}$$

where x denotes the mass position, c_p is the damping constant, k_{p0} is the spring coefficient and $f(\dot{x} + v)$ is the force caused by friction.

2.4 Numerical Simulations

Many engineering problems involve complex differential equations that do not have an exact analytic solution. To approximate a solution to such equations, they can be solved numerically. In order to solve Equation 2.1, Burridge and Knopoff (1967) suggested using the numerical method 2nd order Runga-Kutta. The following section will review both Euler's method and the 2nd order Runga-Kutta based on the compendium by Hellevik (2020).

A second-order differential equation is an equation where the highest order of the derivative is of the second order, such as Equation 2.1. In an *ordinary* differential equation, the term ordinary indicates that the derivatives are only with respect to one independent variable. This is opposed to *partial* differential equation, where the derivatives can be with respect to more than one independent variable.

2.4.1 Euler's Method / First Order Runge-Kutta

One of the simplest ways to understand numerical methods is through Euler's method, or the first order Runge-Kutta method. Consider the initial value problem:

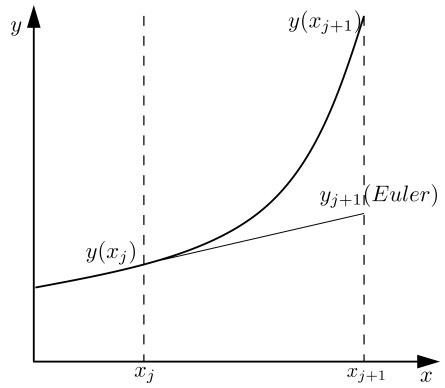


Figure 2.5: Illustration of how Euler's method approximate the function y.

$$\begin{aligned} \frac{dy}{dx} &= f(x, y), \\ y(x_0) &= y_0, \end{aligned} \tag{2.4}$$

where y is the unknown function of x to approximate. Picking a step size $h > 0$, Euler's approximation of the analytical value $y(x_{j+1})$ can be calculated as:

$$y_{j+1} = y_j + hf(x_j, y_j), \tag{2.5}$$

where j is the step number. Equation 2.5 states that the next step, y_{j+1} is the sum of the previous step, y_j, and the step size times the derivative in the previous step $hf(x_j, y_j)$. Each step of the equation is analogous to calculating the displacement of an object moving with constant velocity over the time step, $y = y_0 + tv_0$. Euler's method is illustrated in Figure 2.5. It can be shown using the Taylor's series, that Euler's method has a single step error of $\mathcal{O}(h^2)$ and hence the error for the whole approximation is $\mathcal{O}(h)$.

2.4.2 2^{nd} Order Runge Kutta / Midpoint Method

The 2^{nd} order Runge Kutta, or the explicit midpoint method, is another numerical method. Again, consider the initial value problem described in Equation 2.4. Choosing the step size $h > 0$, the midpoint method can be defined as:

$$y_{j+1} = y_j + hf(x_j + \frac{h}{2}, y_j + \frac{h}{2} f(x_j, y_j)). \tag{2.6}$$

The midpoint method and its relation to Euler's method is illustrated in Figure 2.6. The method expands Euler's method by adding a midpoint between the steps used in Euler's

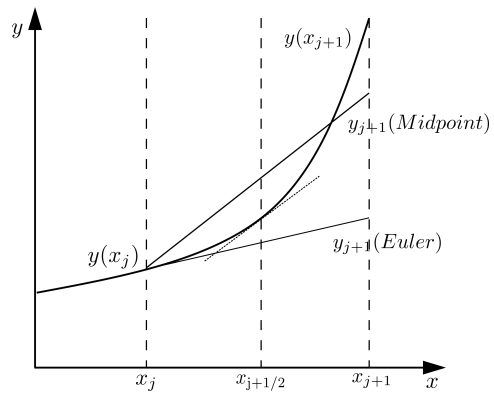


Figure 2.6: Illustration of the midpoint method and its relation to Euler's method.

method to increase the accuracy and stability of the model. The midpoint method is a second order approximation, hence the error is $\mathcal{O}(h^3)$ for each step and $\mathcal{O}(h^2)$ for the full approximation. Regarding stability, Hellevik (2020) shows by the numerical amplification factor that the midpoint method has a larger stability range than Euler's method.

2.4.3 Second Order Equations in Vector Form

In order to solve a second-order differential equation such as the BK equation, the initial value problem stated in Equation 2.4 needs to be modified to yield for second-order derivatives as well. This is solved by writing the function to approximate in vector form:

$$\vec{y} = \begin{bmatrix} y \\ \dot{y} \end{bmatrix}, \quad \dot{\vec{y}} = \frac{d}{dx}(\vec{y}) \begin{bmatrix} \dot{y} \\ \ddot{y} \end{bmatrix}. \tag{2.7}$$

2.5 Structural Dynamics

2.5.1 The General Eigenvalue Problem

The problem $\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$ is an eigenvalue problem if it has a non-trivial solution (Lay, Lay and McDonald (2016)). In that case, the scalar λ is known as the eigenvalue of the system, and the nonzero vector $\boldsymbol{x}$ is the corresponding eigenvector for the matrix $\boldsymbol{A}$.

To solve the eigenvalue problem, the problem can be rewritten to $(\boldsymbol{A}-\lambda)\boldsymbol{x} = 0$ and solved by finding the eigenvalue λ which satisfies the characteristic equation for $\boldsymbol{A}$,
$\det(\boldsymbol{A} - \lambda\boldsymbol{I}) = 0$, where $\boldsymbol{I}$ is the identity matrix. Gaussian Elimination can be used to find the eigenvector $\boldsymbol{x}$ (Lay, Lay and McDonald (2016)). Note that an $n \times n$ matrix

will have a characteristic polynomial of order n and will hence have a maximum of n unique solutions, meaning it can have a maximum of n possible unique eigenvalues and corresponding eigenvectors.

2.5.2 Equation of Motion of a Simple Dynamic System

Consider the SDOF system in Section 2.3, with an applied load $p(t)$ substituting the moving loader plate. In accordance with d'Alembert's principle this can be written as (e.g. Clough and Penzien (1995), Chopra (2017)):

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = p(t). \tag{2.8}$$

From Hooke's law, the angular frequency is expressed as $\omega = \sqrt{k/m}$ and the frequency can be expressed as:

$$f = \frac{\omega}{2\pi} = \frac{1}{2\pi}\sqrt{\frac{k}{m}}. \tag{2.9}$$

From the angular frequency and the damping ratio, which is introduced as $\zeta = c/2m\omega$, Equation 2.8 can be rewritten as (Chopra (2017))

$$\ddot{x}(t) + 2\zeta\omega\dot{x}(t) + \omega^2 x(t) = p(t)/m. \tag{2.10}$$

By setting $p(t) = 0$, a second order homogeneous differential equation which governs the free vibration response of the SDOF-system is obtained. Inserting the solution $x = e^{st}$ gives

$$(s^2 + 2\zeta\omega s + \omega^2)e^{st} = 0, \tag{2.11}$$

where s is a constant (Kreyszig (2011)). The solution to the term in the parenthesis, also known as the *characteristic equation*, can be inserted into the general solution of Equation 2.10, $x(t) = a_1 e^{s_1 t} + a_2 e^{s_2 t}$ (Chopra (2017)). This gives

$$x(t) = e^{-\zeta\omega t}(a_1 e^{i\omega_D t} + a_2 e^{-i\omega_D t}), \tag{2.12}$$

with the complex valued constants a_1 and a_2, and the damped angular frequency $\omega_D = \omega\sqrt{1-\zeta^2}$. By the use of Euler's formula the exponential terms are transformed into trigonometric terms, giving

$$x(t) = e^{-\zeta\omega t}(A\cos(\omega_D t) + B\sin(\omega_D t)), \tag{2.13}$$

with the constants A and B determined by the initial conditions $A = x(0)$ and $B = (\dot{x}(0) + \zeta\omega x(0))/\omega_D$. For the undamped case with $\zeta = 0$ Equation 2.13 is reduced to

$$x(t) = A\cos(\omega t) + B\sin(\omega t) \tag{2.14}$$

with $A = x(0)$ and $B = \dot{x}(0)/\omega$ (Chopra (2017)).

2.5.3 Normal Modes

The motion pattern of an oscillating system in which all parts of the system moves sinusoidally with the same fixed phase and frequency is known as a normal mode of the system. Common names for these fixed frequencies of the normal modes are the system's resonant or natural frequencies. In a linear oscillatory mass spring system, the eigenvalues and eigenvectors give the normal modes of the system (Clough and Penzien (1995)).

The study of normal modes generally deals with several masses (Morin (2007)). Consider a multi degree-of-freedom system (MDOF) with n masses, expressed as an expansion of the homogeneous form of Equation 2.8

$$\sum_{i=1}^{n} m_i\ddot{x}_i + c_i\dot{x}_i + k_i x_i = 0 \quad i = 1, 2, \ldots, n. \tag{2.15}$$

Equation 2.15 expresses a free motion system with n degrees of freedom (Bergan, Larsen and Mollestad (1986), Clough and Penzien (1995)). Equation 2.15 can be expressed on a more compact form using matrices:

$$\boldsymbol{M}\ddot{\boldsymbol{x}} + \boldsymbol{C}\dot{\boldsymbol{x}} + \boldsymbol{K}\boldsymbol{x} = 0, \tag{2.16}$$

where $\boldsymbol{M}$ is the mass matrix, $\boldsymbol{C}$ as the damping matrix, $\boldsymbol{K}$ as the stiffness or spring constant matrix, and $\boldsymbol{x}$ is the displacement vector, and where a dot denotes a time derivative. In the undamped case the damping matrix $\boldsymbol{C}$ can be omitted to obtain

$$\boldsymbol{M}\ddot{\boldsymbol{x}} + \boldsymbol{K}\boldsymbol{x} = 0. \tag{2.17}$$

A system's response vibrating its i^{th} normal mode can be by split into a deflection shape $\boldsymbol{\phi}_i$ and the time variation of the deflections given by the harmonic function $q_i(t)$. Thus the system's response can be given as

$$x(t) = q_i(t)\phi_i, \tag{2.18}$$

where $\boldsymbol{\phi}_i$ is a vector with length equal to the number of degrees of freedom in the model, and gives the direction and amplitude of the deflection relative to that of the other DOFs. In accordance with Equation 2.14, the time variation is given as $q(t) = A_i \cos(\omega_i t) + B_i \sin(\omega_i t)$, with A_i, B_i and ω_i given as in Equation 2.14. Inserting $q(t)$ and Equation 2.18 into Equation 2.17 gives

$$(-\omega_i^2 \boldsymbol{M}\boldsymbol{\phi}_i + \boldsymbol{K}\boldsymbol{\phi}_i)q_i(t) = \mathbf{0}. \tag{2.19}$$

A solution to this equation is $q_i(t) = 0$, but as this term describes the time variation of the normal mode it is of little interest because it implies no motion. This is also know as a trivial solution. Non-trivial solutions can be found by considering the term in the parenthesis. Rewriting the term gives

$$(\boldsymbol{K} - \omega_i^2 \boldsymbol{M})\boldsymbol{\phi}_i = \mathbf{0}. \tag{2.20}$$

Having $i = 1, 2, \ldots, n$, Equation 2.20 can be be recognised as a set of n linear eigenvalue problems (Géradin and Rixen (1997)). Solving for ω_i^2 by setting the determinant of the parenthesis term equal to zero, $\det(\boldsymbol{K} - \omega_i^2 \boldsymbol{M}) = 0$, non-trivial solutions can be found. For a known normal frequency ω_i its corresponding deflection shape, mode shape, normal vector or eigenvector $\boldsymbol{\phi}_i$ can be calculated by Equation 2.20.

The n frequencies ω_i are usually sorted from smallest to largest as

$$\omega_1^2, \omega_2^2, \omega_3^2, \ldots, \omega_n^2, \tag{2.21}$$

with corresponding eigenvectors

$$\boldsymbol{\phi_1}, \boldsymbol{\phi_2}, \boldsymbol{\phi_3}, \ldots, \boldsymbol{\phi_n}. \tag{2.22}$$

The vector consisting of all the frequencies is called the frequency vector $\boldsymbol{\omega}$. It can be shown that for a real, symmetric, and positive definite mass and stiffness matrix, all roots of the eigenvalues will be real and positive (Clough and Penzien (1995)).

Orthogonality Conditions

It can be shown that the deviation of the natural modal shapes leads to orthogonal properties of the mass and stiffness matrices (Bergan, Larsen and Mollestad (1986), Chopra (2017)). The orthogonal properties are given as

$$\boldsymbol{\phi}^{\mathrm{T}} \boldsymbol{M} \boldsymbol{\phi} = \boldsymbol{I}, \quad \boldsymbol{\phi}^{\mathrm{T}} \boldsymbol{K} \boldsymbol{\phi} = \Omega^2, \tag{2.23}$$

where $\boldsymbol{I}$ is the identity matrix and $\boldsymbol{\Omega}^2$ is the diagonal matrix with ${\omega_i}^2$ on the diagonal.

2.5.4 Damped Free Vibrations

If damping is present in the system, such as in the SDOF system described above, the solution to Equation 2.11 can be expressed as (Craig and Kurdila (1981))

$$s_{1,2} = -\frac{1}{2m} \pm \sqrt{(\frac{c}{2m})^2 - \omega^2}. \tag{2.24}$$

The solution describes three different types of motion, determined by whether the value under the root sign, the radical term, is positive, negative, or zero. The three types of motion are known as overdamped, underdamped, and critically damped, respectively. Consider first the critically damped case where the radical term is zero so that $c/2m = \omega$. In this case, the damping coefficient c is known as the critical damping coefficient $c_c = 2m\omega = 2\sqrt{mk}$. A critically damped system will move asymptotically towards equilibrium with no oscillations around this point.

The ratio the given damping has to its critical value can now be introduced as $\zeta = c/c_c$. An underdamped system is described by $\zeta < 1$, critically damped by $\zeta = 1$, and overdamped by $\zeta > 1$. An overdamped system will move slowly to an equilibrium position compared to the underdamped and critically damped cases, but an underdamped system will oscillate around the equilibrium point.

Rayleigh Damping

Viscous velocity damping is difficult to visualize for most structures, and simplified damping matrices are used in computer and mathematical models to avoid a matrix based on the physical properties of the structure (Wilson (2002)). Rayleigh damping is one type of damping often used in mathematical models to simulate the dynamic response. Rayleigh damping makes the assumption that the damping matrix is proportional to the mass and stiffness matrices (Wilson (2002)):

$$\boldsymbol{C} = \eta\boldsymbol{M} + \delta\boldsymbol{K}. \tag{2.25}$$

The definition of the damping ration $\zeta = c/c_c = 2m\omega$ and the orthogonal properties of the mass and stiffness vectors gives:

$$2\omega_n\zeta_n = {\phi_n}^{\mathrm{T}}\boldsymbol{C}\phi_n = \eta{\phi_n}^{\mathrm{T}}\boldsymbol{M}\phi_n + \delta{\phi_n}^{\mathrm{T}}\boldsymbol{K}\phi_n. \tag{2.26}$$

The orthogonal properties further gives:

$$\zeta_n = \frac{\eta}{2\omega_n} + \frac{\delta\omega_n}{2}. \tag{2.27}$$

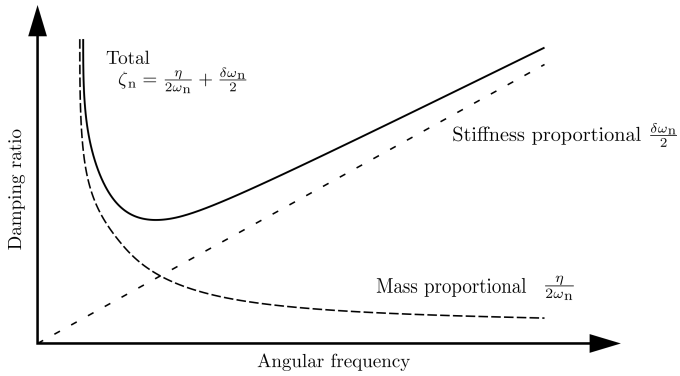


Figure 2.7: Relationship between Rayleigh damping ratio and frequency, and how the mass and stiffness proportional damping affects the total Rayleigh damping.

Equation 2.27 can be used to estimate δ and η, by determining the damping ratio at two separate frequencies and solving for δ and η.

Figure 2.7 shows the relationship between Rayleigh damping and frequency and how the mass and stiffness proportional damping affects the total Rayleigh damping. Generally, the mass proportional damping will have the largest impact at smaller frequencies, while the stiffness proportional damping will have a larger impact at higher frequencies.

Chapter III

The Burridge-Knopoff-Pad Model and Previous Work

As motioned, three other master students have worked on this project, two of whom this thesis is built upon. This chapter will describe the BKP model in a similar way done by the two previous students Ferre (2018) and Standnes (2019). It will also summarize the essence of previous students' work.

3.1 The Burridge-Knopoff-Pad Model

The Burridge-Knopoff-Pad (BKP) model was first presented by Ferre (2018) in his master's thesis, which to his awareness, had never been studied before. The model combines an SDOF model, shown in Section 2.3, and the Burridge-Knopoff model, showed in Section 2.2, in order to capture the squealing sound of car brakes. The SDOF model is frequently used to model braking systems, but Ferre (2018) argues that it is too simple to study brake vibrations and noise and that its simplicity might be the reason it is frequently implemented. The blocks are thus included in the BKP model as a discretization of the elastic brake discs in a physical braking system to represent their natural vibrations.

Figure 3.1 illustrates the BKP model. The figure clearly shows that the model combines the SDOF system from Section 2.3 (top section of the illustration) and the BK model (bottom section of the illustration). The equations governing the BKP model is shown in Equation 3.1, and is also a combination of the equations governing the BK and SDOF models from Equation 2.1 and 2.2, respectively.

The pad position is described by x and each blocks position by u_j, where $j = 1, 2, \ldots, N$ and N is the number of blocks. The friction law $\phi(v + \dot{u}_j)$ is described in Section 3.2. The rest of the parameters in Equation 3.1 are described in Table 3.1.

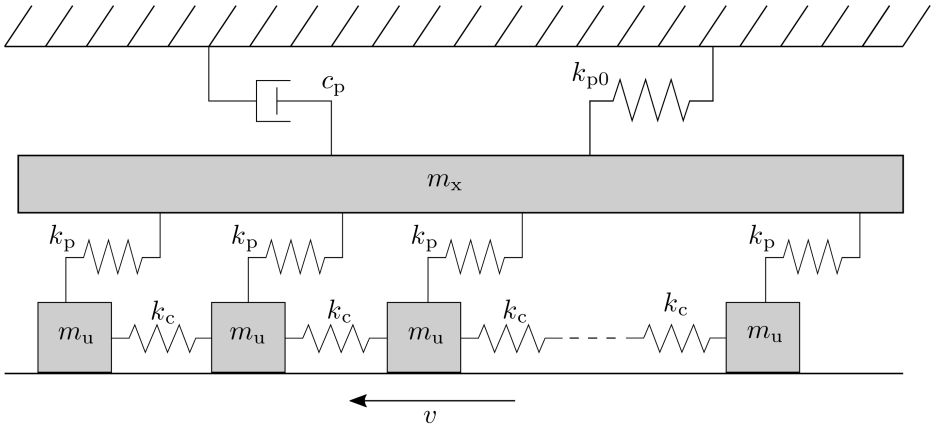


Figure 3.1: Illustration of the Burridge-Knopoff-Pad model. The top part of the illustration including the block with mass m_x originates from the SDFS model and the bottom part of the illustration including the blocks with mass m_u is taken from the BK model.

$$\begin{aligned} m_u\ddot{u}_j(t) &= k_c(u_{i+1}(t) - 2u_i(t) + u_{i-1}(t)) - k_p(u_j - x(t)) - m_u\phi(v + \dot{u}_j) \\ m_x\ddot{x}(t) &= -c_p\dot{x}(t) - k_{p0}x + k_p\sum_{j=1}^{N}(u_j - x(t)) \end{aligned} \tag{3.1}$$

3.2 Friction Law

The dimensionless friction law used is similar to the general stick-slip friction law outlined in Section 2.1.2, and is given by Equation 3.2.

$$\phi(y) = \begin{cases} F_0[-1,1], & y = 0 \\ F_0\frac{1-\sigma}{1+\frac{|y|}{1-\sigma}}sgn(y), & y \neq 0 \end{cases} \tag{3.2}$$

The parameter σ is the friction law scaling parameter and F_0 is the maximum static friction force. The parameter y is the relative velocity between the objects acted on by the friction force, and sgn is the signum function returning real numbers. Note that the friction law allows for negative displacement, a necessity for the simulation as the blocks are dragged backwards by the slider plate up to the instance where the pulling springs overcome the friction force.

3.3 Parameters

Table 3.1 below shows all the parameters in Equation 3.1, in addition some parameters used to simplify the implementation of the model, all chosen by Ferre (2018). The additional parameters and how they are used to determine the *Derived parameters* in Table 3.1 are explained in this section. Note that all parameters are dimensionless.

User defined parameters	**Symbol**	**Value**	
Number of blocks	N	100	
Pad mass	m_x	100	
Top plate-pad spring constant	k_{p0}	100	
Damping ratio	ζ	$\frac{1}{12}$	
Maximum static friction force	F_0	1	
Friction law scaling parameter	σ	0.01	
Slider speed	v	Varying	
Block mass scaling factor	s_m	1	
Pulling spring constant scaling factor	s_p	1	
Neighboring spring scaling factor	s_c	0.01	
Derived parameters			Given by
Block mass	m_u	1	$s_m \frac{m_x}{N}$
Pulling spring constant	k_p	1	$s_p \frac{k_{p0}}{N}$
Neighboring spring constant	k_c	100	$s_c k_{p0} N$
Critical damping coefficient	c_c	≈ 282.84	$2\sqrt{m_x(k_{p0} + N k_p)}$
Pad damping coefficient	c_p	≈ 23.57	ζc_c

Table 3.1: Parameters used in the BKP model.

3.3.1 Scaling Factors

To ensure that the friction force on the pad is independent of the number of blocks, the mass of each block, m_u, is proportional to the mass of the pad divided by the number of blocks, $m_u = m_x s_m / N$. Here, s_m is the block mass scaling factor, which is set to 1. Similarly, by the same argument, the spring constant between the pad and the blocks, k_p, is proportional to the upper spring constant, k_{p0}, divided by N times the scaling factor s_p, which is also set to 1. The neighboring spring constant, k_c, is proportional to N to ensure the blocks' axial stiffness to be independent of the number of blocks. The neighboring spring constant is also multiplied to the neighbouring spring scaling factor s_c, which is set to 0.01.

3.3.2 Damping Ratio

From Section 2.5.4, $\zeta = c/c_c$, where $c_c = 2\sqrt{m_x k}$. In this case, $c = c_p$ and $c_c = 2\sqrt{m_x K}$, where K is the sum of all spring constants connected to the pad, $K = k_{p0} + N k_p$. Inserting numbers, one can calculate c_c resulting in $c_c \approx 282.84$ as seen in Table 3.1.

With the damping ratio set to $1/12$, a number chosen by previous students, $c_p \approx 23.57$, as seen in Table 3.1.

3.4 Numerical Implementation

The BKP model in Equation 3.1 is a set of second-order differential equations that can be solved numerically. As stated, Burridge and Knopoff (1967) suggested using 2nd order Runge-Kutta as the numerical scheme for the BK model. With that as a basis, the previous master students have chosen the numerical scheme 2nd order Runge-Kutta to simulate the BKP model, which is also used in this thesis.

Writing the BKP in vector form as in Equation 2.7 yields Equation 3.3. By substituting Equation 3.3 for y in Equation 2.6, the BKP equation can be solved using the 2nd order Runge Kutta.

$$\begin{aligned} \vec{u_j} &= \begin{bmatrix} u_j \\ \dot{u}_j \end{bmatrix}, \quad \dot{\vec{u_j}} = \\ &\begin{bmatrix} \dot{u}_j \\ \frac{1}{m_u}(k_c(u_{j+1}(t) - 2u_j(t) + u_{j-1}(t)) + k_p(u_j - x(t)) + m_u\phi(v + \dot{u}_j)) \end{bmatrix} \\ \vec{x} &= \begin{bmatrix} x \\ \dot{x} \end{bmatrix}, \quad \dot{\vec{x}} = \begin{bmatrix} \dot{x} \\ \frac{1}{m_x}(-c_p\dot{x} - k_{p0}x + k_p\sum_{j=1}^{N}(u_j - x)) \end{bmatrix}. \end{aligned} \tag{3.3}$$

3.4.1 Time Step

Ferre (2018) chose the time step, Δt, based on the value which gives the vibrational mode its highest frequency. The highest frequency of the BKP model will be the mode where all adjacent blocks move with the opposite phase. The natural undamped frequency of this mode is given as:

$$\omega = 2\pi f = \sqrt{\frac{2(2k_c) + k_p}{m_u}}. \tag{3.4}$$

With a safety factor of $1/20$, the time step can be calculated as:

$$\Delta t = \frac{1}{20f} = \frac{1}{20}\frac{2\pi\sqrt{m_u}}{\sqrt{4k_c + k_p}} = 0.0157. \tag{3.5}$$

The above number is thus a definite upper bound for Δt to ensure numerical stability. With that as a basis, the time step Δt was set to 0.005.

3.5 Results from Previous Students

In this section, a summery of the results of the previous students will be provided. Ferre (2018), as the first student working on the project, introduced the model and created a codebase to simulate it, in addition to performing some analysis. Standnes (2019) continued by analyzing the model in different ways. Many of these results have been considered when continuing on the project.

3.5.1 General Behaviour

Standnes (2019) showed how the friction amplitude differed by using different initial conditions on the model. He showed that most simulations are quite similar but that some simulations stick out and differ from the rest. Standnes (2019) also created animations of the model, which shows how the pad and block position and velocity develops through time [1].

Standnes found differences in how the pad position changes with slider velocity, depending on whether the slider velocity increase or decrease during the simulation. The difference was found largest in the velocity range between 0.68 and 1.11, where the model transitions from low to high friction amplitudes. Standnes (2019) also uses the Fourier spectrum to view different peaks of amplitude from different frequencies of the model for some constant slider velocities.

3.5.2 Comparing the BK and BKP model

The previous students have compared the BK and BKP model. They found that both the BK and BKP model have two "dips" in friction amplitude around 0.11 and 0.16 (depending slightly on the simulation). Standnes (2019) found that even though the models show similar behaviour, the second dip is not always present in the BKP model. Standnes argues that running larger batches of both models would be interesting for further work to see if this behaviour continues.

Standnes (2019) also shows plots where it is evident that including a pad in BK model lowers the friction amplitude in the slider velocity interval 0.25 to 0.68 compared to not

[1] Animations: https://izome.github.io/burridge-knopoff-pad-karsten/

having the pad, both for increasing and decreasing slider velocities. However, in the slider velocity interval 0.86 to 1.06, the decreasing BKP model shows the highest amplitude.

Chapter IV

Implementation Tools

This chapter will outline the implementation tools used to simulate and analyze the BKP model. The implementation tools include programming languages, programming structures, technologies, and some of the main classes and variables. The repository has additional code documentation explaining all the functions, variables, classes, and libraries used in the code, but this chapter will give an overall view of the procedure.

4.1 Overall Code Structure

Figure 4.1 shows the overall code structure. The code used to simulate the model is written in C++ (referring to C++11). A YAML-file, including the parameters described in Section 3.3 and a string with a proper run name, is used as input in the C++ code. YAML, which is a recursive acronym for *YAML Ain't markup language*, is a human-readable data serialization standard compatible with many programming languages (Ben-Kiki, Evans and döt Net (2001-2009)). When the C++ simulation program starts, it creates a folder with the run name specified in the YAML-input file. As the program runs, it saves the results, including the friction force and the pad and blocks' velocity and position, in separate CSV-files. CSV stands for *comma separated values* and is a file format used to save data in a simple and effective way (Python Software Foundation (2021)). The CSV-files are later used as input in Python-scripts that output figures enabling visualizations of the simulations.

The code-base and architecture for running the Runge-Kutta simulations are taken from Standnes (2019), which again took inspiration from Ferre (2018). Further adjustments are made in order to implement the additional functionalities required. As Standnes (2019) states, the implementation strives to have a good flow so that the code does not need changes between runs. Similarly, modifications and additions to the code are herein performed with re-usability in mind. This means that the functions are not dependant on a particular instance and can be reused between runs, making it easy for the user to choose what functionalities to use.

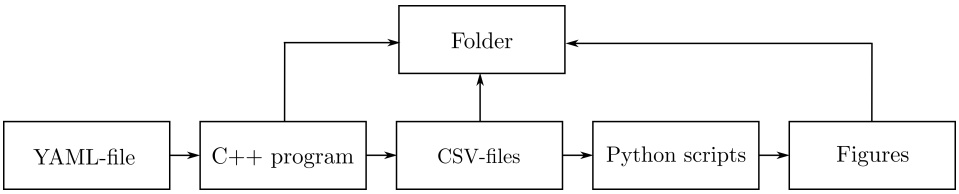


Figure 4.1: Overall code structure. The C++ program takes a YAML-file as input and outputs CSV-files, which are the input of different Python scripts, which again output figures. Output data and figures are all saved in the same folder.

4.2 C++ Code

The C++ code is built up in an object-oriented manner with classes. The code is split into three classes, called Simulation, Parameters, and Operation. Simulation and Parameters were made by Standnes (2019) and have herein been further modified. Each class has a header file (.hpp) and one or several C++-files (.cpp). The class Parameters deals with reading the YAML-input-file, called parameters.yaml, and saving the parameters to a Parameters object. The desired input parameters are specified in the parameters.yaml file before running a new simulation. Section 4.6 describes parameters.yaml in detail.

When a Simulation object is initialized it takes in an instance of Parameters and creates a result-folder. The Simulation object then simulates the BKP model using the 2^{nd} order Runge-Kutta method and creates CSV-output files which are added to the generated folder. These output files include the friction force and the pad and blocks position and velocity as CSV-files. The Operation class contains methods dealing with smaller operations such as computing eigenvalues, eigenvectors, and mode shape contributions, which are further discussed in Chapter 5. When an Operation object is initialized, it takes in an instance of Simulation and when the program is finished, it outputs the eigenvalues, eigenvectors, and modal contribution matrices as CSV-files, all added to the same result folder. A main.cpp file with a main method is used to create the class instances and initializes the simulation and operations. The connection between the header files and the main script is illustrated in Figure 4.2.

4.2.1 Compilation

In C++, the code needs to be compiled to machine code that the processor can execute before running the program (Stroustrup (2014)). This is in contrast to, for instance, the language Python, where the code compiles through the interpreter [1] (Asadi (2016)). The advantage with compiled languages is that the execution time is faster than for interpreted

[1] In interactive mode, Python code can also be run as a compiled program, but it is usually defined as an interpreted language.

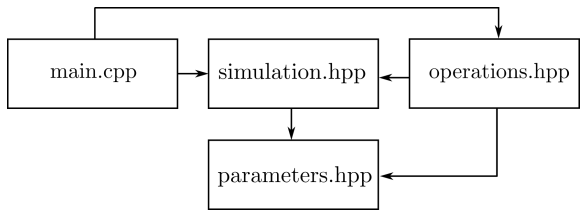


Figure 4.2: Relationship between the main script, main.cpp, and the header files in the program.

languages (Asadi (2016)), which is the main reason it is used for the simulations of this project.

During compilation, each C++ file is compiled into object-files, which are linked together by a *linker* to make them executable (Brigs (2019)). If a program consists of various scripts (and hence various object files after compilation), and a change is done in one of the scripts, the linker allows for the recompilation of only the scripts that have been edited. One such linker that helps to monitor what files in the program has changed is a Makefile (Brigs (2019)). Makefiles includes both flags/dependencies and rules, which specify what will be recompiled after a change is made (Brigs (2019)). Makefiles can be self-written, but there are also programs that automatically create makefiles, such as *Cmake*.

While the previous students compiled using Cmake, however the adopted code could not compile and Cmake made errors challenging to debug. In this work a self-written makefile was created using the compiler g++. The inability to compile could be attributed to several reasons, e.g., use of a different operating system, outdated packages, or insufficient documentation of dependencies. See the readme.md files in the repository for a more detailed description of how to compile using the makefile.

4.2.2 Libraries

These are the main libraries used in the C++ code:

- Armadillo
- Eigen
- YAML

Armadillo is a library simplifying all matrix operations necessary in this application (Sanderson and Curtin (2016)). Almost all vectors and matrices in the program are made using Armadillo. Eigen is a template library for linear algebra with matrices, vectors, and operations, amongst others (Guennebaud, Jacob et al. (2010)). As mentioned, YAML

is a file format, and the YAML library is necessary to import for the program to read the YAML input file (Ben-Kiki, Evans and döt Net (2001-2009)).

4.3 Python Code

Python3 (otherwise referred to as "Python") is used to visualize the data output from the simulations. Most of the code is built up in an object-oriented manner with classes and methods. Most of the scripts deal with plotting, and others are purely help-functions. The script `help_functions.py` contains the class `LoadFile`, which deals with operations such as loading the YAML-file, loading the simulation outputs, and saving created figures. The `LoadFile` class is used in almost all other classes and scripts.

The libraries Matplotlib and Seaborn are used for plotting. The Matplotlib package is the main Python package for data visualization (Hunter (2007)). Seaborn is a visualization library used on top of Matplotlib with a large focus on styling (Michael Waskom and the Seaborn development team (2020)).

In order to keep an organized repository, both the Python scripts and the figures are saved to the folder with the simulation run name. Changes are continuously made in the Python scripts, making it helpful to have a copy of the previously used script. Note that the disc space taken up by some copies of the scripts is small compared to the disc space of simulations.

4.4 Technologies

4.4.1 Editor

The editor and programming environment used is Visual Studio Code (VS Code). VS Code is a free, open-source text editor used to view, run, edit and debug code (VS Code (n.d.)). It is the most widely used for client-side applications written in JavaScript, but it also supports C++ and Python (VS Code (n.d.)). The previous students used VS Code for this project, hence it was guaranteed it has the necessary functionalities for the project. Based on the above, VS Code was chosen.

4.4.2 Version Control and Project Repository

The repository is found on a gitolite-server set up by the supervisor and accessed with SSH. Git is used as version control. The author of this thesis has also cloned the repository to her private GitLab profile so that others can easily access the work [2].

[2] `https://gitlab.stud.idi.ntnu.no/theamm/burridge-knopoff-pad`

4.4.3 Hardware

A Mac Book Air with 4 GB ram and 1,4 GHz processor is used for the project. The Mac Book is not very powerful, and a more powerful computer would be preferable. A backup disk was also needed to store the files, as one single simulation easily reach 1-2 GB and sometimes reached more than 20 GB.

4.5 Documentation and Code Base

The documentation and repository for the project can be accessed though the following:

- Gitolite repository on a remote server.
- The authors private GitLab repository (equal to gitolite repository)[3].
- Doxygen HTML page, including all code documentation found in the repositories.
- Doxygen Latex document (equal to the Doxygen HTML page, but in pdf.-format).
- Web page with extended appendices [4].

4.5.1 Repository

The repository contains the code base for the project. The repository also contains four readme.md (markedown)-files — one on the main page and one in three of the three sub-directories. Note that previous students' functionalities that are not used in this thesis have been kept unchanged but are included in the repository "archive" folder.

4.5.2 Doxygen

Doxygen is a standard tool for generating documentation from C++, and it also supports several other languages, such as Python. The doxygen files made in this project only contains the C++ code documentation, as including the python code was considered excessive. Doxygen is used to generate both an on-line HTML-page and an off-line Latex document with both the documentation from the source code and visualization of the code structure.

4.5.3 Web Page

A web page was created as an extended appendix which can be helpful for further work with the project [5]. A description of how the web page is created is found in Appendix B.

[3] See footnote 2.
[4] `https://theamartine.github.io/Burridge-Knopoff-Pad-Model/`
[5] See footnote 4.

4.6 Model Input

The following code is the parameters.yaml-file that is used as input in the model. Table 3.1 describes most of the parameters, but this section explains the rest.

```
Parameters:
  dt: 0.005
  seed: 101
  num_events: 1
  N: 100
  max_time: 80000
  slider_speed: 0
  increment: 0
  interval: 0
  file_name: "test"
  progress_indicator: true
  m_F0: 1
  m_alpha: 0.5
  m_sigma: 0.01
  m_mass_x: 100
  m_scale_mass: 1
  m_zeta: 0.0833
  m_k_P0: 100
  m_scale_P: 1
  m_scale_C: 0.01
  m_t: 0.0
  m_v0: 0.0001
  m_u_min: 0
  blocks:
  start_speed_continuous: 2
  end_speed_continuous: 0
  save_interval_dt: 100
  threshold_speed: 0.1
  m_eta: 0.2
  m_delta: 0.047
  lyapunov_delta: 0.00000001
Debug:
  debug_no_friction: false
  debug_no_neighbor_springs: false
  debug_no_stationary_springs: false
  debug_no_damper: false
  debug_no_pad: false
  debug_negative_initial_values: false
  debug_only_negative_initial: false
  debug_only_write_friction: false
  debug_continuous_slider_speed: true
  debug_one_degree_freedom_mode: false
  rayleigh_damping: false
```

The number of time steps will be equal to `max_time` divided by `dt`. The parameter `save_speed_interval_dt` determines the fraction of time steps that will be part of the

output files. Thus, the number of saved time steps is equal to the total number of time steps divided by `save_speed_interval_dt`. By not saving each time step, disk space is spared and time used to save output files decreased, along with the run time to plot numerical results.

Standnes (2019) tried both step-wise and continuous change of slider velocity. The parameters `increment`, `interval` and `threshold_speed` were made by Standnes (2019) and are not used in this thesis, but will be explained due to their presence on the parameter list. Step-wise change means that a velocity is kept constant for a certain amount of time steps before increasing or decreasing with a jump. The parameter `increment` determines the amount the slider velocity is incremented at each step-wise change. The number of time steps run at each slider velocity is determined by `interval` divided by the total number of time steps. With a step-wise slider velocity, there will be created separate output files for each slider velocity.

In this thesis, the program has been run with continuous increase or decrease of slider velocity. That means that the slider velocity changes with the same step size each time step, giving a constant acceleration or deceleration. The two parameters `start_speed_continuous` and `end_speed_continuous` are used to set the slider speed if the boolean value `debug_continuous_slider_speed` is set to `true`. According to Standnes' research, the system is more robust when running step-wise simulations, which can be because the system will have some time to stabilize for each change in slider velocity. Nevertheless, step-wise change also leads to more extensive sudden disruptions making the model require more time to stabilize than if using continuous increasing or decreasing slider velocity.

If the parameter `threshold_speed` is used, the program will increase or decrease the slider velocity continuously until the program reaches the specified threshold. Then the program will run with that slider speed (which is now equal to the threshold speed) throughout the run. Standnes (2019) argues that the use of threshold velocity did not prove useful to understand the model better.

The difference between the parameters `N` and `blocks` should also be noted. As `N` is an integer representing the number of blocks used in the simulations, the list of `blocks` is the blocks that will be part of the output file. If `blocks` is left blank, all blocks will be part of the output. As for `save_speed_interval_dt`, the reasons to not always output all blocks are to save disk space and decrease the time used to save the simulation output and to create plots.

The parameter `file_name` should be a short but descriptive name for the given simulation. This will be the name of the directory where the results are saved.

The parameter `seed` determines how the matrices used in the simulations are initialized.

Different seed numbers lead to different initial conditions.

The values `m_eta` and `m_delta` are the two Rayleigh constants of Equation 2.25. These are implemented in addition to the damper c_p in the BKP model of Equation 3.1. In order to only have Rayleigh damping (and c_p set to zero), the Boolean value `debug_no_damper` needs to be set to `true`.

The parameter `lyapunov_delta` is used in the code to calculate the maximum Lyapunov exponent. The implementation is not completed.

As for `debug_no_damper`, the other Boolean values are also used to isolate different parts of the model or to use different parts of the program. For example, by setting the parameter `debug_no_pad` to `True`, the system will simplify the BKP model to the BK model.

Chapter V

Analysis Methods

Whereas the previous chapter dealt with how the model is simulated and how the results are plotted in terms of programming and computer tools, this chapter will provide the methods used to analyze the model on a more theoretical level. It will first introduce the general behaviour of the model, and how some some of the parameters from the previous chapter are chosen. Further, the chapter will describe how the percentage of stick-slip behaviour is calculated. Section 5.4 will show how the eigenvalue problem is implemented in both the BK and BKP model, and how it is used to study the models. Lastly, the chapter will describe the implementation of Rayleigh damping in the BKP model.

5.1 General Behaviour

Figure 5.1 illustrates the pad position plotted as a function of slider velocity as the slider velocity decreases with constant deceleration from 2 to 0. The run specifications are found in Appendix A.1. Figure 5.1 shows that in the beginning of the simulation, when the slider velocity is close to 2, the system uses some time to reach a steady state before the pad is dragged backwards by the movement of the slider plate and the position gets a negative value. At high velocities, in the slider velocity range $v \in [1.9, 0.75]$, the pad position oscillate with decreasing amplitudes. When the slider velocity is approximately 0.7, there is a sudden jump in position amplitude. As the slider velocity continues to decrease, the amplitude of the position gets smaller. When the slider velocity is approximately 0.1, there is a sudden shrink in position amplitude before it gets larger after a short time interval.

As will be shown in Section 5.2, varying the initial conditions will lead to changes in the system's behaviour. Nevertheless, most of the general behaviour explained in this section will remain the same.

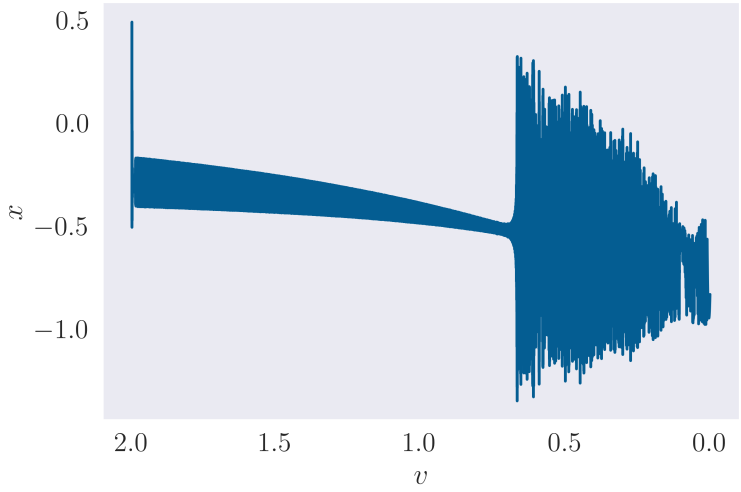


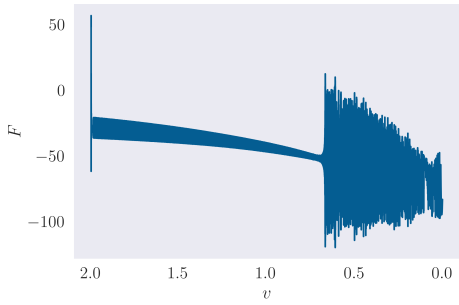
Figure 5.1: Pad position as a function of slider velocity where the slider velocity decreases continuously from 2 to 0. (Run specifications: Appendix A.1.)

In the rest of the thesis, in order to quickly see the general behaviour of the model, friction is plotted as a function of slider velocity, as in Figure 5.2(a) which clearly shows the same general behaviour as the pad position in Figure 5.1. Note that in the case when the slider velocity decreases continuously, plotting against slider velocity is equivalent to plotting against time, as $v = 2$ is at $t = 0$ and $v = 0$ is at $t = 80\ 000$, where $80\ 000$ is the max time of this run. The latter is seen by comparing Figure 5.1 and Figure 5.2(b), where the pad position plotted against time. The pad velocity plotted against time in Figure 5.2(c). The general behaviour of the blocks' position and velocity is similar to that of the pad and can be seen in Appendix C.

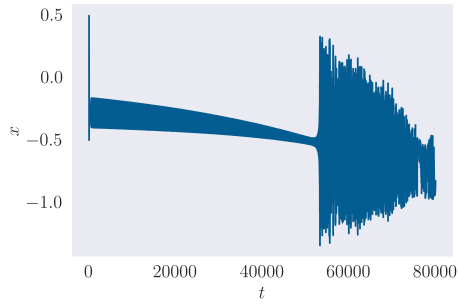
5.2 Parameters

Different initial conditions have proven to affect the system differently. The following section will show how the parameters seed, max time, start/end velocity, and save interval affect the response of the system.

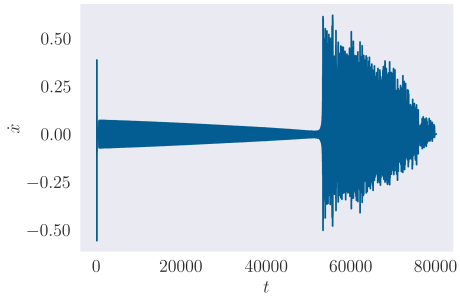
Note that there are many different ways to compare and plot different runs. The previous students have calculated the error/difference between each time step in a run and compared it to the mean value of all runs. In this thesis, the primary way to compare runs has been to plot friction versus time of each run next to each other. A considerable advantage with this approach is that one can easily see the difference in the runs at critical sections of the



(a) Friction as a function of slider velocity.



(b) Pad position as a function of time.



(c) Pad velocity as a function of time.

Figure 5.2: Different plots of the pad behaviour from a simulation with constant deceleration. (Run specification: Appendix A.1.)

model. A disadvantage is that it does not accurately quantify to what extent the simulations differs.

5.2.1 Seed

As mentioned in Section 4.6, the seed determines the simulation's initial condition. Figure 5.3 shows the friction plotted against slider velocity for several runs where the seed is the only varying factor. As seen in Figure 5.3, the seed choice leads to some variations. Many runs, such as the run with seed 160 in the bottom right corner, has an area of "medium increased" amplitude, around $v = 1$, before the area of larger amplitudes starts around $v = 0.7$. This behaviour can also be seen for many other seeds (e.g. 1 and 10).

Another difference is the amplitude at higher slider velocities. For instance, seed 101 and seed 40 have a higher amplitude in the range from slider velocity 1.9 to 1, compared to the other runs. Thirdly, there is an evident amplitude drop at very low slider velocities ($v \approx 0.2$) for some runs, such as seed 30, 40, and 70, which is not visible for e.g. seed 90 and 100.

There are differences in the model's response depending on the seed. Therefore, when analysing the model, different seeds will be considered. However, to retain consistency with previous work with the model, seed 101 will primarily be used to analyse the model, though seed 100 is often used as a compliment. From Figure 5.3 these two seeds, 100 and 101, have distinct behaviours in some regions, making it reasonable analyse both.

5.2.2 Max Time

Different simulations are run to decide the max time, which again determines the number of time steps. Figure 5.4 shows a comparison between six simulations where the only parameter changed is the max time.

The two subplots with blue shade, which represent a max time of $10\ 000$ and $20\ 000$, show severe differences from plots with a higher max time, indicating that it is too low. The behaviour of the model does not change significantly when increasing the max time beyond $40\ 000$, and this is thus assumed to be sufficient to represent an accurate response of the model. Based on this, a max time of $80\ 000$ is used when the slider velocity moves from 2 to 0 or 0 to 2. $80\ 000$ was also the max time used by Standnes (2019) in most of his simulations. A max time of $80\ 000$ is equivalent to $16\ 000\ 000$ time steps with the fixed dt of 0.005.

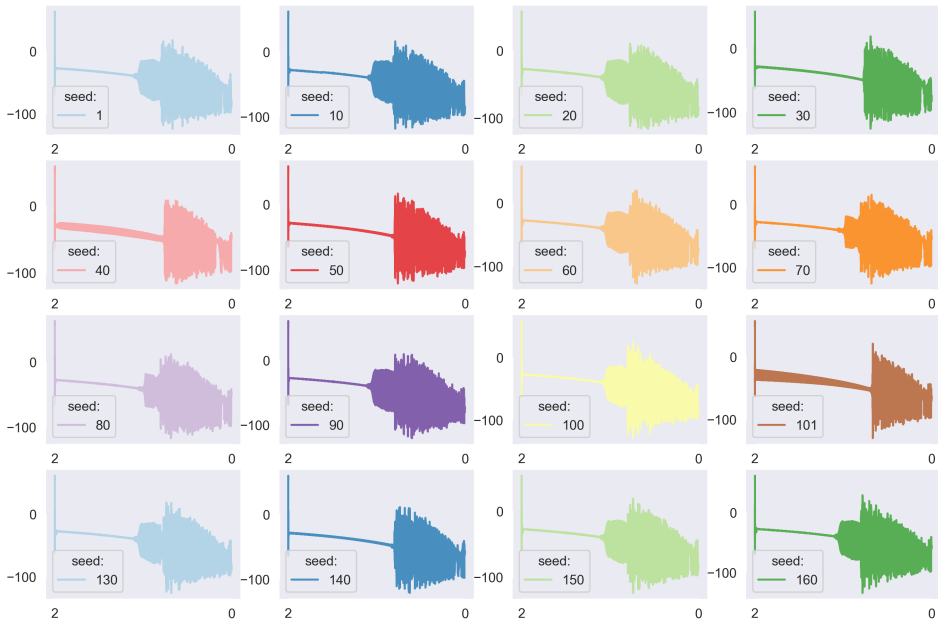
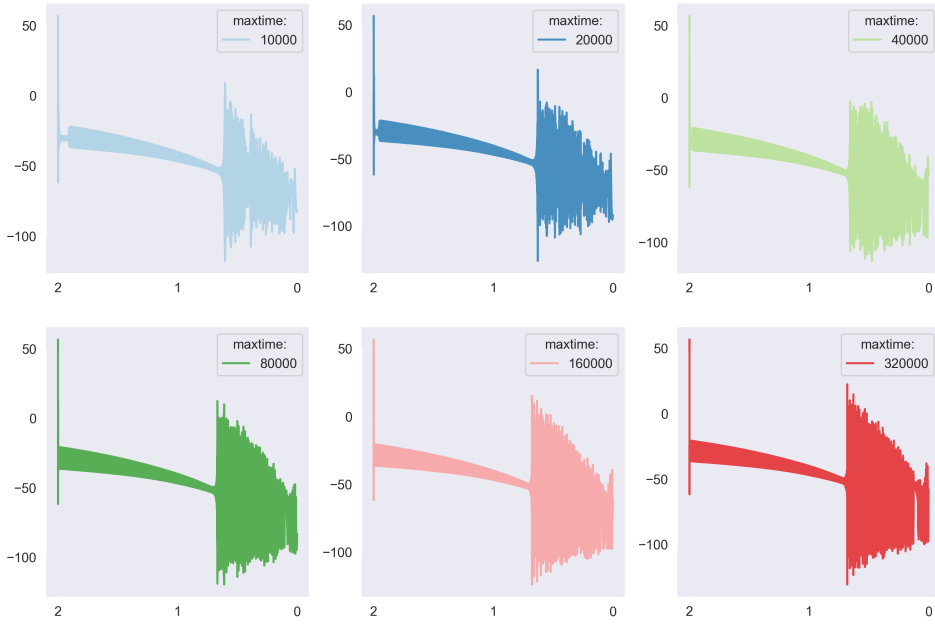
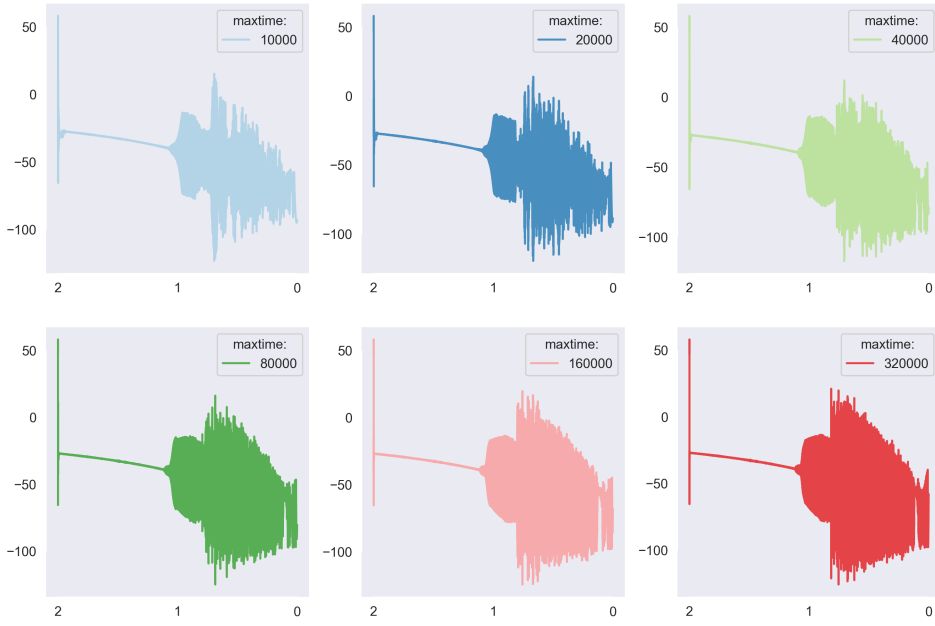


Figure 5.3: Friction force versus slider velocity for 16 different runs of different seed. (Run specifications seed 101: Appendix A.1. Only the seed is changed in the rest of the runs.)



(a) Seed 101.



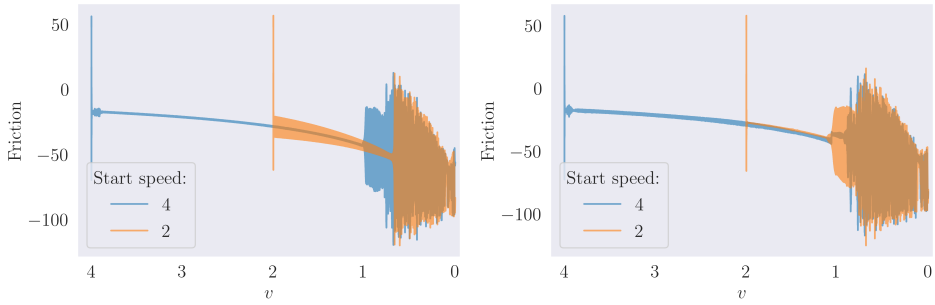
(b) Seed 100.

Figure 5.4: A max time comparison between six different runs for two different seeds. (Run specifications seed 101 max time 80 000: Appendix 1. Only max time and seed are the varying parameters in the other runs.)

5.2.3 Start and End Slider Velocity

Visualized in Figure 5.5 are two runs starting at $v = 4$ and $v = 2$ for seed 100 and 101. As can be seen, the conditions at $v = 2$ is different for the two runs. Note that when starting the simulation at $v = 4$, the max time number is doubled compared to the runs starting at $v = 2$.

As the behaviour is mostly interesting when v is below approximately 1.3, it is chosen to use a start slider velocity of 2, as this is considered sufficient to capture that area.



(a) Seed 101.

(b) Seed 100.

Figure 5.5: Two different runs plotted on top of each other for two different seeds, where the orange plots has an initial velocity of 2, while the blue has an initial velocity of 4. (Run specifications: Appendix A.1, A.3, A.4 and A.5)

5.2.4 Save Interval

As mentioned in Section 4.6, the save interval determines the fraction of the saved time steps. Decreasing the number of saved time steps will not affect the simulation's run time but will decrease the run time of all post-processing and plotting operations. Standnes (2019) analyzed how the save interval affects the system. He shows that the friction amplitude error increased proportionally as fewer time steps were saved and used a save interval of $1/100$, which has also been used in this thesis. For some runs, the save interval is set to $1/10$. This is specified in Appendix A.

5.3 Stick

It was desired to get more familiar with how stick depends on slider velocity. For different slider velocities, the mean percentage of stick for each block can be calculated.

In relation to Section 2.1.1, stick is when there is zero relative velocity between a block

and the slider plate. The latter translates to when a block's velocity is equal to the slider velocity. However, checking when two continuously changing numbers are equal is complicated in a numerical simulation, as everything changes step-wise. To get around this issue, the definition of stick used here is when the velocity of a block, rounded to three decimals, is equal to the slider velocity, rounded to three decimals. Therefore, the stick percentage calculated is not entirely accurate, but it can indicate how stick varies with slider velocity.

There are also other ways to calculate stick. Corrado and Mascia (2016) calculates stick in the BK model. They define a block as "stuck" if both the elastic resultant force is less than the maximum static friction force *and* the velocity is equal to zero. Corrado and Mascia (2016) define zero relative velocity as when there is a sign change and the velocity becomes negative. This is different than the method used in this thesis to define zero relative velocity. Further, Corrado and Mascia (2016) state that using a threshold value, similar to what is done here, for creating a range for the zero velocity is a popular method to calculating stick.

5.4 The Eigenvalue Problem

Section 2.5.3 showed that when neglecting damping, the eigenvalue problem can be expressed as in Equation 2.17.

$$\boldsymbol{M}\ddot{\boldsymbol{x}} + \boldsymbol{K}\boldsymbol{x} = 0. \tag{2.17}$$

In order to get a better understanding of the BKP model, it was desired to study the most dominant frequencies using eigenvalues and eigenvectors. This section will demonstrate how to compute the eigenvalues and eigenvectors for the BK and BKP model, and how they are used to analyse how the activation of different modes change during simulations of the BKP model.

5.4.1 Mass and stiffness matrices for the BKP Model

In order to compute the eigenvalues and eigenvectors for the BKP model, the equation of motion for the BKP model in Equation 3.1 has to be rewritten on the form of Equation 2.17. In the case of the BKP equation, $\boldsymbol{x}$ is the position vector for both the blocks and the pad, on the form $[u_1, u_2, u_3, \ldots, u_N, x]$. Its second time derivative is $\ddot{\boldsymbol{x}}$. The number of degrees of freedom in this system is $N + 1$, one degree of freedom for each block, and one for the pad. Hence, both the mass and stiffness matrices will be of dimension $N + 1 \times N + 1$, where the first N columns and rows represent the blocks and the last row and column represent the pad. The mass matrix is presented in Equation 5.1 with all the masses in the system on the diagonal, where it is shown how the N blocks each have mass m, and the pad has mass M.

The stiffness matrix is derived by considering all the spring constants in the BKP model and translating it to matrix form. From the illustration of the BKP model in Figure 3.1, the first block from the left is connected to a block on its right side with a spring of spring constant k_c and to the pad with a spring of spring constant k_p. The $(0,0)$ element of the stiffness matrix will hence be $k_p + k_c$. The $(0,1)$ element will be $-k_c$ as it describes the stiffness from block number two from the left onto block number one. Similarly, the $(0,N)$ element will be $-k_p$, as $-k_p * x$ describes the spring force from the pad to the block. In the same way, all the other elements are derived, resulting in the stiffness matrix in Equation 5.2.

$$\begin{bmatrix} m_1 & 0 & \dots & 0 & 0 \\ 0 & m_2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & m_n & 0 \\ 0 & 0 & \dots & 0 & M \end{bmatrix} \Rightarrow \begin{bmatrix} m & 0 & \dots & 0 & 0 \\ 0 & m & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & m & 0 \\ 0 & 0 & \dots & 0 & M \end{bmatrix} \tag{5.1}$$

$$\begin{bmatrix} k_c + k_p & -k_c & 0 & \dots & 0 & 0 & -k_p \\ -k_c & 2k_c + k_p & -k_c & \dots & 0 & 0 & -k_p \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 2k_c + k_p & -k_c & -k_p \\ 0 & 0 & 0 & \dots & -k_c & k_c + k_p & -k_p \\ -k_p & -k_p & -k_p & \dots & -k_p & -k_p & k_{p0} + N * k_p \end{bmatrix} \tag{5.2}$$

5.4.2 Mass and Stiffness Matrices for the BK Model

The mass and stiffness matrices of the BK model will be similar to those in the previous section. However, the BK equation does not have the pad on the $(N+1)^{\text{th}}$ row and column. Hence, the mass and stiffness matrices of the BK equation will be equal to those of the BKP with the $(N+1)^{\text{th}}$ row and column removed. The mass matrix for the BK model is shows in Equation 5.3 and the stiffness matrix in Equation 5.4.

$$\begin{bmatrix} m & 0 & \dots & 0 \\ 0 & m & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & m \end{bmatrix} \tag{5.3}$$

$$\begin{bmatrix} k_c + k_p & -k_c & 0 & \dots & 0 & 0 \\ -k_c & 2k_c + k_p & -k_c & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 2k_c + k_p & -k_c \\ 0 & 0 & 0 & \dots & -k_c & k_c + k_p \end{bmatrix} \tag{5.4}$$

5.4.3 Computing Eigenvalues and Eigenvectors

There are different functions, both in C++ and Python, for calculating eigenvalues and eigenvectors. The function `eig_pair` from the Armadillo library has the advantage of taking in two square matrices as input and solves the eigenvalue problem on the form it is in Equation 2.17 (Sanderson and Curtin (2016)). The two input matrices correspond to $\boldsymbol{K}$ and $\boldsymbol{M}$ in the eigenvalue problem. For mass and stiffness matrices of size $(N+1) \times (N+1)$, the function calculates and outputs $N+1$ eigenvalues and $N+1$ corresponding eigenvectors, each of length $N+1$. The eigenvalues are added to an Armadillo vector, and the eigenvectors are added as columns in an Armadillo matrix, where an index in the eigenvalue vector corresponds to the same column index in the eigenvector matrix.

Note that the Armadillo `eig_pair` function outputs both a real and imaginary part, but from the Section 2.5.3, when mass and stiffness matrices are symmetric, the imaginary part of the output will always be zero. As the mass and stiffness matrices for both the BK and BKP equations are symmetric, the imaginary part is zero and the program can safely be simplified to output only the real part of the eigenvalues.

Often, only the low frequency eigenmodes will be relevant for the research. Wilson (2002) argues that seismic motions excite only the lower frequencies of a structure and that neglecting large eigenvalues does not typically lead to errors. After looking at how the different modes affect the system, the 21 lowest modes were used. This will be showed in Chapter 6.

5.4.4 Normalization

There are different ways to normalize the eigenvectors. Examples of this are to use the first, second, and infinite norm and normalizing based on the mass and stiffness matrix. Mass generalized normalization is used following other structural dynamics literature (Wilson (2002)). Mass generalised normalisation means that

$$\boldsymbol{\phi}^{\mathrm{T}} \boldsymbol{M} \boldsymbol{\phi} = \boldsymbol{I}, \tag{5.5}$$

where $\boldsymbol{\phi}$ is an eigenvector (Chopra (2017)). In order to scale $\boldsymbol{\phi}$ so that Equation 5.5 holds, the following transformations are used (Chopra (2017)):

$$m_i = \boldsymbol{\phi_i}^{\mathrm{T}} \boldsymbol{M} \boldsymbol{\phi_i} \quad \boldsymbol{\phi_i} = \boldsymbol{\phi_i} / \sqrt{\boldsymbol{m}}, \tag{5.6}$$

where $\boldsymbol{m}$ is the vector containing the $N + 1$ m_i elements.

5.4.5 Mode Shape Plots

After computing the eigenvectors and eigenvalues, these can be plotted to view which modes are dominant at different times and slider velocities. With $N = 100$ blocks and one pad in the system, there will be 101 degrees of freedom, hence 101 eigenvalues with corresponding eigenvectors. Note that in this thesis, mode 0 refers to the mode shape of the lowest frequency. This contradicts some literature that uses mode 1 as the lowest frequency mode (Clough and Penzien (1995)).

Figure 5.6 illustrates the approach to plot the mode shapes used in this thesis. the figure shows each of the 101 values in the eigenvector corresponding to the second mode plotted as bars. The first bar on the left is the eigenvector value corresponding to the first degree of freedom in the mass and stiffness matrices, i.e., the first block from the left in the BKP model illustration in Figure 3.1. Further, the second bar represents the second block and so forth. The last bar represents the last degree of freedom from the mass and stiffness matrices, which describes the pad.

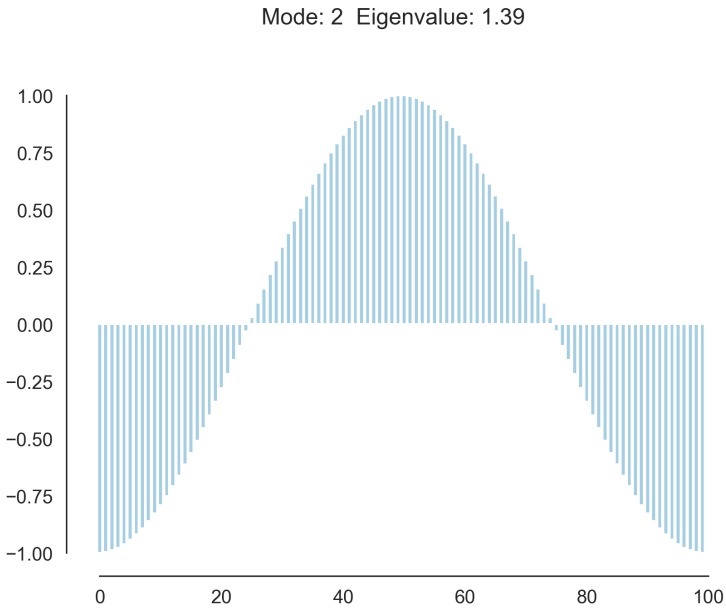


Figure 5.6: The eigenvector of mode 2 plotted as bars.

5.4.6 Interpretation of Different Modes

The goal of doing an eigenmode analysis was to understand what modes contribute to disturbances in the system, which can help understand the model's periodicity. Consider the mode shape matrix $\mathbf{\Phi}$ consisting of $N+1$ eigenvectors $\boldsymbol{\phi}$, each of length $N+1$:

$$\mathbf{\Phi} = \boldsymbol{\phi_1}, \ldots, \boldsymbol{\phi_{N+1}}. \tag{5.7}$$

Consider also the displacement matrix which consists of t vectors, where t is the number of time step, each of length $N+1$:

$$\boldsymbol{D} = \boldsymbol{x_1}, \ldots, \boldsymbol{x_t}. \tag{5.8}$$

By taking the dot product between a displacement vector at a given time step and an eigenvector, as shown in Equation 5.9, yields a number, e_{ij}, representing the contribution of the respective mode to displacement at the respective time.

$$e_{ij} = \boldsymbol{x_j} \cdot \boldsymbol{\phi_i} \qquad j = 1, \ldots, t \qquad i = 1, \ldots, N+1 \tag{5.9}$$

Similarly, one can take the dot product between each mode shape and the velocity vector $\dot{\boldsymbol{x}}_{\boldsymbol{j}}$, at each time step j, to obtain how much each mode contributes to the change in velocity over time.

$$e_j = \dot{\boldsymbol{x}}_{\boldsymbol{j}} \cdot \boldsymbol{\phi_i} \qquad j = 1, \ldots, t \qquad i = 1, \ldots, N+1 \tag{5.10}$$

One can also use the previous dot products to obtain the kinetic and potential energy in each mode. The kinetic energy for a given time step is given as

$$\begin{aligned} T &= \sum_{i=1}^{N+1} \frac{1}{2} m_i \dot{x}_i \\ &= \frac{1}{2} \dot{\boldsymbol{x}} \boldsymbol{M} \dot{\boldsymbol{x}} \\ &= \frac{1}{2} \dot{\boldsymbol{x}} \boldsymbol{\Phi} \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{M} \dot{\boldsymbol{x}} \\ &= \sum_{i}^{N+1} (\dot{\boldsymbol{x}} \boldsymbol{\phi_i}^{\mathrm{T}})(\boldsymbol{\phi_i} \boldsymbol{M} \dot{\boldsymbol{x}}) \\ &= \sum_{i,k}^{N+1} (\dot{\boldsymbol{x}} \boldsymbol{\phi_i})(\boldsymbol{\phi_i}^{\mathrm{T}} \boldsymbol{M} \boldsymbol{\phi_k})(\dot{\boldsymbol{x}} \boldsymbol{\phi_k}) \end{aligned} \tag{5.10}$$

From Section 2.5.3 $\boldsymbol{\phi}^{\mathrm{T}}{}_{\boldsymbol{i}} \boldsymbol{M} \boldsymbol{\phi_k}$ is a diagonal matrix, and by extracting the diagonal part of Equation 5.10 (Wilson (2002)):

$$T_{i,i} = \frac{1}{2} (\dot{\boldsymbol{x}} \boldsymbol{\phi_i})^2 m_i \tag{5.11}$$

Similarly, the expression for potential energy will be:

$$V_{i,i} = \frac{1}{2} (\boldsymbol{x} \boldsymbol{\phi_i})^2 k_i, \tag{5.12}$$

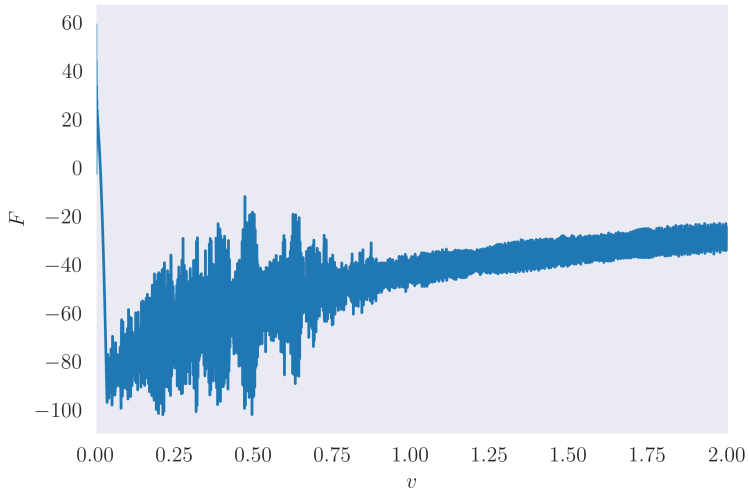
where m_i and k_i are the elements of the diagonal lines of the diagonal mass and stiffness matrices, respectively.

5.4.7 Heat Maps

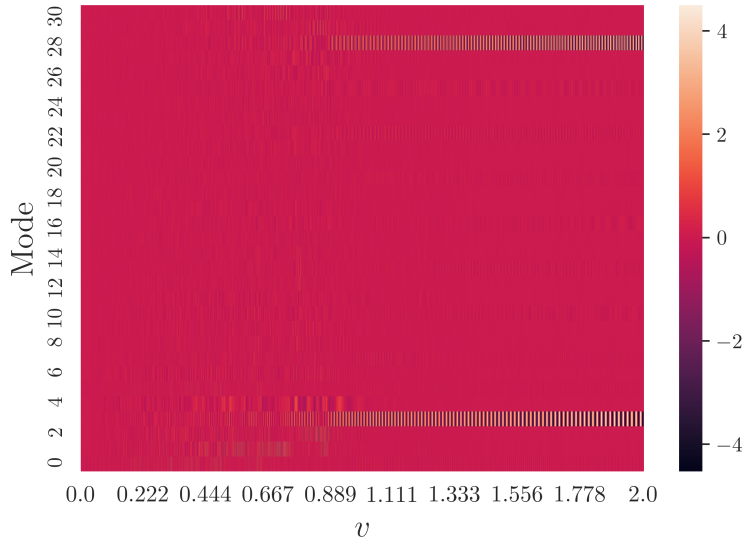
The modal contributions found in the previous section can be used to create spatiotemporal plots, also known as heat maps. The plots will have each mode on the vertical axis and time (or slider velocity, for runs with constant acceleration or deceleration of the slider velocity) on the horizontal axis. The content of the heat map will be the modal contribution

amplitudes for displacement, e, velocity, $\dot{e}$ or energy, which where found in the previous section.

A heat map is shown in Figure 5.7 for a small test run with few time steps and 30 blocks. The heat map shows how each mode contributes to velocity. Comparing this to the the friction plot in Figure 5.7(a) for the same run, the regions with much mess for low slider velocities in the heat map corresponds to the high amplitude in the friction plot for the same slider velocities. Reversely, the regions of low friction amplitude corresponds to regions with distinct activated modes contributing to velocity. This type of analysis will be done for several simulations in Chapter 6 Results.



(a) Friction plot.



(b) Velocity heat map.

Figure 5.7: Friction as a function of slider velocity for a test simulation with increasing slider velocity from 0 to 2. The heat map shows the contribution to velocity for each normal mode. (Run specifications: Appendix A.8.)

5.4.8 Removal of Dominant Modes

As shown in the displacement heat map in Figure 5.7(a), some modes are very dominant, making it difficult to visualize how the other modes affect the model. To get a better view of the other modes as well, three approaches have been tested. The first is to remove the most dominant modes from the plot. Note that with this approach, the modes are not removed from the simulation, only from the heat map. The second approach is to remove the mean value. Figure 5.8(b) shows a displacement heat map for the same run as Figure 5.8(a), but with subtracted mean. The third approach was to use only the steady-state solution of the displacement and velocity vectors. The latter was tested with poor results.

5.5 Rayleigh Damping

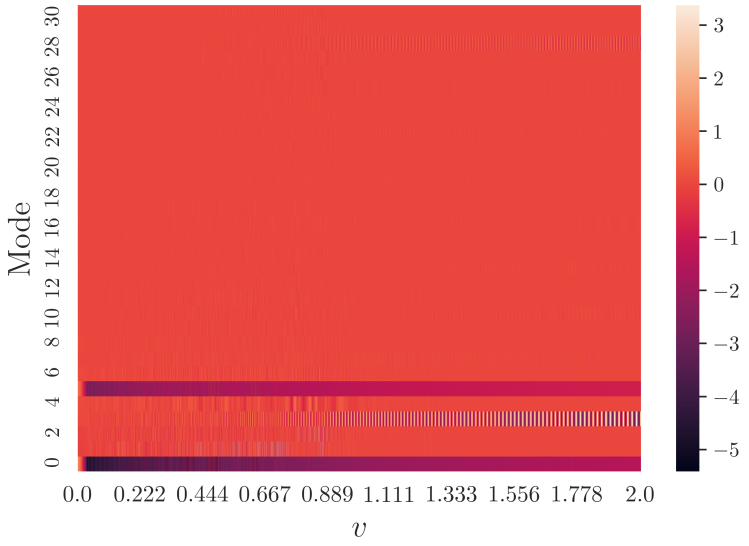
Section 2.5.4 describes Rayleigh damping as proportional to the mass and stiffness matrix, as in Equation 2.25.

$$\boldsymbol{C} = \eta \boldsymbol{M} + \delta \boldsymbol{K}. \tag{2.25}$$

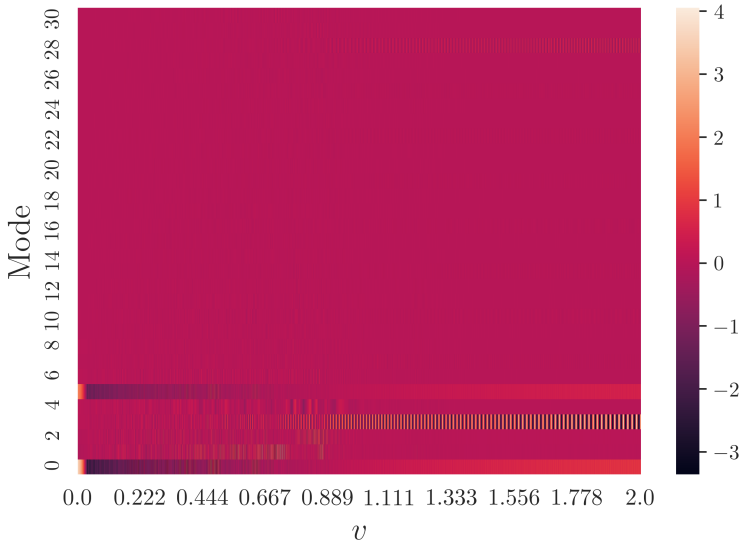
Rayleigh damping differs from the damping in the BKP model, Equation 3.1, where there is only one element in the damping matrix, namely the $(N+1, N+1)$ element between the pad and the top loader plate, which is equal to c_p.

Introducing Rayleigh damping a challenge is to choose the constants η and δ. To obtain an understanding of how mass and stiffness proportional damping affect the system, both of them can be adjusted. In this project, the primary approach has been to adjust the two constants separately.

In a paper by Chowdhury and Dasgupta (2003), a method is developed to calculate δ and η by first estimating two values for ζ at two different frequencies using Equation 2.27. A simplified method was tested here. Equation 2.27 was used to calculate η and δ using two significant modes. Setting ζ to a chosen value and using the frequency from two significant modes, one obtains two equations with two unknowns which can be solved for η and δ. A drawback with this approach is that it can be challenging to choose significant modes and to guess or estimate values for ζ.



(a) Displacement heat map.



(b) Displacement heat map with subtracted mean.

Figure 5.8: Heat map of modal contribution to displacement, with and without the mean value subtracted for each modal displacement vector. (Run specifications: Appendix A.8.)

Chapter VI

Results

This chapter will provide the thesis results. It starts by providing a stick-slip analysis for the BKP model. Secondly, a more extensive normal mode study will be given. This analysis will demonstrate the different mode shapes of the BKP model before using these to view what modes are activated at different slider velocity regions of the model. The next three sections will compare how the model behaves at increasing and decreasing slider velocities, for different seeds, and it will compare the BK and BKP mode, all in terms of eigenmodes. Section 6.5 shows how the model depends on the number of blocks. Finally, Section 6.6 shows how different values of the Rayleigh damping constants affect the BKP model.

6.1 Stick-Slip Analysis

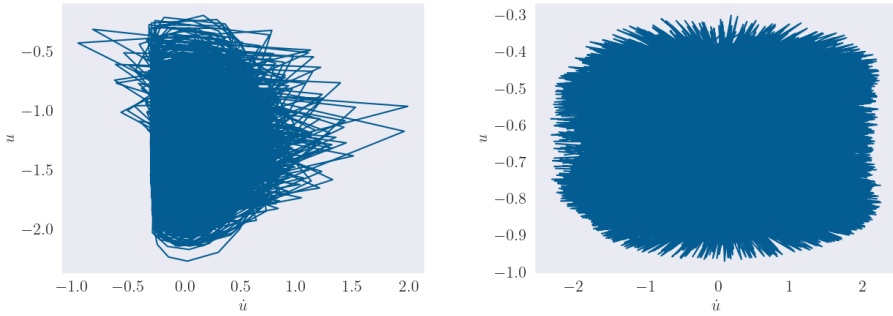
It was desired to see how the stick-slip behaviour varies with slider velocity. This is something both Standnes (2019) and Corrado and Mascia (2016) has proved is happening for the regular Burridge-Knopoff model.

Figure 6.1(a) and (b) shows the block position, u, plotted against block velocity, $\dot{u}$, for block number 50. The figures shows two different subsections of the slider velocity range for the same run as in Figure 5.1. Figure 6.1(a) shows a subsection of the data where the velocity is low, i.e. varies from 0.3 to 0.2. Figure 6.1(a) clearly shows that the stick-slip behaviour is present, with the straight line representing the area in which the block velocity is equal to the slider velocity, and hence stick occurs. Figure 6.1(b) shows a subsection of the simulation where the slider velocity is high, i.e. varies form 1.7 to 1.6. Figure 6.1(b) shows that the stick-slip behaviour seen in Figure 6.1(a) is not present at larger velocities.

Figure 6.2 shows the average percentage of stick for each of the 100 blocks for different simulations, where each simulation is run with a constant slider velocity. Note that the graph is scaled by a factor of ten each tenth decimal place, as seen at the v-axis on the figure.

The figure shows that the percentage of stick-slip behaviour is relatively high at low velocities and decreases as the slider velocity increase. When the slider velocity is low, below 0.01, the figure shows that the stick percentage reaches almost 10%. It also shows that when the slider velocity is higher, above 0.9, the stick percentage is close to 0%. There is also a drop in stick percentage at the constant velocities 0.005 and 0.006, which does not follow the usual trend.

See Appendix D for the same type of plot made using a different seed. The behaviour with a different seed shows the same tendency.



(a) Slider velocity range from 0.3 to 0.2.

(b) Slider velocity range from 1.7 to 1.6.

Figure 6.1: Block position plotted against block velocity at two different subsections of the slider velocity for block number 50. (Run specifications: Appendix A.1.)

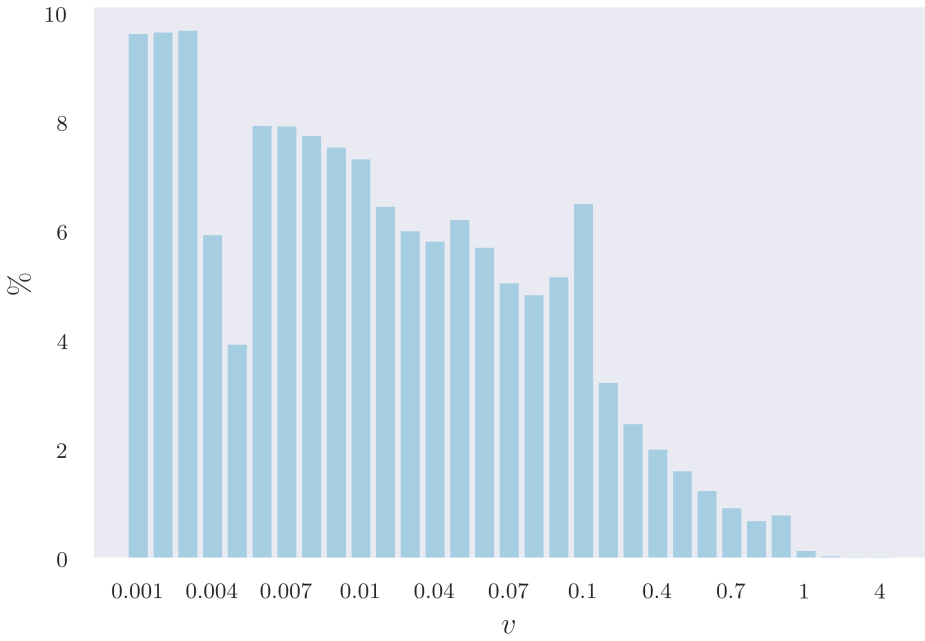


Figure 6.2: Percentage of stick for 31 runs ran at different constant slider velocity. Note that the graph is scaled by ten each tenth decimal place, as seen on the v-axis on the figure. (Run specifications: Appendix A.9, with varying start and end speed according to the figure.)

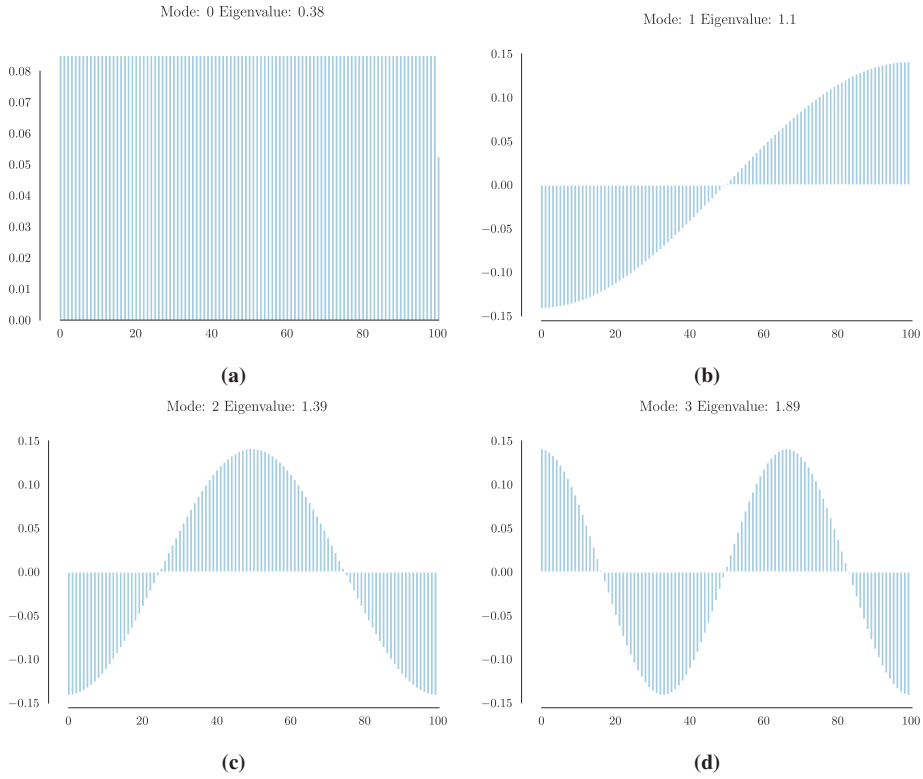
6.2 Normal Mode Analysis

6.2.1 Mode Shapes

Figure 6.3 shows the first eight mode shapes for the BKP model. As mentioned in Section 5.4.5, the first 100 bars in the bar plot represent the blocks, and the 101^{st} bar in the bar plot represents the pad. Mode 0, represented in Figure 6.3(a), shows the mode where all the blocks and the pad move in the same direction. Mode 1, illustrated in Figure 6.3(b), follows a sinusoidal behaviour with half of a wave cycle. Mode $2-4$, Figure 6.3(c)-6.3(e), also show sinusoidal behaviours where the number of wave cycles increase with one half each mode.

Mode 5, Figure 6.3(f), shows a different behaviour, where all the blocks move in one direction, while the pad moves in the opposite. Mode 6 and 7, Figure 6.3(g) and 6.3(h), follow the behaviour as before, where the motion is sinusoidal with increasing amounts

of sine waves. This pattern continues more or less for each of the following modes. The mode shape of all 101 modes can be seen on the web page [1]. Figure 6.4 shows the four modes corresponding the system's highest frequencies. For these modes, the sign of each value in the eigenvectors change for every value. The overall shape for these modes, given by the amplitude of these alternating values, is also shaped as a wave with low frequency.



(a) **(b)** **(c)** **(d)**

[1] https://theamartine.github.io/Burridge-Knopoff-Pad-Model/

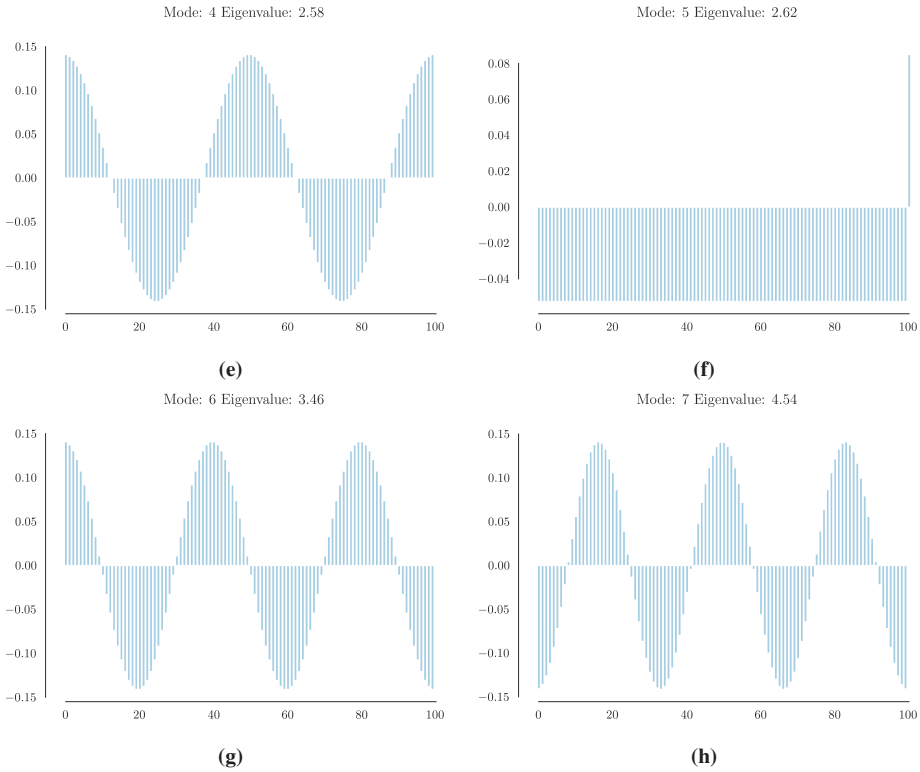


Figure 6.3: Bar plot of the first eight mode shapes. (Run specifications: Appendix A.1.)

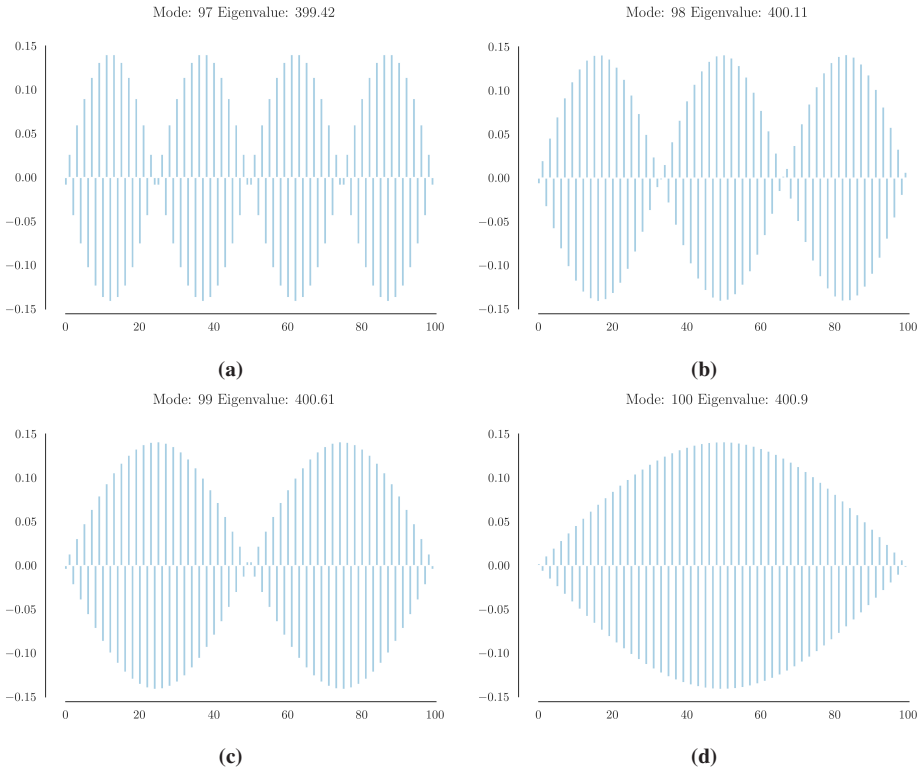


Figure 6.4: Bar plot of the four modes corresponding to the system's highest frequencies. (Run specifications: Appendix A.1.)

6.2.2 Modal Contribution to the BKP Model

Figure 5.1 showed the pad position versus velocity for a run where the slider velocity decreases from 2 to 0. The results in the rest of this section will use the same simulation as in Figure 5.1, with run specifications found in Appendix A.1.

Figure 6.5 shows a heat map of all hundred modes for the full slider velocity range 2 to 0, which clearly demonstrates that the two lower modes are dominant. In accordance with Section 5.4.3, the lower modes are of most interest and the rest of the thesis will therefore focus on the lowest 21 modes.

Figure 6.6 shows a plot of friction versus slider velocity and two heat maps of the modal contribution to displacement and velocity for the lowest 21 modes. Appendix E shows the contribution to energy. The next paragraphs will focus on the three following subsections

of Figure 6.6:

1. The area of high velocities where there are two visible modes.
2. The area where there is a transition to higher friction amplitudes around $v = 0.7$, and the heat maps show a transition to a different behaviour.
3. The area of low slider velocity.

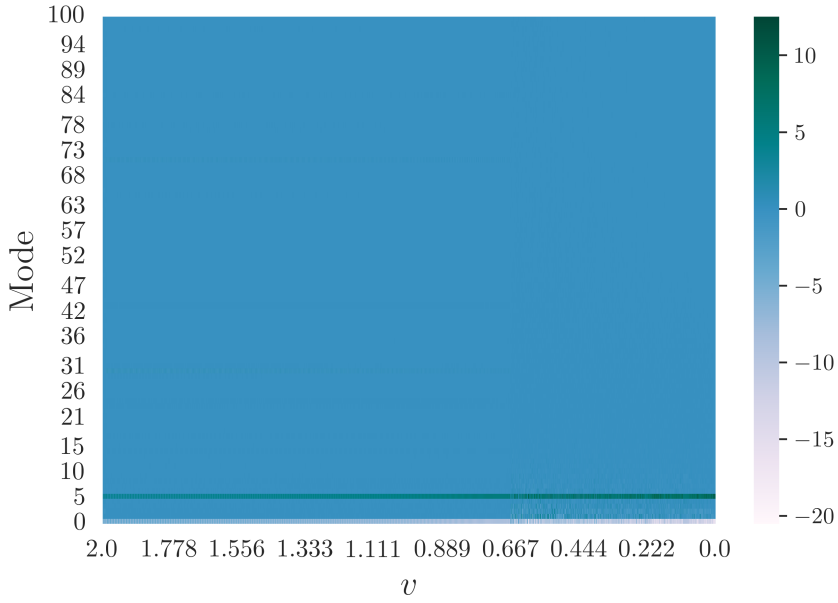
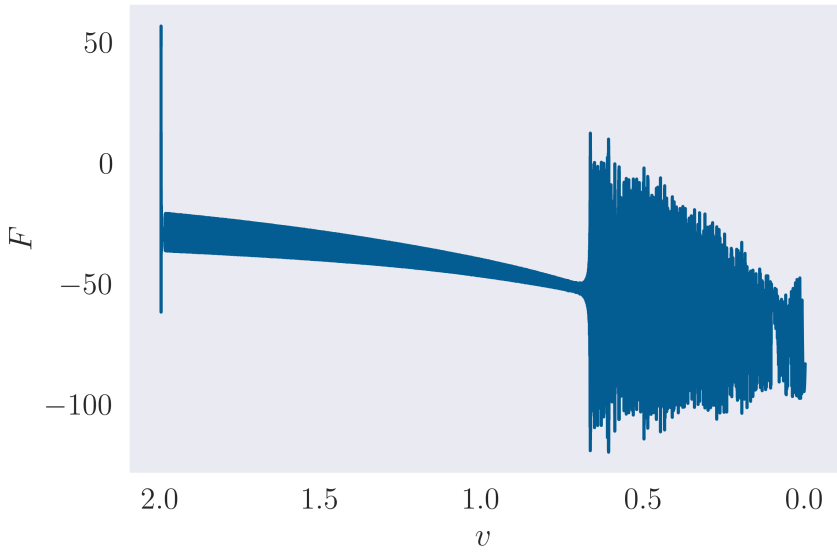
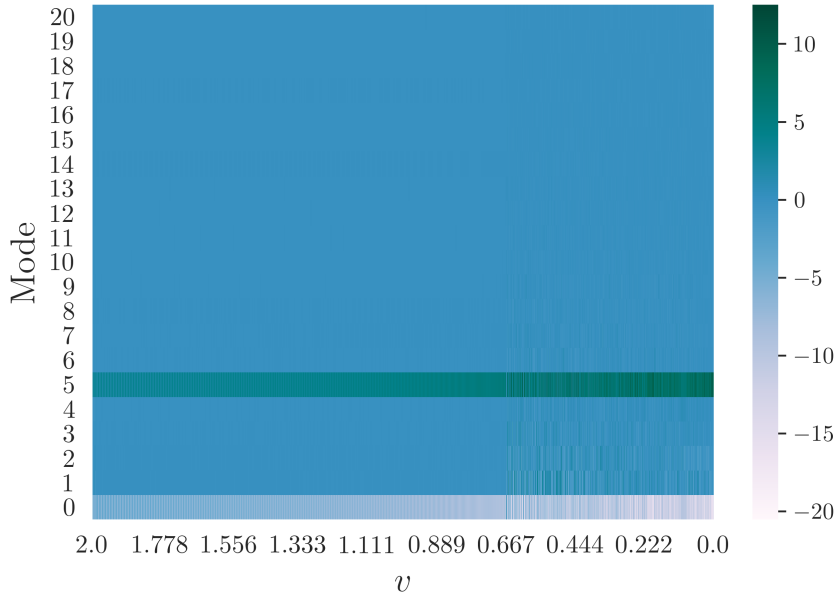


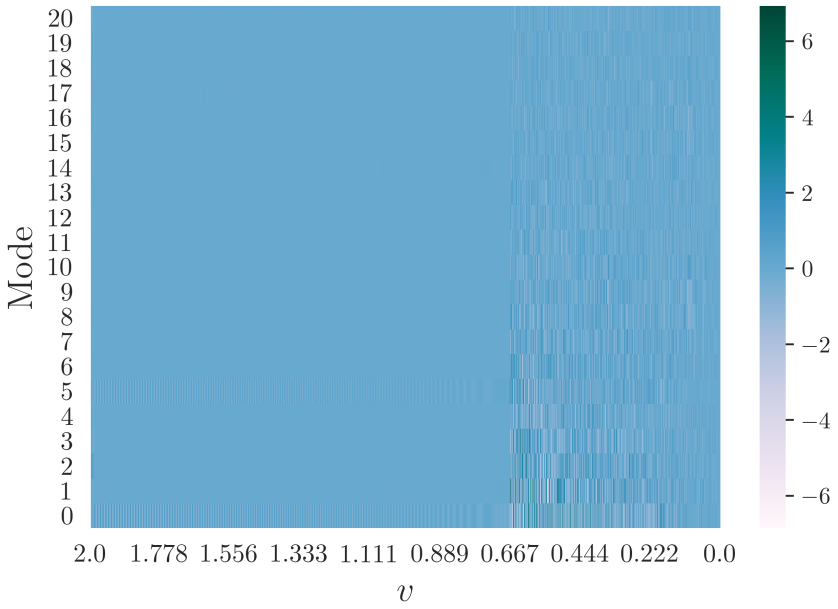
Figure 6.5: Each modes contribution to displacement for all 101 modes.



(a) Friction.



(b) Displacement heat map.



(c) Velocity heat map.

Figure 6.6: Friction versus slider velocity and heat maps of the modal contribution to displacement and velocity for the 21 lowest frequencies as the velocity decreases from 2 to 0 for seed 101. (Run specifications: Appendix A.1.)

High Slider Velocity Range

The first area to discuss is the high slider velocity range, from 1.9 to about 0.8, where Figure 6.6(a) shows that the friction has a low amplitude. Figure 6.6(b) shows that that eigenmodes 0 and 5 are the two most active modes and have an extensive contribution to displacement. Note from the mode shapes in Figure 6.3 from the previous section that mode 0 is when the blocks and pad move in the same direction and mode 5 is when the blocks move in one direction and the pad in the opposite. The eigenvalues of these two modes are 0.38 and 2.62, respectively. From Equation 2.9, this corresponds to the two frequencies 0.098 and 0.258.

Figure 6.6(c), the modal contribution to velocity, shows that these two modes do not show the same dominance with regards to velocity.

Figure 6.7 shows a displacement heat map in the velocity range 1.7 to 1.5 where mode 0 and 5 are removed. The figure shows that several modes are active, but the activation of each mode is continuous in the sense that modes are activated to the same degree throughout the whole slider velocity region.

The latter is even more clear in Figure 6.8, which shows heat maps for the slider velocity range 1.700 to 1.698. Figure 6.8(a) shows the contribution to displacement for the 21 lowest modes. Figure 6.8(b) shows the modal contribution to displacement when mode 0 and 5 are removed. Figure 6.8(c) shows the modal contribution to the energy in the system when mode 0 and 5 are removed. All three figures show that there are few jumps in which modes are activated in the slider velocity range 1.700 to 1.698, and the motion is seemingly periodic.

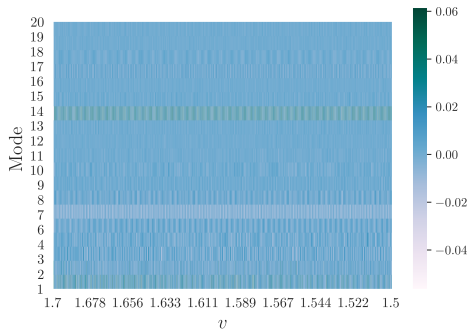
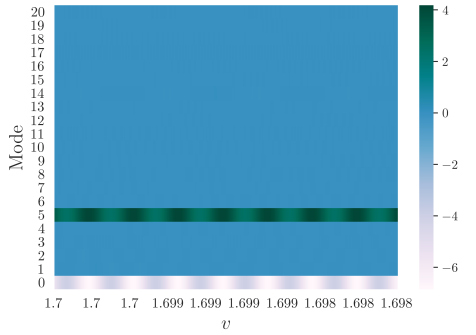
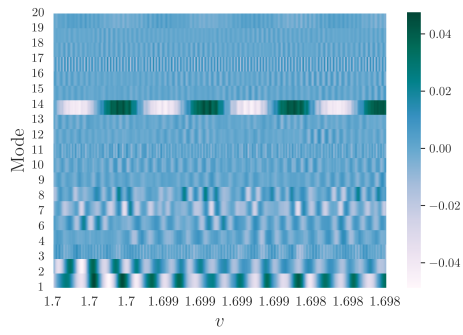


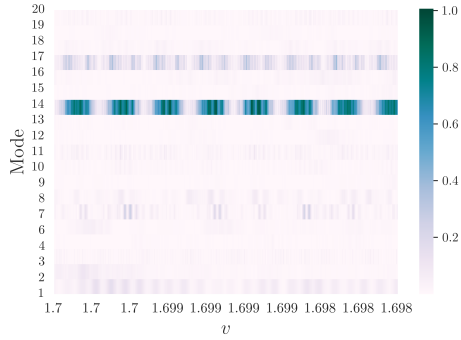
Figure 6.7: Each eigenmodes contribution to displacement when the slider velocity decreases continuously from 1.7 to 1.5 with mode 0 and 5 removed. (Run specifications: Appendix A.1.)



(a) Displacement heat map.



(b) Displacement heat map. Modes 0 and 5 removed.

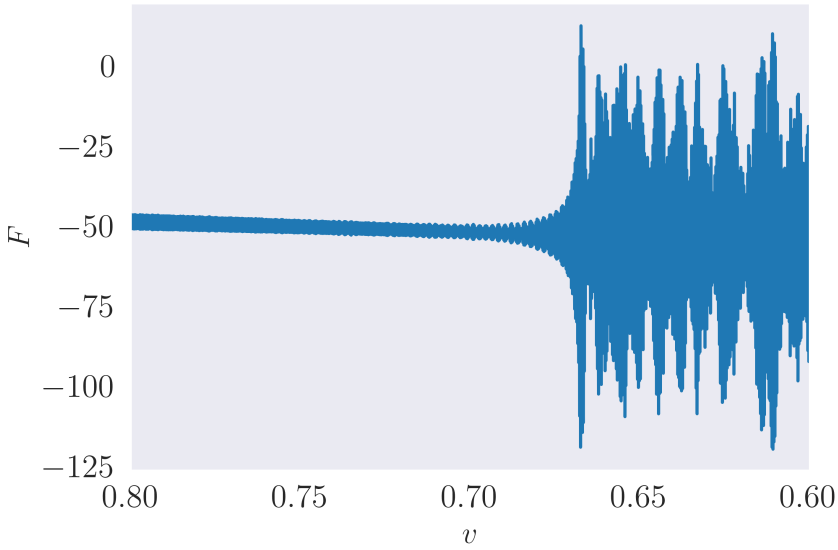


(c) Energy heat map with removed modes 0 and 5.

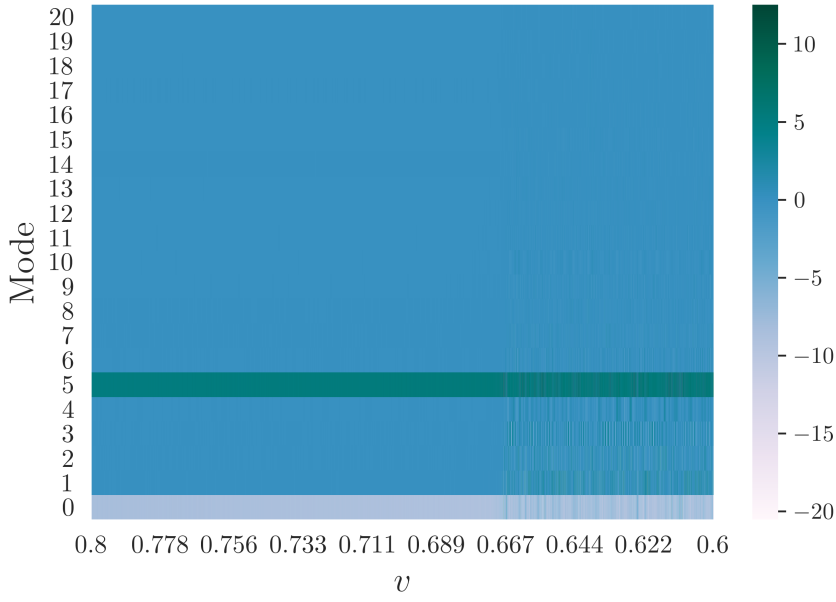
Figure 6.8: Modal contribution heat maps to displacement and energy in the velocity range 1.700 to 1.698. (Run specifications: Appendix A.2.)

Transition to More Active Modes

The second slider velocity range to examine is in the transition to the suddenly larger friction amplitude and more activated modes. Figure 6.9(a) shows the friction plotted against slider velocity in the slider velocity range $v \in [0.8, 0.6]$, showing the sudden increase in friction amplitude at $v \approx 0.67$. Figure 6.9(b) shows how there is also a change in what mode are active at $v \approx 0.67$. This is also clear from the velocity heat map in Figure 6.6(c). At $v \approx 0.67$, Figure 6.9(b) shows that the previously distinct modal contributions fades to a section where many modes contribute to displacement in a less ordered manner. After this point, mode 0 and 5 still stands out, but many other modes are also activated and contribute to displacement and velocity.



(a)



(b)

Figure 6.9: Figure a) shows each eigenmodes contribution to displacement of the blocks and pad when the slider velocity decreases continuously from 0.8 to 0.6. Figure b) shows friction plotted against slider velocity in the same range. (Run specifications: Appendix A.1.)

Low Slider Velocity Range

Figure 6.10 shows zoomed heat maps of the slider velocity section 0.4 to 0.2. In this area, the friction has a lower amplitude than in the slider velocity region $v \in [0.8, 0.6]$, but the friction looks more chaotic compared to the higher slider velocity ranges with $v > 0.67$. Figure 6.10 shows the same modal behaviour as the lower slider velocity region of Figure 6.9(b), where several modes contributes to displacement.

Figure 6.11 shows heat maps for the slider velocity region 0.400 and 0.398. From Figure 6.11(a), mode 0 and 5 are consistently activated though the slider velocity region. However, the energy heat map in Figure 6.11(b), where mode 0 and 5 are removed from the plot, shows that there are large jumps in what modes that are activated. Mode 1 is activated in the beginning, before the figure shows a jump to the activation of mode 2, which again vanishes after some time. This is a sign that the model exhibits chaotic motion.

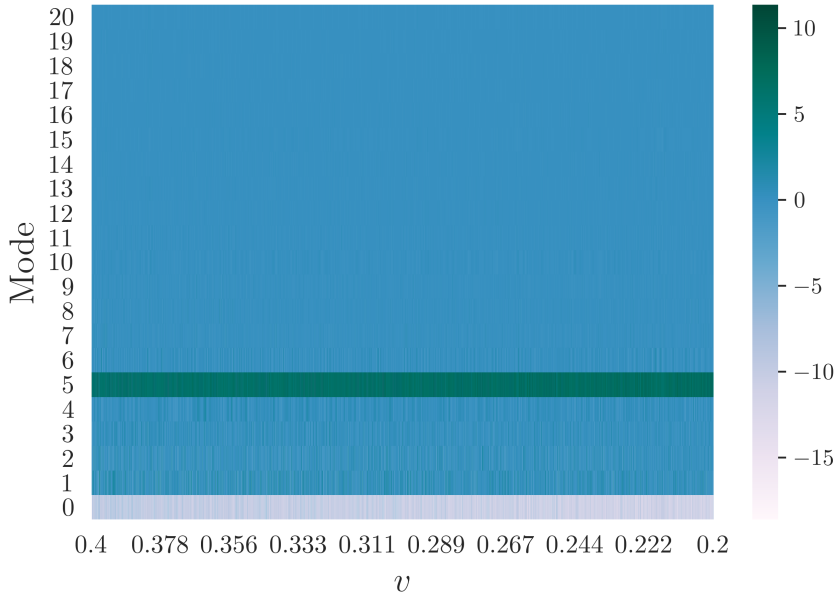
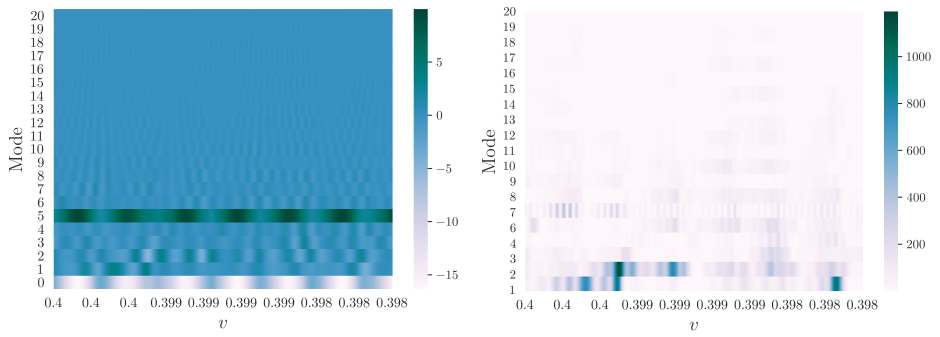


Figure 6.10: Each eigenmodes contribution to displacement when the slider velocity decreases continuously from 0.4 to 0.2. (Run specifications: Appendix A.1.)



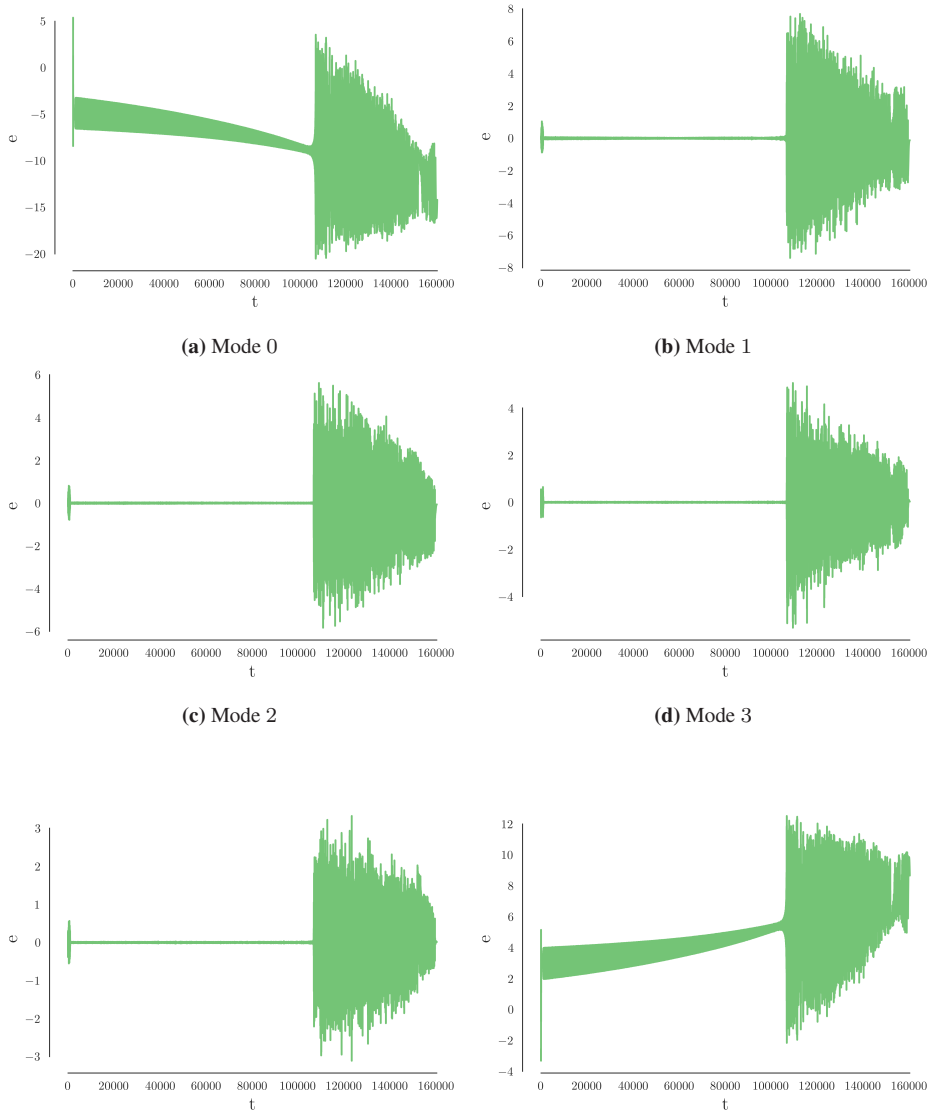
(a) Displacement heat map.

(b) Energy heat map with removed modes 0 and 5.

Figure 6.11: Modal distribution heat maps to displacement and energy in the velocity range 0.400 to 0.398. (Run specifications: Appendix A.2.)

6.2.3 Displacement at Each Mode

Figure 6.12 shows the dot product between displacement and mode shape plotted against time, for the first six modes. From Section 5.4.6 the parameter e denotes this dot product. Figure 6.12 shows how mode 0 and 5 differ from the rest. The displacement amplitude is also higher for these two modes, and opposed to the other modes, nor mode 0 and 5 have a mean value of zero.



(a) Mode 0

(b) Mode 1

(c) Mode 2

(d) Mode 3

(e) Mode 4

(f) Mode 5

Figure 6.12: Each modes contribution to displacement for the first six modes. (Run specifications: Appendix A.1.)

6.2.4 Seed Comparison

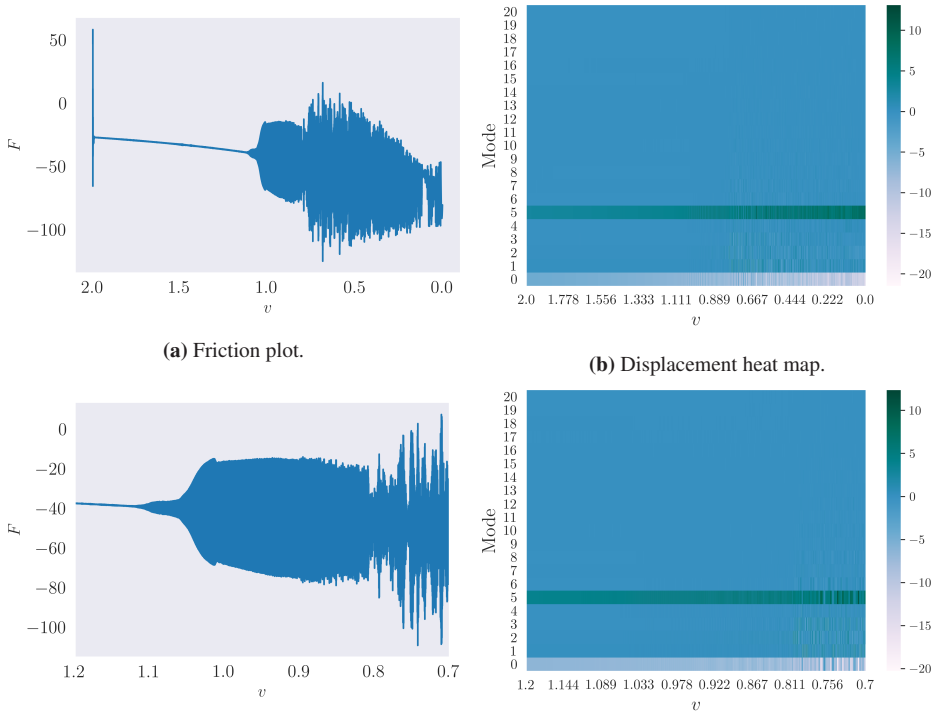
The former heat maps were taken from runs with seed 101. From the seed comparison in Section 5.2.1 and Figure 5.3, it is evident that for instance seed 100, 101 and 140 show different characteristics of how friction amplitude develops with slider velocity. It was therefore desired to see if the results from the previous section also yields different seeds. In this section, the displacement heat maps of seed 100 and 140 will be shown and compared to seed 101.

Figure 6.13(a) shows friction against slider velocity for seed 100. Figure 6.13(b) shows its respective displacement heat map. In the slider velocity range 1.1 to 0.8, the friction amplitude increase to a "medium" level, $F \in [-20, 80]$, before increasing to the same level as for seed 101, $F \in [10, -120]$ at $v \approx 0.7$. Figure 6.13(c) and (d) show a zoomed friction plot and heat map of this area. At $v = 1.1$, the friction amplitude increases, but, the heat map in Figure 6.13(d) does not show a large change in the modal contribution to displacement until the slider velocity is at $v \approx 0.8$, meaning that model looks to exhibit periodic motion in the area of "medium" increased friction amplitude.

Another difference between the behaviour of the friction force of seed 100 and 101 is the region of higher slider velocities, $v \in [1.9, 1.2]$. For seed 101, this region has a higher friction amplitude than for seed 100. Figure 6.14 shows what modes contribute to displacement in the velocity range 1.7 to 1.5 when mode 0 and 5 are removed from the plot. Even though both seeds shows that mode 0 and 5 are the most dominant to displacement, Figure 6.14 shows that mode 1 of seed 101 takes up much of the modal contribution to displacement, with a value of 0.25. This is opposed to mode 1 of seed 101, showed in Figure 6.7, which has a maximum value of 0.06.

Figure 6.15(a) shows friction against slider velocity in the range $v \in [2, 0]$ for seed 140. The heat map in Figure 6.15(b) shows the modal displacement in the range $v \in [1.7, 1.5]$, with mode 0 and 5 removed. In this velocity range, the friction amplitude is more similar to seed 101, but the heat map shows more similarity to seed 100.

Regardless of the differences, all seeds show the same general behaviour. At high velocities, the same modes contribute to displacement during long slider velocity ranges and there are no jumps between modes, for all three seeds. Similarly, at low slider velocities, the most activated modes change during short time intervals for all three seeds.



(a) Friction plot.

(b) Displacement heat map.

(c) Zoomed friction plot.

(d) Zoomed heat map.

Figure 6.13: Friction plot and displacement heat map for seed 100 with slider velocity decreasing from 2 to 0. (Run specifications: Appendix A.4.)

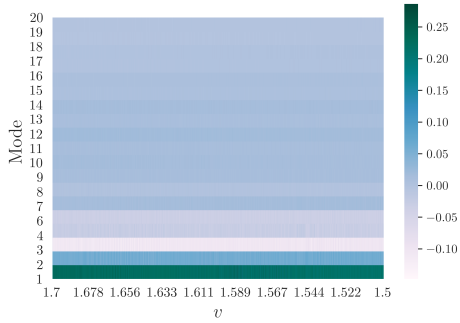
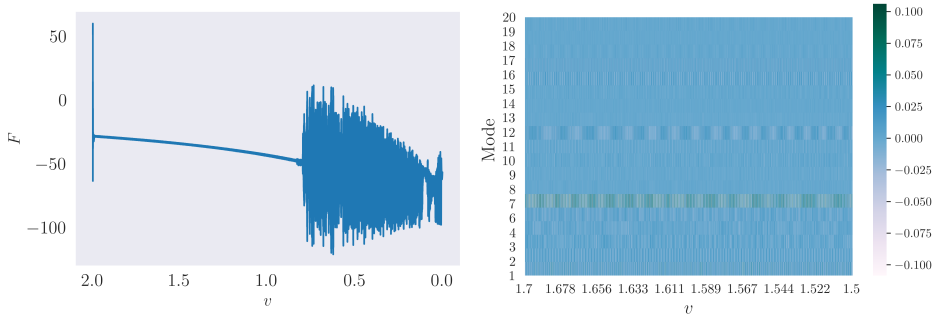


Figure 6.14: Displacement heat map for seed 100 in the velocity range 1.7 to 1.5. Mode 0 and 5 are removed. (Run specifications: Appendix A.4).



(a) Friction.

(b) Displacement heat map.

Figure 6.15: Friction plot of the whole velocity range and displacement heat map for $v \in [1.7, 1.5]$ for seed 140. (Run specifications: Appendix A.6.)

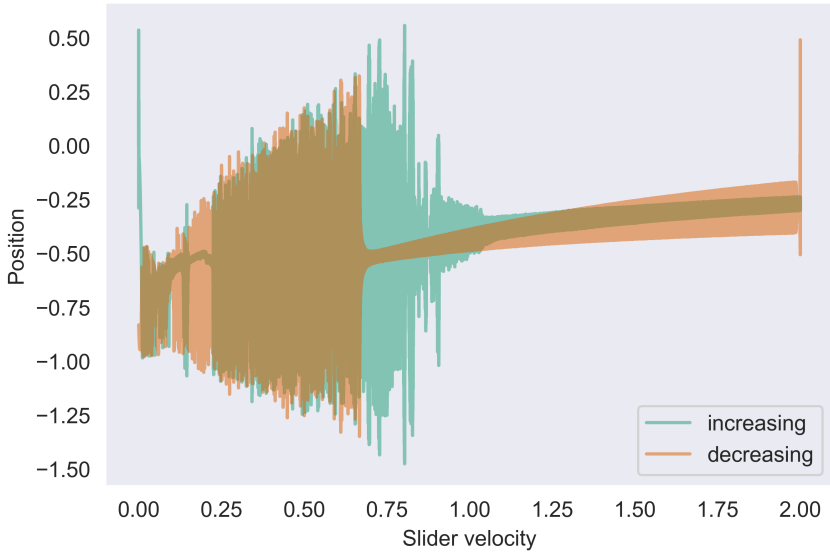
6.3 Increasing Slider Velocity

As mentioned in Section 3.5.1, Standnes (2019) studied the difference between increasing and decreasing slider velocity, which has been further analyzed with the extension of studying the normal modes. This has been done in order to understand the general behaviour of the model better, and to study if the results from decreasing slider velocity also holds for increasing slider velocity. Standnes (2019) highlighted four slider velocity regions to compare. This section will show a comparison of what modes are present in some of these same regions. Appendix F compares all the same regions as in the thesis of Standnes (2019).

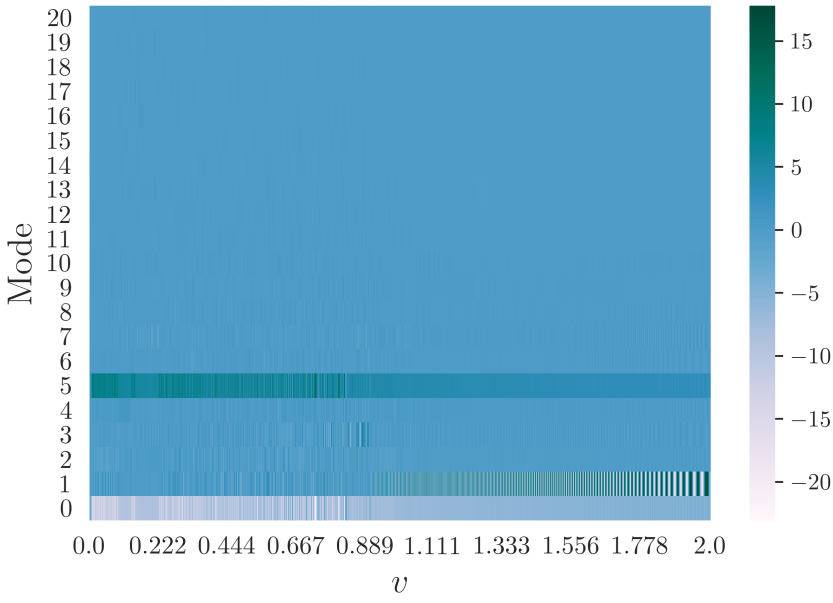
Figure 6.16(a) shows the pad position as a function of increasing and decreasing slider velocity and Figure 6.16(b) shows the displacement heat map for the increasing case. For the low regions of slider velocity, both the increasing and decreasing cases show a modal behaviour where many modes are present, and it is hard to distinguish the differences in the two cases. Even in the region around $v = 0.2$ where the position plot in Figure 6.16(a) shows a clear difference between the two simulations, there are little distinctions in the heat maps.

In the velocity range where the pad position of the two cases differ most, $v \in [0.67, 1.1]$, the modal displacement also shows a clear difference. Figure 6.16(b) shows a chaotic behaviour with many present modes from around $v \approx 0.8$, where the friction amplitude also has a sudden increase. As for decreasing slider velocity, the heat map for increasing slider velocity is consistent with its corresponding friction plot, showing that as the friction amplitude has a sudden increase, the heat map shows chaotic behaviour.

When the slider velocity is above $v \approx 1.2$ the friction amplitude is low for both cases. Recall from Figure 6.5 that the decreasing slider velocity case showed a periodic motion with modes 0 and 5 as the most activated when $v > 1.2$. Figure 6.16(b) shows that with increasing slider velocity, mode 1 also has a large contribution to displacement. Appendix F shows zoomed heat maps on the regions discussed.



(a) Decreasing versus increasing slider velocity.



(b) Modal displacement as a function of increasing slider velocity.

Figure 6.16: Increasing slider velocity.(Run specifications: Appendix A.1(decreasing) and A.7 (increasing)).

6.4 The BK Versus the BKP Model

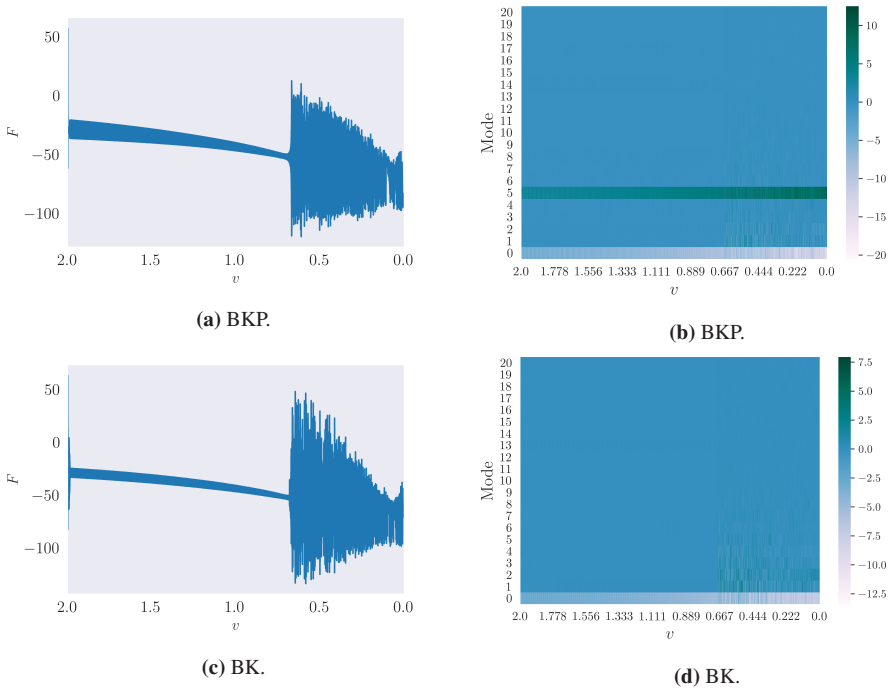
It was desired to analyze the differences and similarities between the BK and BKP model to get a better understanding of what the pad brings to the system. As mention in Section 3.5.2, Standnes (2019) compared the BK and the BKP model. He primarily focused on the slider velocity range $v \in [0.1, 0.25]$ for increasing slider velocity where there are dips in the friction force, as mentioned in Section 3.5.2. This section will address this same velocity range.

When creating heat maps for the BK model, both the mode shapes of the model itself and the mode shapes of the BKP model has been used. The mode shapes of the BK model are found from the mass and stiffness matrices in Section 5.4.2 and are seen in Appendix H. As the BK model does not have a pad, mode 5 follows the sinusoidal behaviour like the other modes.

Figure 6.17 and 6.18 show a comparison between the BK and BKP model when the slider velocity is decreasing. Figure 6.17 shows the whole velocity range from 2 to 0. Figure 6.18 shows a section of the slider velocity range $v \in [0.1, 0.25]$. One difference between the BK and BKP model in the figures is that the BKP model has a slightly larger displacement amplitude, especially in the lower rages of slider velocity, which can be seen on the bar on the right-hand side of the heat maps and in the friction plots. Except from the latter, the two models shows a similar general behaviour.

Figure 6.19 and 6.20 show a comparison between the BK and BKP model when the slider velocity is increasing. Figure 6.19 shows the whole slider velocity range from 0 to 2 and Figure 6.20 shows the slider velocity range $v \in [0.1, 0.25]$. Similar to the case of decreasing slider velocity, the BKP model shows higher mode amplitudes than the BK model for increasing slider velocity, seen on the bar on the right hand side of the heat maps. In the higher ranges of slider velocity, the BK model shows the highest modal amplitude. In the latter region, Figure 6.19 shows that for the BK model, mode 4 is activated to a large degree, while in the BKP model, mode 1 is more active.

As when comparing increasing and decreasing slider velocity, this comparison show that, despite some differences, the response of the models are in most ways similar. The same type of figures as showed in this section made using a different seed is found in Appendix H.



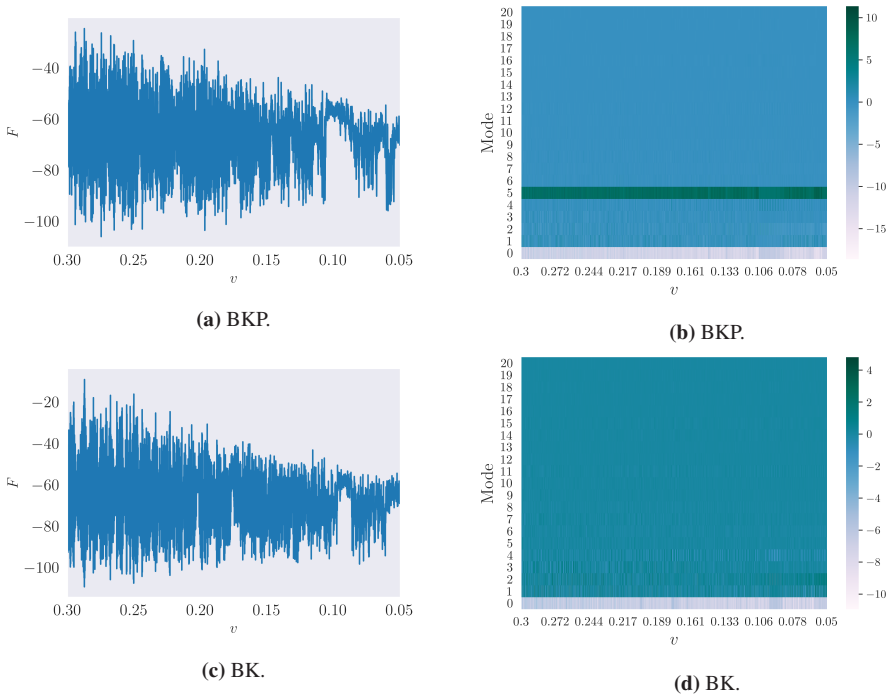
(a) BKP.

(b) BKP.

(c) BK.

(d) BK.

Figure 6.17: Displacement heat maps of the the BK and BKP model with decreasing slider velocity from 2 to 0. (Run Specifications: Run A.10.)



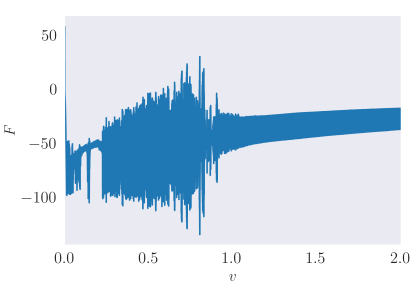
(a) BKP.

(b) BKP.

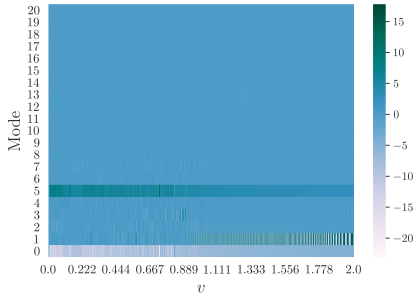
(c) BK.

(d) BK.

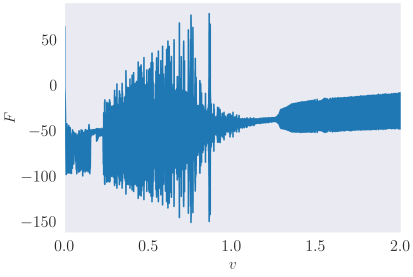
Figure 6.18: Plots of the BK and BKP model for decreasing slider velocity in the slider velocity subsection 0.3 to 0.05. (Run Specifications: Run A.10.)



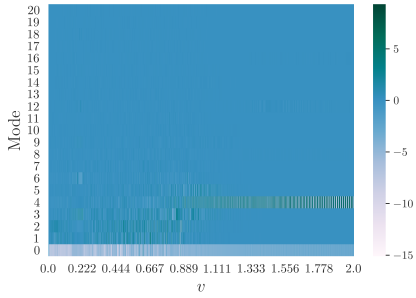
(a) BKP.



(b) BKP.

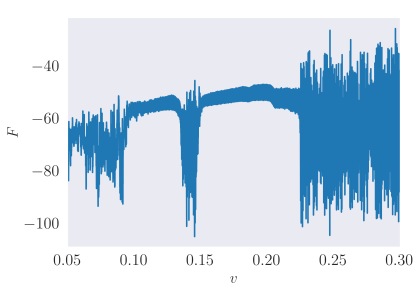


(c) BK.

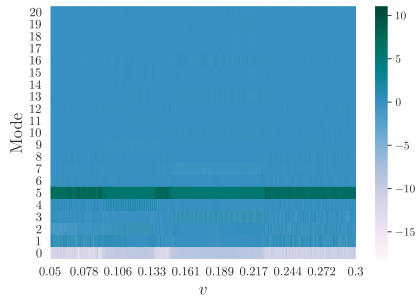


(d) BK with BK mode shapes.

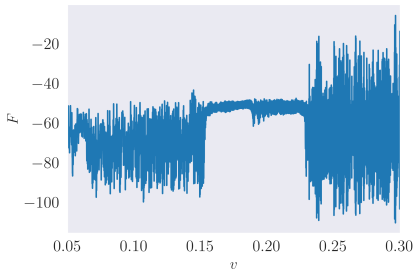
Figure 6.19: Heat maps for increasing slider velocity in the range 0 to 2 for BK and BKP. (Run specification: Appendix A.11.)



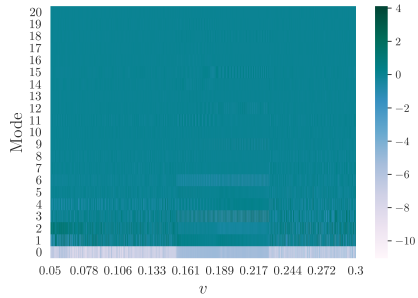
(a) BKP.



(b) BKP.



(c) BK.



(d) BK with BK mode shapes.

Figure 6.20: Increasing slider velocity in the range 0.5 to 0.3 for BK and BKP. (Run specification: Appendix A.11.)

6.5 Number of Blocks

It was desired to study how the number of blocks affects the BKP model. As a comparison it was also desired to study the number of blocks in the BK model. Note that unless other is specified, all results refer to the BKP model. Recall from Section 3.3 that the spring constants and block masses are dependent on the number of blocks. When decreasing the number of blocks, the total weight of the model, each block's axial stiffness, and the total stiffness between the blocks and pad will remain constant.

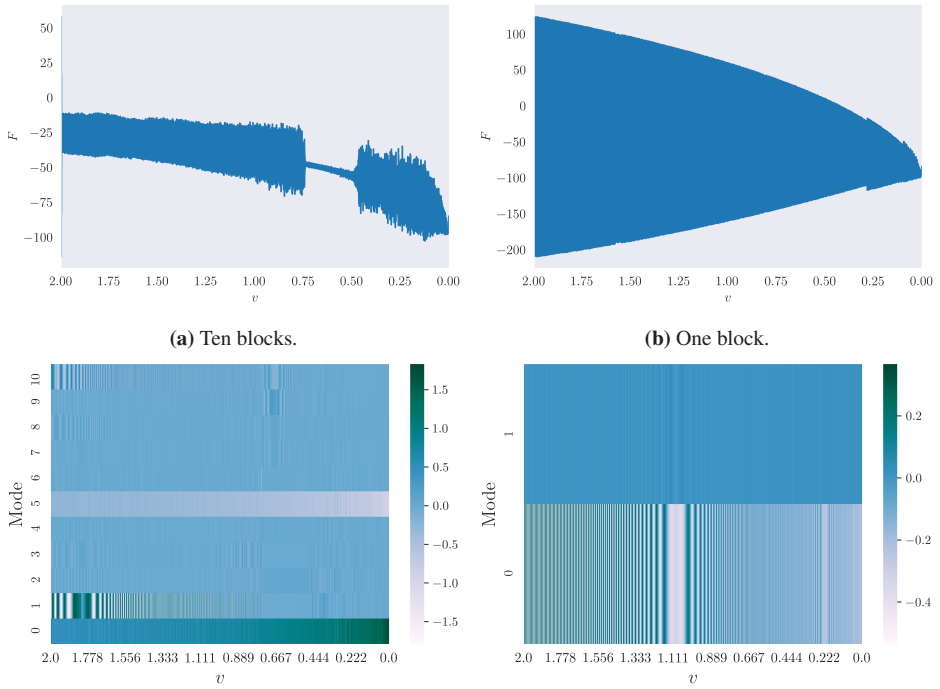
As the stiffness between the pad and each block gets larger when decreasing the number of blocks, simulating with one block becomes almost analogous to having no blocks and hence having the SDOF model showed in Section 2.3. Figure 6.21 shows a comparison between the friction force and displacement heat maps for one and ten blocks. Comparing Figure 6.21(a) and (b), the friction amplitude when having one block is generally much larger than when having ten blocks, especially in higher ranges of slider velocity.

Four notes should be made from Figure 6.21. Firstly, Figure 6.21(b) shows a high friction amplitude for high slider velocities, but the heat map of Figure 6.21(d) indicates periodic motion. The periodicity is also indicated in the friction plot, given its even shape. Secondly, Figure 6.21(d) at $v \approx 1.1$ shows a region of lower modal displacement amplitude for mode 0 which is not present in the friction plot 6.21 (b).

Thirdly, Figure 6.21(c) shows that mode 5 is activated to large degree. Figure 6.22 illustrates the mode shape of mode 5 when using 10 blocks, showing that in this mode, all blocks move in the same direction and the pad moves in the opposite, similar to the mode shape of mode 5 when having 100 blocks. Lastly, the heat map with 10 blocks shows alternations between what modes are most active throughout the run, suggesting chaotic motion. This is in contrast to when having 100 blocks, where the heat maps show little alternation between modes in the higher slider velocity intervals.

Figure 6.23 shows friction plotted against slider velocity for several simulations with an increasing number of blocks for two different seeds. The figure shows that there is a difference in the response of the model depending on the number of blocks used. With 100 blocks, it is assumed that the system is independent of the number of blocks. However, it looks like the medium amplitude region around slider velocity $v = 1$ for seed 100 (and other seeds, see Section 5.2.1) at 100 blocks, becomes clear when increasing the number of blocks above 100 also for those seeds that do not have this section at 100 blocks. Neither seed 30, 101 nor 140 shows this section with 100 blocks, but all of them show this section with 130 blocks. This is showed in Figure 6.23(a) for seed 101 and in Appendix G for seed 30 and 140. Figure 6.23 also shows that the general behaviour when exceeding from 30 to 40 blocks, is very similar to the behaviour at 100 blocks, indicating that the system converges at a lower number of blocks than 100.

Figure 6.24 shows a comparison between different number of blocks for two different seeds of the BK model. The BK model shows clear differences from the the BKP model when the number of blocks is below 100. It shows that the friction amplitudes are high at high slider velocity, similarly to what the BKP model shows when having only one block. It also shows that when $v < 1$ a different behaviour begins. Figure 6.25 shows a displacement heat map for the BK model ran with 10 blocks. It shows that even tough the friction amplitude is high when $v > 1$, the chaotic alternation between modes starts when the slider velocity is approximately $v = 1$ and continues until the slider velocity is zero.



(a) Ten blocks.

(b) One block.

(c) Ten blocks.

(d) One block.

Figure 6.21: Comparison between one and ten blocks. Figure a) and c) shows friction as a function of slider velocity, while Figure b) and d) shows displacement heat maps for different modes.

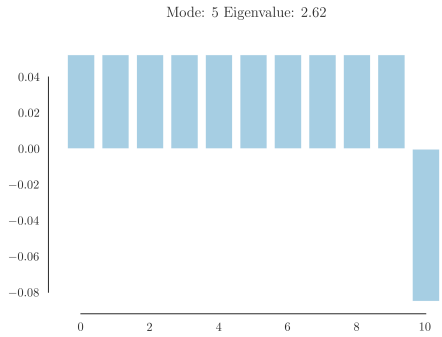
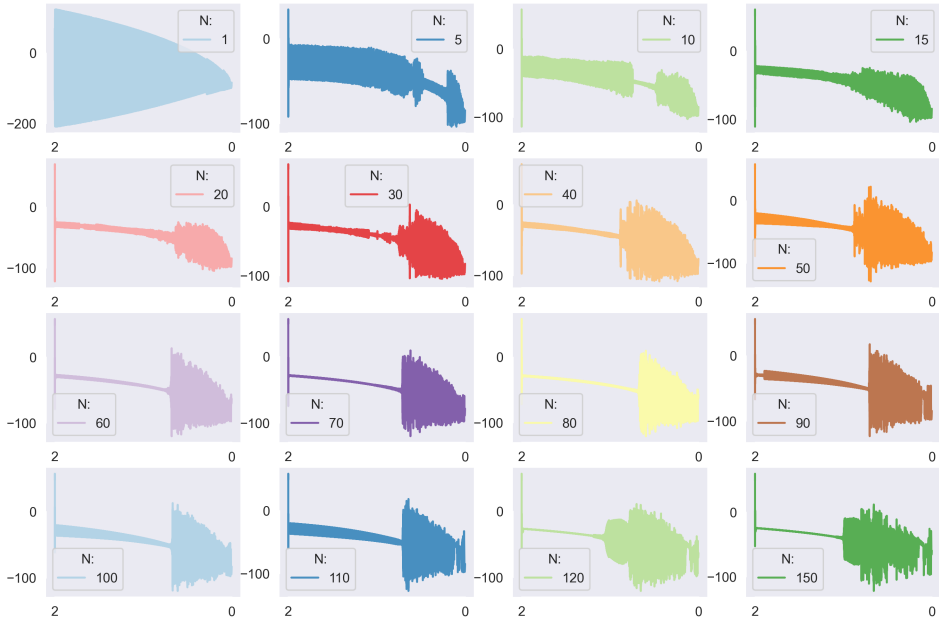
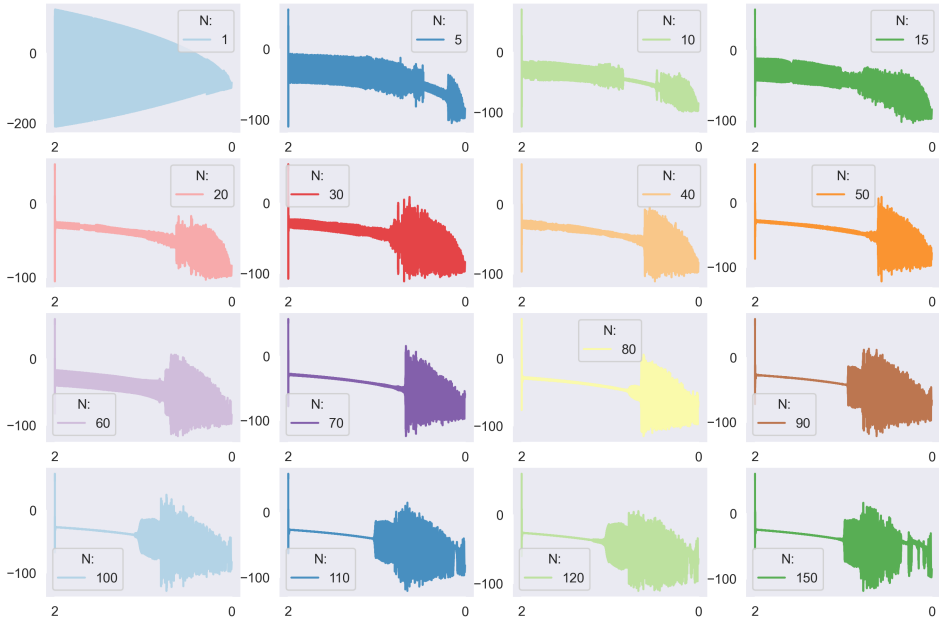


Figure 6.22: Mode shape of mode 5 with 10 blocks.

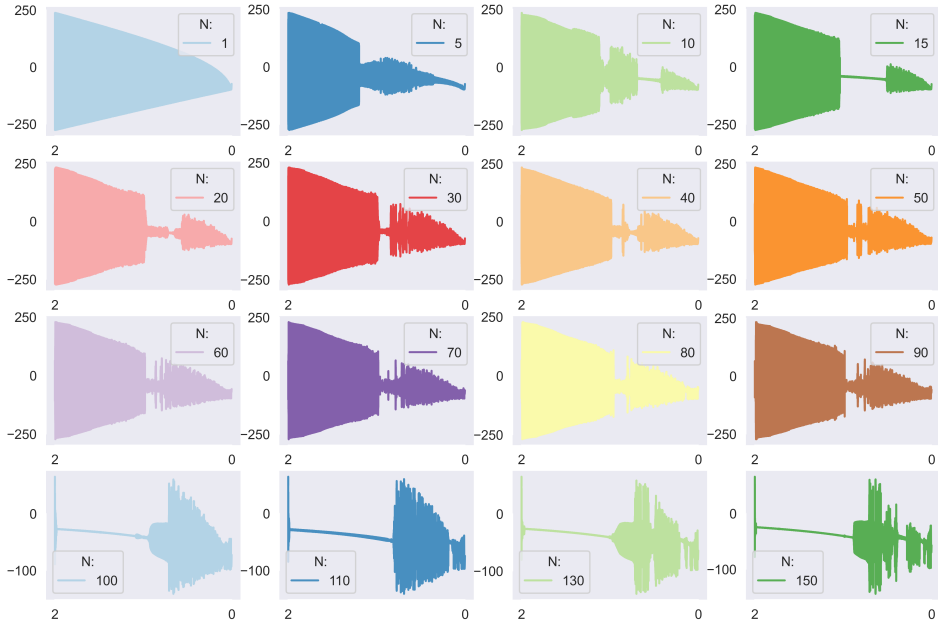


(a) Seed 101.

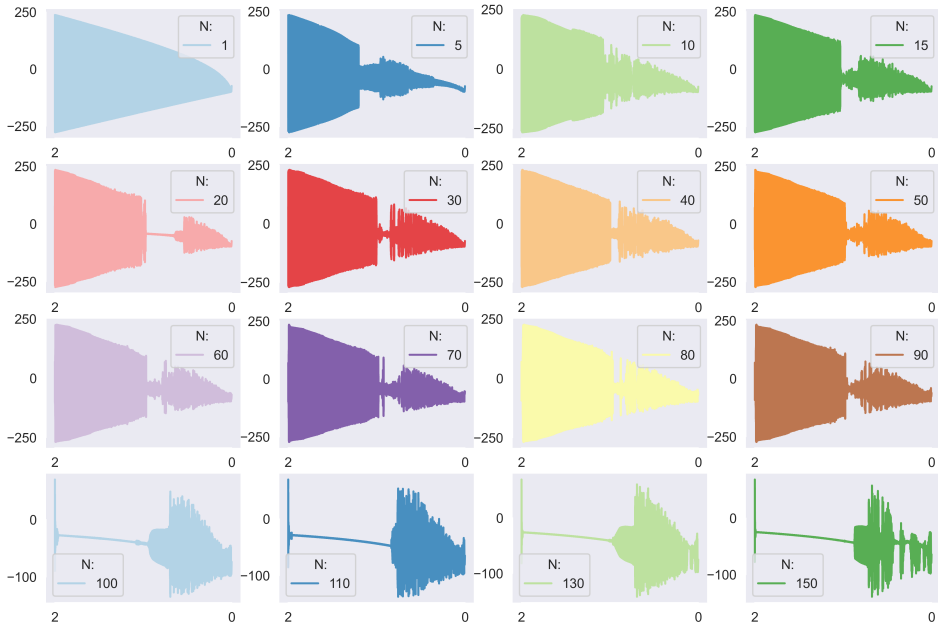


(b) Seed 100.

Figure 6.23: Comparison of different number of blocks for two different seeds for the BKP model.



(a) Seed 101.



(b) Seed 100.

Figure 6.24: Comparison of different number of blocks for two different seeds for the BK model.

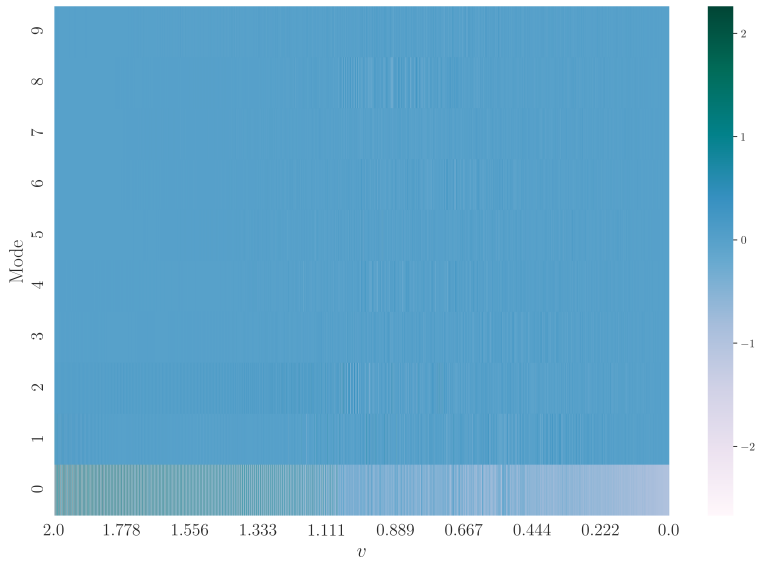


Figure 6.25: Displacement heat map of the BK model with 10 blocks.

6.6 Rayleigh Damping

In order to study how damping affects the model, this section will show the results from the implementation of Rayleigh damping. The next two subsection will show different values for the stiffness and mass proportional damping constants separately. In these subsections, the damping element with damping coefficient c_p between the pad and the stationary plate as in the original BKP model from Equation 3.1 is remained unchanged, meaning that the mass and stiffness proportional damping are added in addition to c_p. In the figures in this section, the latter is specified by the value of $c_p = 23.57$, which is the value specified in the parameter list in Table 3.1. Section 6.6.3 will show one example where both the mass and stiffness proportional damping is tested together while c_p is set to 0.

6.6.1 Stiffness Proportionality

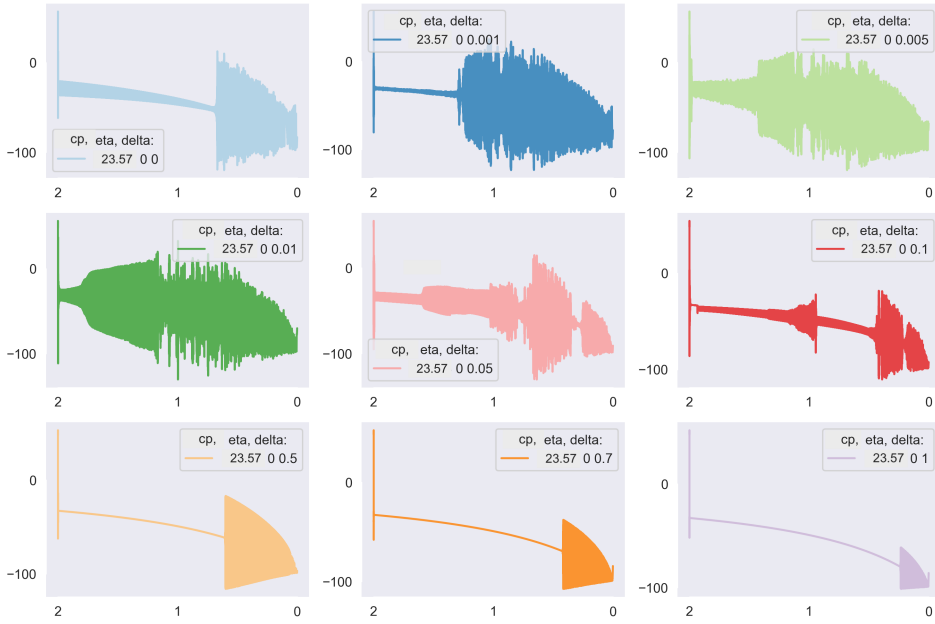
Figure 6.26 shows friction plotted as a function of slider velocity when the model is run with nine different values of the stiffness proportional damping factor δ. The same plots made with different seeds is shows in Appendix I and shows similar responses.

The last row in Figure 6.26(a) when δ is in the range $\delta \in [0.5, 1]$ shows that increasing the stiffness proportional damping leads to lower amplitudes, i.e., more damped behaviour, in the higher ranges of velocity. This stiffness proportional damping behaviour is consistent

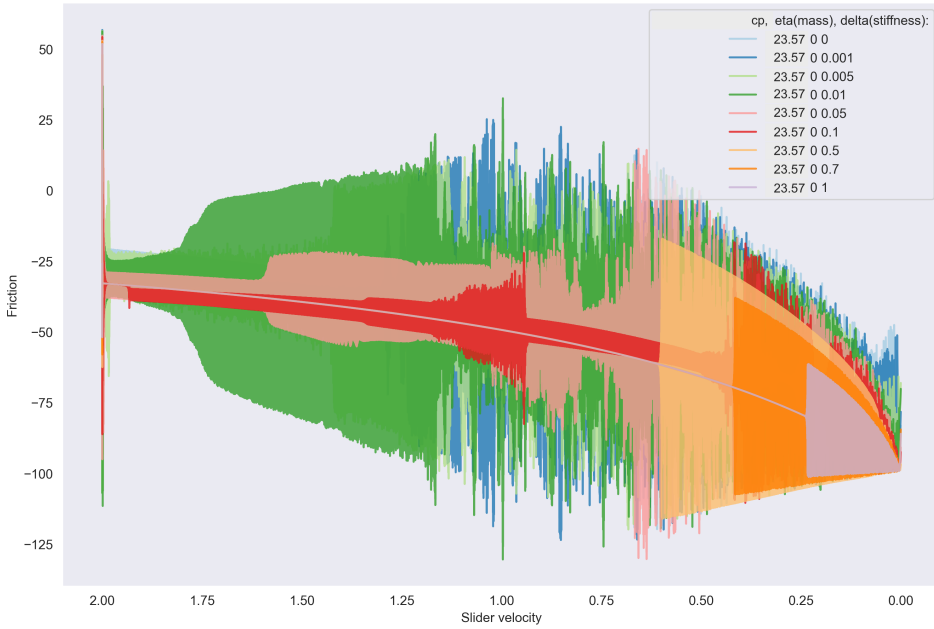
with the Rayleigh curve in Figure 2.7, showing that stiffness proportional damping damps high frequencies, which again generally relates to high velocities. However, when δ is in the range 0.001 to 0.1, adding higher values of stiffness proportional damping increases the friction amplitude in the higher ranges of slider velocity.

To further investigate the latter, Figure 6.27 shows the displacement heat map for $\delta = 0.005$. From $v \in [2, 1.5]$ it shows the dominance of mode 0 and 5 in addition to mode 2 and some other modes. It is clear what modes that are activated, indicating periodic motion. At slider velocity $v \approx 1.4$, there is a change in friction amplitude seen in Figure 6.27(a). This change is also visible in the heat map in Figure 6.27(b), where there is a transition between some modes, especially mode 2 and mode 3. However, it is still clear what modes are most activated. From $v \approx 1.1$, there is another change in both friction amplitude and the heat maps. After this, the heat map starts to look like the very disordered behaviour discussed earlier, where several modes contribute to displacement and there are little consistency in what modes that are activated.

Figure 6.28 shows the friction force plotted for two simulations made to compare damping proportional to k_p and k_{p0}, to k_c. In the light blue plot, the simulation is run with the damping constant proportional to k_p and k_{p0} equal to 0.1 while the damping constant proportional to k_c is equal to zero. Reversely, in the dark blue plot, the damping constant proportional to k_p and k_{p0} is equal to zero, while the damping proportional constant to $k_c = 0.1$. The figure shows how the damping proportional to k_p and k_{p0} contributes to the response of increased friction amplitude in the higher ranges of slider velocity. In other words, it is the damping between the pad and stationary top loader plate and between the pad and the blocks that contribute to this response, while only adding damping between neighboring blocks does not lead to many changes in the response compared to when using only the original c_p damping constant.

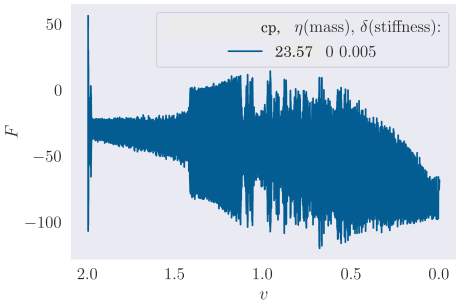


(a)

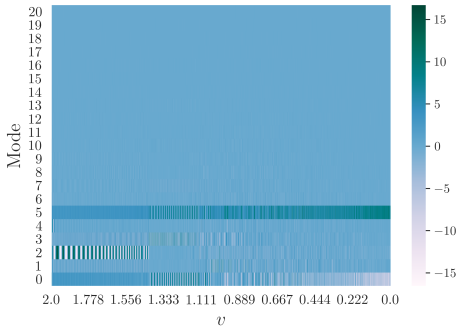


(b)

Figure 6.26: Friction force as a function of slider velocity with different values of the stiffness proportional damping, δ. (Run specifications: A.12.)



(a) Friction as a function of slider velocity with $\delta = 0.005$ and $eta = 0$.



(b) Displacement heat map as a function of velocity.

Figure 6.27: Friction plot and heat map when $\delta = 0.005$ and $\eta = 0$. (Run specifications: Appendix A.13.)

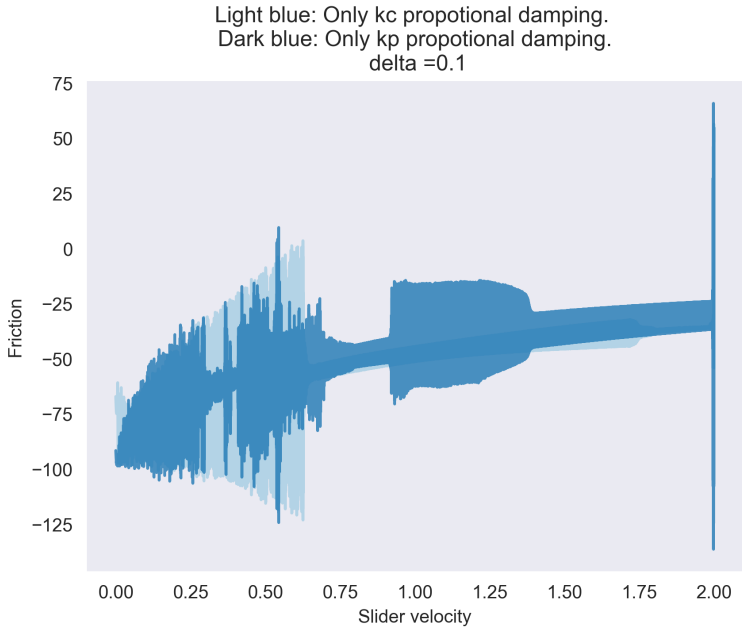


Figure 6.28: Comparison between damping proportional to stationary k_p and pad-top-plate k_{p0} spring constants versus neighbouring k_c spring constant. (Run specifications: A.12.)

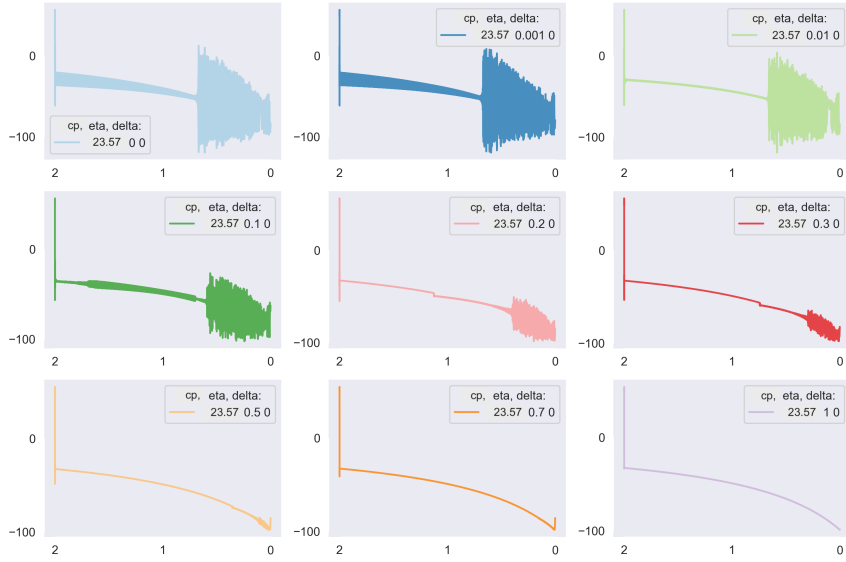
6.6.2 Mass Proportionality

Figure 6.29 shows the friction force plotted against slider velocity for varying values of the mass proportional factor η. The same figure made with a different seed is shows in Appendix I. The transition from the base case in the top left corner of figure 6.29(a), where both the mass and stiffness proportional damping coefficients are equal to 0, to the case where $\eta = 0.01$, shows that increasing η leads to a more damped behaviour. The last row shows that high numbers of η damps the system to a state of almost no fluctuations. When $\eta = 1$, the plot shows a single line. Looking at the Rayleigh curve in Figure 2.7, it is expected that large mass proportional damping damps the behaviour at low frequencies, and thus generally low velocities, to a more considerable degree than the stiffness proportional damping.

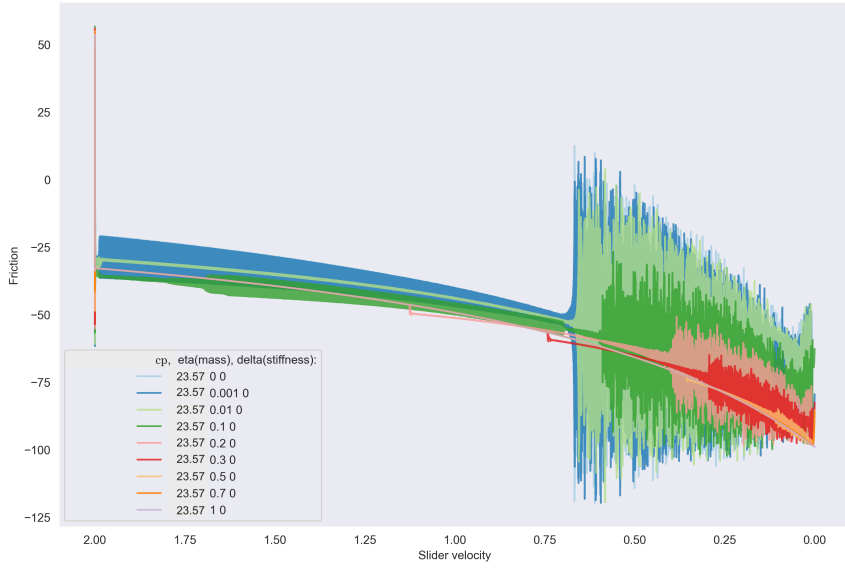
When $\eta = 0.2$, there is a sudden jump in friction force at $v \approx 1$. The same jump is to be found at $v \approx 0.7$ when $\eta = 0.3$. This jump is further investigated in Figure 6.30 and 6.31. In Figure 6.30, η is plotted with different values between 0.15 and 0.4. Figure 6.30 shows that the jump propagates closer to 0 as η is increases. Figure 6.31 gives a closer look at the jump for $\eta = 0.2$ with respective displacement heat maps. Figure 6.31(b), shows that even though there is a jump, the two modes 0 and 5 are still very dominant. When removing these two modes from the plot, as showed in Figure 6.31(c), there is indeed something happening with the modal distribution at the time of the jump. It shows that several modes are activated in a chaotic manner.

6.6.3 Using Significant Modes

Recall Figure 6.8 showing heat maps for seed 101. As mode 5 and mode 14 show a higher amplitude than other modes, they can be chosen as significant modes. Setting ζ to $1/10$, recalling the approach described in Section 5.5 and using Equation 2.27 yields $\eta = 0.20$ and $\delta = 0.047$. Figure 6.32 shows the model response with $\eta = 0.20$ and $\delta = 0.047$, and no other damping, i.e. $c_p = 0$. The figure shows a very damped system until a slider velocity of $v \approx 0.8$, where the response from this point shows similar shape with $\delta = 0.1$ in Figure 6.26.



(a)



(b)

Figure 6.29: Mass proportional damping for different values of η for seed 101. (Run specifications: Appendix A.14.)

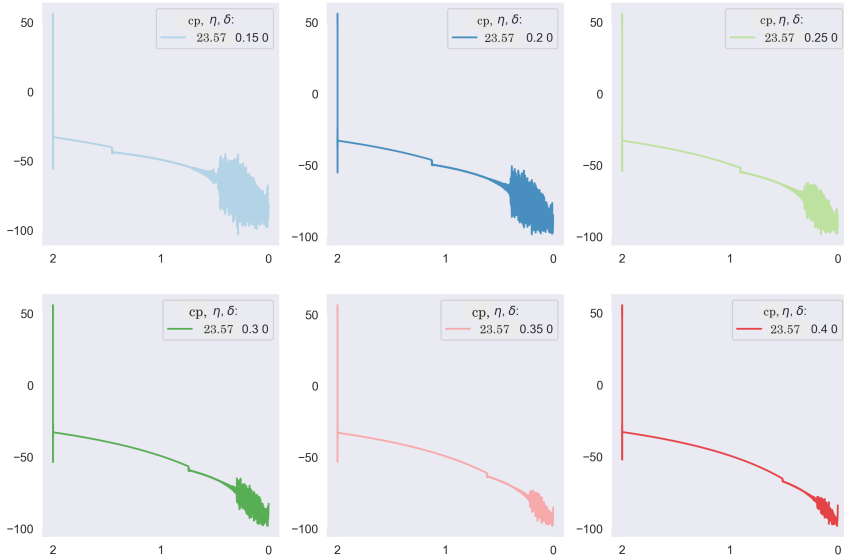
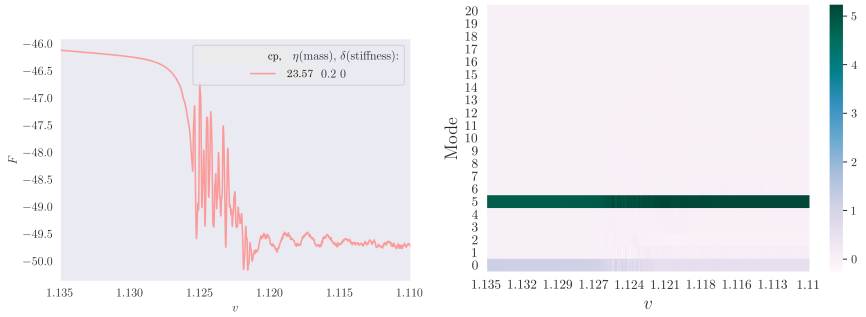
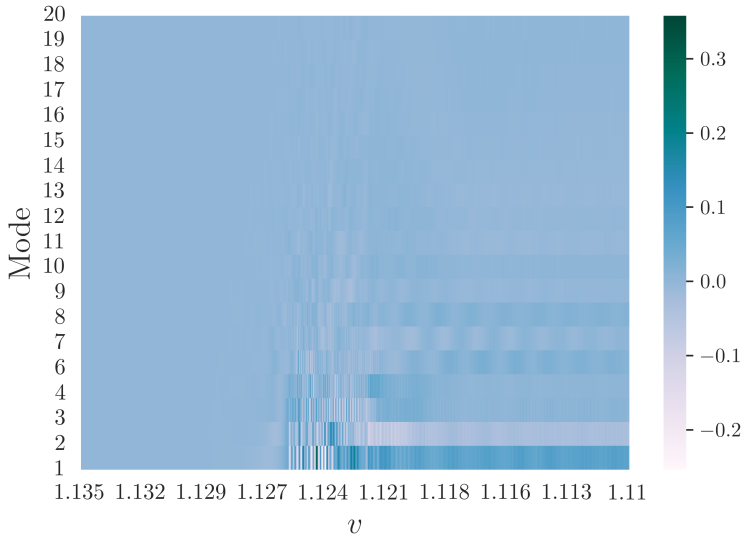


Figure 6.30: Mass proportional damping where η changes from 0.15 to 0.4. (Run specifications: Appendix A.14.)



(a) Friction plot.

(b) Displacement heat map.



(c) Displacement heat map with mode 0 and 5 are removed.

Figure 6.31: Friction plot and heat plot for $\eta = 0.2$ in the velocity range $v = [1.135, 1.110]$. (Run specifications: Appendix A.15.)

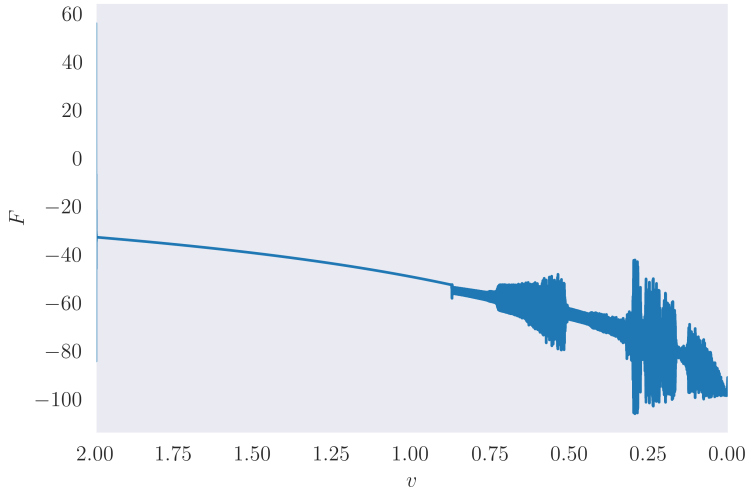


Figure 6.32: Example of Rayleigh damping with $\eta = 0.20$ and $\delta = 0.047$. (Run specifications: Appendix A.16.)

Chapter VII

Discussion

7.1 Mode Study

This thesis has presented a method developed to visualize the contribution of eigenvalues and normal modes to displacement, velocity, and energy at different time steps of the newly developed Burridge-Knopoff-Pad model. The method has proven useful, and the results of the analysis show that the model abruptly changes from oscillating with periodic behaviour to more chaotic motion when changing the external conditions in terms of the slider velocity. The results hold for both increasing and decreasing slider velocity and for different seeds. The sudden change to chaotic motion can be analogous to the onset of squealing sound from car brakes and shows the model's potential.

The Burridge-Knopoff-Pad model shows consistency with research on the area of brake squeal and friction-induced vibrations. As stated in Section 1.4, researchers have indicated, through numerous analytical and experimental studies, that that brake squeal is a non-linear transient phenomenon and that its response can be treated as chaotic. Assuming that the BKP model can represent brake squeal, the results obtained in this thesis further indicate that brake squeal response can be treated as chaotic and thus supports other research. Vice versa, assuming that a good model for representing brake squeal must be capable of exhibiting chaotic motion, the BKP model proves to be a good model to represent brake squeal on this point.

Researchers have previously proposed the hypothesis that systems with a large number of degrees of freedom are usually chaotic (Gallavotti and Cohen (1995)). This is consistent with the BKP model. The study of the model's sensitivity to the number of blocks shows that chaotic behaviour is obtained for as few as ten, and possibly less, blocks. Reducing the number of blocks to as low as 5 or 10 shows chaotic motion, but the model's general behaviour with 10 blocks shows severe differences from the 100-block case. However, increasing the number of blocks in the BKP model shows that once the number of blocks is approximately 30 to 40, additional blocks do not alter the results' general trends. In other

words, the response using 30 blocks closely resembles the response of 100 blocks. This suggests that simulating with 40 blocks could be sufficient for the solution to converge. If the model converges with a low number of blocks, excessive blocks should be avoided as it would reduce computation time.

It is deemed that the results validate that the model transitions to chaotic behaviour, but further work is required to assess the chaotic behaviour quantitatively. A maximum Lyapunov exponent could be calculated to verify the chaotic behaviour with a varying number of blocks and would be a natural place to continue working on the model. The degree of chaos in the numerical results is not quantified, and gathering experimental data from physical experiments and relate it to the model was also outside this thesis's scope. A Poincaré map to study the degree of chaos, bifurcation regimes to study the route to chaos, and validation from physical experiments are thus suggestions for further work.

7.2 BKP versus BK and SDOF model

The BKP model is a combination of the BK and an SDOF model. Through comparisons, the response of the BKP model with 100 blocks resembles the BK model with 100 blocks but looks different from both the SDOF model and the BKP model with one block. The response of the BKP model with one block resembles the SDOF model, often used to study braking systems. The one-block BKP model shows periodic behaviour for both large and small slider velocities. As mentioned, researchers state that brake squeal is non-linear and can be treated as chaotic. Therefore, as neither the SDOF model nor the BKP model with one block detects chaotic motion, they are not capable of representing the sound-induced vibrations of a braking system, and more degrees of freedom are needed.

The study of the number of blocks shows a severe difference between the BK and BKP model. With less than 100 blocks, the BK model shows a much higher friction amplitude for high velocities than that of the BKP model, but the friction amplitude in this region exhibits periodic motion for both models. The extra degree of freedom added to the BK model to create the BKP model gives an indirect coupling between the blocks that the BK model does not have. This makes the pad work as a damper for the model, but above 100 blocks, the two models suddenly resemble each other.

One explanation for this change of behaviour of the BK model at 100 blocks can be the chosen values for the parameters. The block mass m_u is equal to the pad mass, which is set to 100 for all simulations, divided by the number of blocks, making the total block mass independent of the number of blocks. On the contrary, the neighbouring spring constant increases with the number of blocks, $k_c = 1 * N$. That means that when the number of blocks exceeds 100, the neighbouring spring constant becomes larger than the total block mass, making the system stiffness controlled. It also means that when the number of blocks is less than 100, the neighbouring spring constants are smaller than the total block mass,

making the system mass controlled. The latter yields for both the BK and BKP model used in this thesis. From the Rayleigh graph in Figure 2.7, mass controlled systems are characterized with being damped at low frequencies, which are often most present at low slider velocities. It thus makes sense, from a theoretical point of view, that the BK model is not damped at high slider velocities with a block number below 100. When exceeding 100 blocks, the BK model gets stiffness controlled, meaning that the model gets damped at high frequencies, which can be the reason why it is damped at high slider velocities. Comparing this with the BKP model, which only has the same type of undamped motion for high slider velocities when having 1 block, it is evident that the pad works as a damper for the model, and that it damps especially at high slider velocities.

Both in this thesis and the theses by previous students, it has been tried to find differences between the two models when running them with 100 blocks. There are indeed differences and unexplained behaviours regarding, among other things, different types of dips at lower slider velocities and differences in what modes are activated at higher slider velocities. However, the differences do not deal with the general response of the system. Thus, they are considered to be of low significance, as the differences look to be almost equally as much due to numerical differences as the actual differences in the models. As there are few severe differences between the models with 100 blocks, it can be concluded that, if simulating braking systems with 100 blocks, there is little reason to use the BKP instead of the BK model (even though the BK model is made to represent earthquakes). However, as stated in the previous section, there are reasons to represent brake squeal with less than 100 blocks, and in that case, the BKP model can be a better representation for brake squeal than the BK model.

Studying the BK and BKP equations parameters goes beyond this thesis's scope but would be a natural thing to do if continuing to compare the models. Especially to verify why the BK model has a sudden change of its response around 100 blocks, in which behavioural change the BKP model does not have. Also, as the BKP model looks to converge faster to the 100-block case than the BK model, it would be interesting to study if the BKP model would be a better model to study earthquakes than the BK model if using few blocks. With that being said, the BK model is often studied with much more than 100 blocks when studying earthquakes. To study earthquake models goes beyond the area of friction-induced vibrations, but as the work with this model also proves, it could be beneficial for different areas of science to learn from each other.

7.3 Damping

Chapter 6 shows an analysis using Rayleigh damping in the BKP model. In accordance with the Rayleigh curve in Figure 2.7, the stiffness proportional damping matrix affects high frequencies. The opposite is evident for the mass proportional matrix. Thus the study is in some ways consistent with the theoretical framework and shows that the simulations

are working.

However, several unexplained behaviours were seen in this study. First, adding small values for stiffness proportional damping seems to add more chaos to the system. Secondly, some values of the mass proportional damping constant lead to a sudden jump in friction amplitude and a sudden jump of chaotic behaviour. Both these observations are somewhat unexplained, but they verify papers stating the importance of damping when studying the chaotic motion of friction-induced vibration systems, discussed in Section 1.4. The fact that the BKP model is consistent with other friction-induced vibration research regarding damping is another indication to use the model to represent brake squeal.

From the previous sections, it is desired to view the system with fewer blocks. For further work, it would be beneficial to study damping in relation to the number of blocks. That is, to find the amount of damping and a small block number that leads to the case where when increasing the block number further, the results do not change anymore. This is already found to be the case of $30 - 40$ blocks in this study, but with the right amount of damping, it might happen with a much smaller amount of blocks. For this to happen, more stiffness proportional damping affecting low freqencies would probably be needed. To further investigate how different values of the stiffness proportional damping constant affects the system, relate it to the number of blocks, compare it to more previous work, and incorporate the damping in the eigenmode calculations would thus be interesting approaches for further work.

Chapter VIII

Conclusion and Further Work

8.1 Conclusion

The Burridge-Knopoff-Pad model is a spring-mass damper system combining the Burridge-Knopoff model with a single-degree-of-freedom model to capture the squealing sound from car brakes. The model consists of a pad and a number of blocks included as a discretization of the elastic brake discs of a physical braking system, to represent their natural vibrations. This project's goal was to study the newly developed Burridge-Knopoff-Pad model in different ways with a large focus on eigenmodes and damping.

The results show that when changing the system's external conditions in terms of a slider velocity, the motion exhibits a sudden change in behaviour from periodic to chaotic motion. This result is in agreement with research stating that brake squeal can be treated as chaotic motion. Simulations were largely conducted using a total of 101 degrees of freedom (100 blocks); however, a sensitivity study into the impact of the number of blocks showed that the model does not exhibit severe changes when increasing from 30 to 40 blocks, indicating that the number of blocks can be set much lower than 100. On the other hand, as researchers have previously indicated that brake squeal exhibits chaotic motion, the periodicity found in the SDOF system and the one-block BKP model makes these models too simple to describe friction-induced vibrations.

With 100 blocks, the BK and BKP model share the same general behaviour with respect to both friction and position development, and the jumps between chaotic and periodic trends. With fewer blocks a more pronounced difference was found between the two models, where the BKP model noticeably exhibits greater damping for higher ranges of slider velocity as a result of the pad included in the model. From the comparison between the models, it can therefore be concluded that with more than 100 blocks, the pay off of using the BKP model diminishes and is perhaps eclipsed by the extra computational costs of using a more complex model, whereas, for fewer blocks, the results suggest that the BKP model is better suited for representing the sound-induced vibrations of a braking

system.

It has also been demonstrated that damping affects how and where the model enters chaotic regions, and it was found several unexplained behaviours, such as the increase in amplitude when adding different values of Rayleigh damping factors. This also confirms research stating the importance of damping in chaotic friction-induced vibration systems. Hence, the normal mode study results, the study of the number of blocks and Rayleigh damping all show relation to related work and proves the potential of the model, indicating that further work is needed.

8.2 Proposal for Further Work

- Confirm chaotic behaviour by calculating the maximum Lyapunov exponent.
- Analysis of chaos, bifurcation, and the Poincaré section as described in Artuso et al. (2020) and done by e.g. Chang and Hu (2016).
- Do a more dedicated study on Rayleigh damping, especially stiffness proportional damping, and find the exact number of blocks and amount of damping needed for the solution to converge.
- Perform a more substantial parametric study to further outline the difference between the Burridge-Knopoff and Burridge-Knopoff-Pad model and to get more familiar with the latter.
- Explore the application of the Burridge-Knopoff-Pad model in earthquake modeling.
- The stick percentage was studied for two different seeds. It would be interesting to investigate why there is a drop in stick percentage around the constant slider velocity 0.004 and 0.005.
- Test the model using a numerical schemes other than 2^{nd} order Runge-Kutta, with an emphasis on stability, accuracy, and computational cost.

References

Artuso, P.C.R., Mainieri, R., Tanne, G. and Vattay, G., 2020. *Chaos: Classical and quantum*. 17th ed. Niels Bohr Institute, Copenhagen. Available from: www.chaosbook.org.

Asadi, A., 2016. *The python book*. 2nd ed. Imagine Publishing. ISBN: 9781785462382.

Ben-Kiki, O., Evans, C. and Net, I. döt, 2001-2009. Yaml ain't markup language (yaml™) version 1.2 [Online]. Accessed: 31.01.2021. Available from: https://yaml.org/spec/.

Bergan, P.G., Larsen, P.K. and Mollestad, E., 1986. *Svingninger av konstruksjoner*. 2nd ed. Tapir.

Brigs, W., 2019. *C++ for lazy programmers: Quick, easy, and fun C++ for beginners* [Online]. Apress, Berkeley, CA. Available from: https://doi.org/10.1007/978-1-4842-5187-4.

Burridge, R. and Knopoff, L., 1967. Model and theoretical seismicity. *Bulletin of the seismological society of america* [Online], 57. Available from: https://doi.org/10.1093/gji/11.1.265.

Carlson, J.M. and Langer, J.S., 1989. Mechanical model of an earthquake fault. *Institute for theoretical physics, university of california* [Online]. Available from: https://doi.org/10.1103/PhysRevA.40.6470.

Chang, S.C. and Hu, J.F., 2016. Nonlinear dynamics and control in an automotive brake system. *Department of mechanical and automation engineering, , da-yeh university, changhua* [Online], 5. Available from: https://doi.org/10.4172/2167-7670.1000135.

Chopra, A.K., 2017. *Dynamics of structures - theory and applications to earthquake engineering*. 5th ed. Pearson.

Chowdhury, I. and Dasgupta, S.P., 2003. Computation of rayleigh damping coefficients for large systems. *Electronic journal of geotechnical engineering*, 43, pp.6855–6868.

Clough, R. and Penzien, J., 1995. *Dynamics of structures*. 3rd ed. Computers & Structures, Inc.

Corrado, P.M. and Mascia, M.C., 2016. Assessment of predictor-corrector strategy for the burridge-knopoff model. *Cornell university* [Online]. Available from: arXiv:1607.08267.

Craig, R.R. and Kurdila, J.A.J., 1981. *Fundamentals of structural dynamics*. 2nd ed. Wiley.

Erickson, B., Birnir, B. and Lavallée, D., 2010. Periodicity, chaos and localization in a burridge-knopoff model of an earthquake with dieterich-ruina friction. *Uc santa barbara: Center for complex and nonlinear science* [Online]. Available from: https://escholarship.org/uc/item/3r5811tp.

Ferguson, C.D., Klein, W. and Rundle, J.B., 1998. Long-range earthquake fault. *Computers in physics* [Online], 12, p.34. Available from: https://doi.org/10.1063/1.168681.

Ferre, H.N., 2018. The burridge-knopoff-pad model - a new model for studying noise generation & brake dynamics. *NTNU open* [Online]. Available from: https://doi.org/11250/2498106.

Freedman, Y., 1998. *Sears and Zemansky's University Physics with modern physics*. 10th ed. Pearson.

Gallavotti, G. and Cohen, E.G.D., 1995. Dynamical ensembles in stationary states. *Journal of statistical physics* [Online], 80. Available from: https://doi.org/10.1007/BF02179860.

Guennebaud, G., Jacob, B. et al., 2010. Eigen v3. http://eigen.tuxfamily.org. Accessed: 20/01/2021.

Géradin, M. and Rixen, D., 1997. *Mechanical Vibrations - Theory and Application to Structural Dynamics*. 2nd ed. Wiley.

Hellevik, L.R., 2020. Numerical methods for engineers [Online]. Accessed = 15/10/2020. Available from: http://folk.ntnu.no/leifh/teaching/tkt4140/.

Hunter, J.D., 2007. Matplotlib: A 2d graphics environment. *Computing in science & engineering* [Online], 9(3), pp.90–95. Available from: https://doi.org/10.1109/MCSE.2007.55.

Høgberg, R., 2019. Simulating fretting fatigue using the burridge-knopoff-pad model. *Ntnu open* [Online]. Available from: https://doi.org/11250/2658913.

Jean-Jacques Sinou, L.J., 2007. Mode coupling instability in friction-induced vibrations and its dependency on system parameters including damping. *European journal of mechanics a/solids* [Online], 26, pp.106–122. Available from: https://doi.org/10.1016/j.euromechsol.2006.03.002.

Kreyszig, E., 2011. *Advanced engineering mathematics*. 10th ed. Wiley.

Lay, D.C., Lay, S.R. and McDonald, J.J., 2016. *Linear Algebra and its applications*. 5th ed. Pearson.

Mascia, C. and Moschetta, P., 2020. Numerical evidences of almost convergence of wave speeds for the burridge–knopoff mode. *Sn appl. sci.* [Online]. Available from: https://doi.org/10.1007/s42452-020-03856-y.

Michael Waskom and the Seaborn development team, 2020. *mwaskom/seaborn* (v.latest). Zenodo. Accessed: 11/01/2021. Available from: https://doi.org/10.5281/zenodo.592845.

Morin, D., 2007. Introduction to classical mechanics with problems and solutions. *Cambridge university press*.

Papangelo, A., Grolet, A., Salles, L., Hoffmann, N. and Ciavarella, M., 2016. Snaking bifurcations in a self-excited oscillator chain with cyclic symmetry. *Communications in nonlinear science and numerical simulation* [Online], 44, pp.108–119. Available from: https://doi.org/10.1016/j.cnsns.2016.08.004.

Persson, B.N., 1998. *Sliding friction: physical principles and applications*. Springer.

Python Software Foundation, 2021. Csv file reading and writing [Online]. Accessed: 31.01.2021. Available from: https://docs.python.org/3/library/csv.html.

Sanderson, C. and Curtin, R., 2016. Armadillo: a template-based C++ library for linear algebra [Online]. Accessed: 20/09/2020-05/02/2021. Available from: http://arma.sourceforge.net/docs.html.

Standnes, K., 2019. Numerical simulation comparing the burridge-knopoff and burridgeknopoff-pad model. *Ntnu open* [Online]. Available from: https://doi.org/11250/2621461.

Stroustrup, B., 2014. *Programming: Principles and practice using C++*. 2nd ed. Addison-Wesley Professional.

VS Code, n.d. Vs code documentation [Online]. Accessed: 03.02.2021. Available from: https://code.visualstudio.com/docs.

Wernitza, B.A. and Hoffmannb, N.P., 2016. Recurrence analysis and phase space reconstruction of irregular vibration in friction brakes: Signatures of chaos in steady sliding. *Journal of sound and vibration* [Online], 331, pp.3887–3896. Available from: https://doi.org/10.1016/j.jsv.2012.04.003.

Wilson, E.L., 2002. *Three-dimensional static and dynamic analysis of structures*. 3rd ed. CSI.

Appendix - A

Run Specifications

Unless other is specified, all specifications will be as in Appendix A.1

A.1 Main Run

```
Parameters:
  dt: 0.005
  seed: 101
  num_events: 1
  N: 100
  max_time: 80000
  slider_speed: 0
  increment: 0.1
  interval: 0
  file_name: "simulation_1"
  progress_indicator: true
  m_F0: 1
  m_alpha: 0.5
  m_sigma: 0.01
  m_mass_x: 100
  m_scale_mass: 1
  m_zeta: 0.0833
  m_k_P0: 100
  m_scale_P: 1
  m_scale_C: 0.01
  m_t: 0.0
  m_v0: 0.0001
  m_u_min: 0
  blocks:
  start_speed_continuous: 2
  end_speed_continuous: 0
  save_interval_dt: 100
  threshold_speed: 0.1
  m_eta: 0.2
  m_delta: 0.047
  lyapunov_delta: 0.00000001
```

```
Debug:
  debug_no_friction: false
  debug_no_neighbor_springs: false
  debug_no_stationary_springs: false
  debug_no_damper: false
  debug_no_pad: false
  debug_negative_initial_values: false
  debug_only_negative_initial: false
  debug_only_write_friction: false
  debug_continuous_slider_speed: true
  debug_one_degree_freedom_mode: false
  rayleigh_damping: false
  calculate_Lyapunov: false
```

A.2 Small Save Interval

```
Parameters:
  seed: 101
  max_time: 80000
  save_interval_dt: 10
  start_speed_continuous: 2
  end_speed_continuous: 0
```

A.3 Start Speed 4

```
Parameters:
  seed: 101
  max_time: 160000
  start_speed_continuous: 4
  end_speed_continuous: 0
```

A.4 Seed 100

```
Parameters:
  seed: 100
  max_time: 80000
  start_speed_continuous: 2
  end_speed_continuous: 0
```

A.5 Start Speed 4, Seed 100

```
Parameters:
  seed: 100
  max_time: 160000
  start_speed_continuous: 4
  end_speed_continuous: 0
```

A.6 Seed 104

```
Parameters:
  seed: 104
  max_time: 80000
  start_speed_continuous: 2
  end_speed_continuous: 0
  save_interval_dt: 100
```

A.7 Increasing Run

```
Parameters:
  seed: 101
  max_time: 80000
  start_speed_continuous: 0
  end_speed_continuous: 2
  save_interval_dt: 100
```

A.8 Small Run

```
Parameters:
  seed: 101
  N: 100
  max_time: 800
  start_speed_continuous: 2
  end_speed_continuous: 0
  save_interval_dt: 100
```

A.9 Constant Slider Velocity

```
Parameters:
  seed: 101
  max_time: 80000
  start_speed_continuous: 0.001
  end_speed_continuous: 0.001
  save_interval_dt: 100
```

A.10 BK Decreasing Run

```
Parameters:
  seed: 101
  max_time: 80000
  start_speed_continuous: 2
  end_speed_continuous: 0
  save_interval_dt: 100
Debug:
  debug_no_pad: True
```

A.11 BK Increasing Run

```
Parameters:
  seed: 101
  max_time: 80000
  start_speed_continuous: 0
  end_speed_continuous: 2
  save_interval_dt: 100
 Debug:
  debug_no_pad: True
```

A.12 Stiffness Proportional Damping

```
Parameters:
  seed: 101
  max_time: 80000
  start_speed_continuous: 2
  end_speed_continuous: 0
  save_interval_dt: 100
  r_eta: 0
  r_delta: varying
 Debug:
  rayleigh_damping: true
```

A.13 Stiffness Proportional Damping

```
Parameters:
  seed: 101
  max_time: 80000
  start_speed_continuous: 2
  end_speed_continuous: 0
  save_interval_dt: 100
  r_eta: 0
  r_delta: 0.005
Debug:
  debug_no_damper = false
  rayleigh_damping: true
```

A.14 Mass Proportional Damping

```
Parameters:
  seed: 101
  max_time: 80000
  start_speed_continuous: 2
  end_speed_continuous: 0
  save_interval_dt: 100
  r_eta: varying
  r_delta: 0
```

```
 Debug:
  debug_no_damper = false
  rayleigh_damping: true
```

A.15 Mass Proportional Damping

```
Parameters:
  seed: 101
  max_time: 80000
  start_speed_continuous: 2
  end_speed_continuous: 0
  save_interval_dt: 100
  r_eta: 0.2
  r_delta: 0
 Debug:
  debug_no_damper = false
  rayleigh_damping: true
```

A.16 Rayleigh Damping

```
Parameters:
  seed: 101
  max_time: 80000
  start_speed_continuous: 2
  end_speed_continuous: 0
  save_interval_dt: 100
  r_eta: 0.20
  r_delta: 0.047
 Debug:
  debug_no_damper = true
  rayleigh_damping: true
```

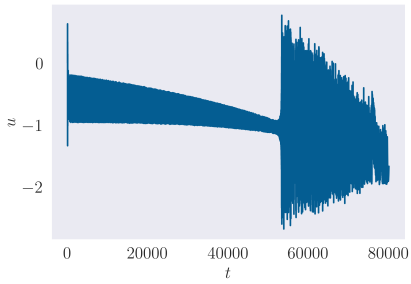
Appendix - B

Web Page

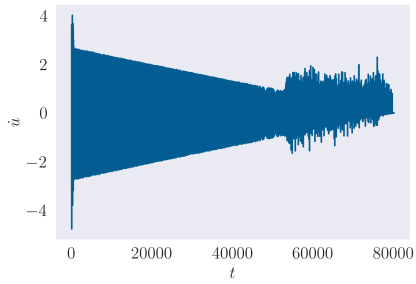
A web page was created as an extended appendices. The service *GitHub Pages* was used in order to make the website, which is a free way to create simple web pages. The web page is made using purely markdown (.md) and HTML programming and the site generator jeykill. Jeykill makes the theme for the page and makes it responsive (working on a standard computer screen, iPad, and on galaxy and iPhone). By default, when creating a repository for a GitHub Pages page, an index.md file is created which will be the default main page of the web page. Markeown is a simple language to make text documents, and is the standard languge used to create readme-files.

Appendix - C

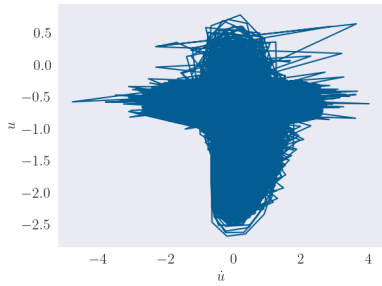
General Behaviour



(a) Block position as a function of time.



(b) Block velocity as a function of time.



(c) Block position as a function of block velocity.

Figure C.1: General behaviour of block number 50. (Run specification: Appendix A.1.)

Appendix - D

Stick Percentage for Seed 1

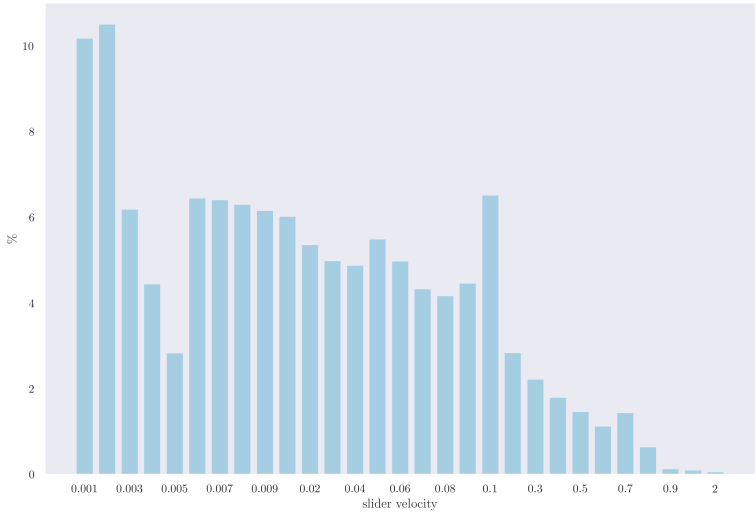


Figure D.1: Mean stick percentage for each block for different constant slider velocities for seed 1.

Appendix - E

Energy Heat Map

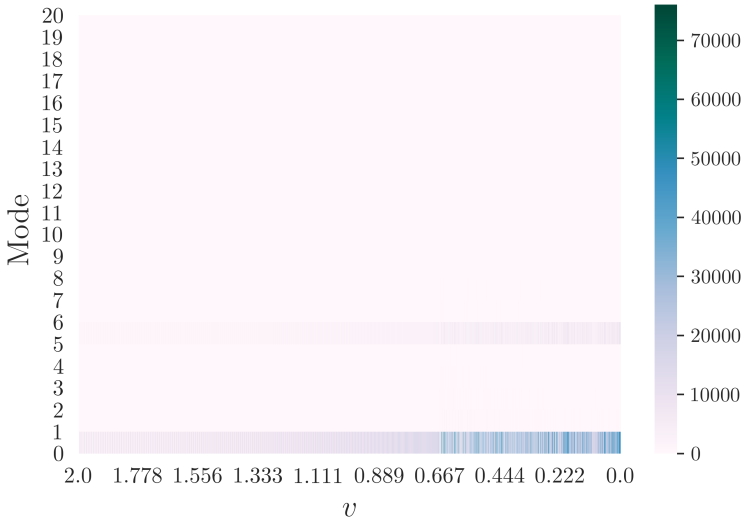


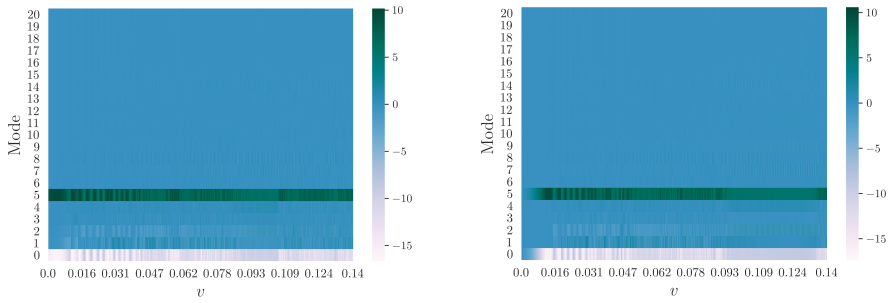
Figure E.1: Energy heat map. Run specification (Appendix A.1.)

Appendix - F

Increasing Slider Velocity

Standnes (2019) studied the difference between the response of increasing and decreasing slider velocity. He showed four slider velocity regions where the two runs differ. Modal displacement heat maps for the same four slider velocity sections are shown in Figure F.1 to F.4. Figure F.5 and F.6 shows zoomed heat maps of the slider velocity regions discussed in Section 6.3.

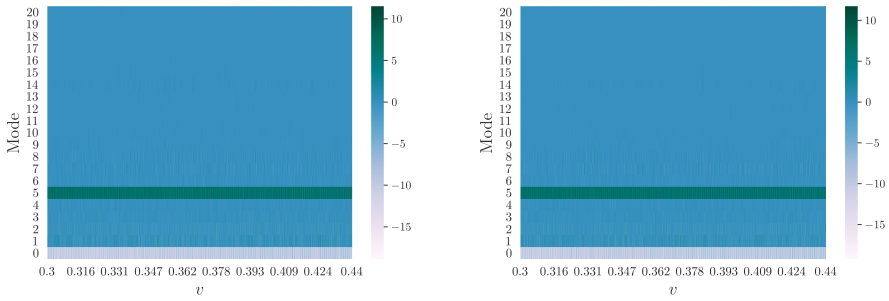
Note that the heat maps for the decreasing case are horizontally flipped so they start at the lowest slider velocity. For the run with slider velocity ranging from 2 to 0, this means that slider velocity 0 is on the left hand side of the figure even though the slider velocity of the simulation starts at 2. This is done for a more manageable comparison between the cases.



(a) Decreasing.

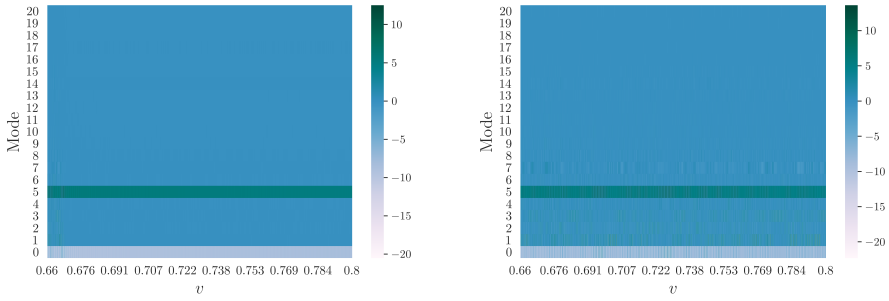
(b) Increasing.

Figure F.1: Heatmaps for increasing and decreasing runs, in the slider velocity range 0 to 0.14.



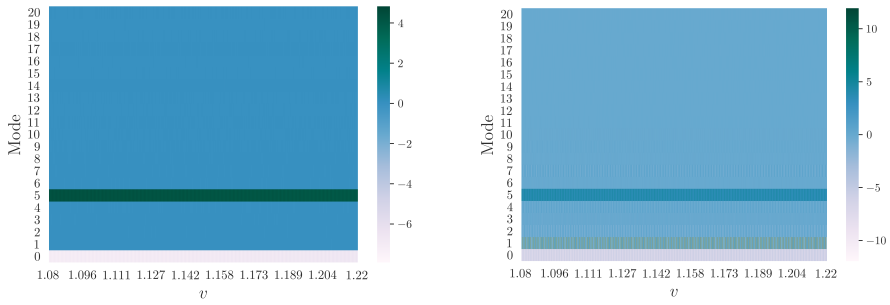
(a) Decreasing. **(b)** Increasing.

Figure F.2: Heatmaps for increasing and decreasing runs, in the slider velocity range 0.3 to 0.44.



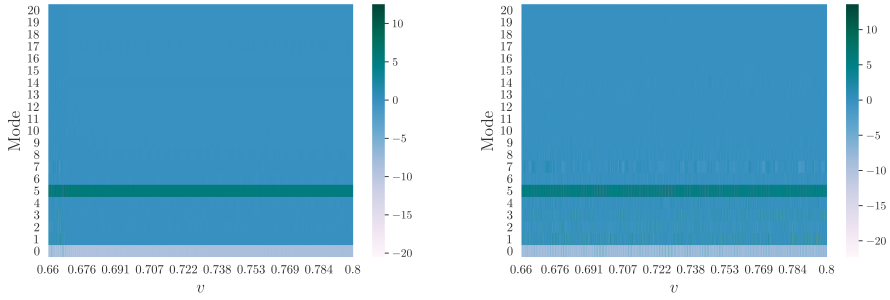
(a) Decreasing. **(b)** Increasing.

Figure F.3: Heat maps for increasing and decreasing runs, in the slider velocity range 0.8 to 0.66.



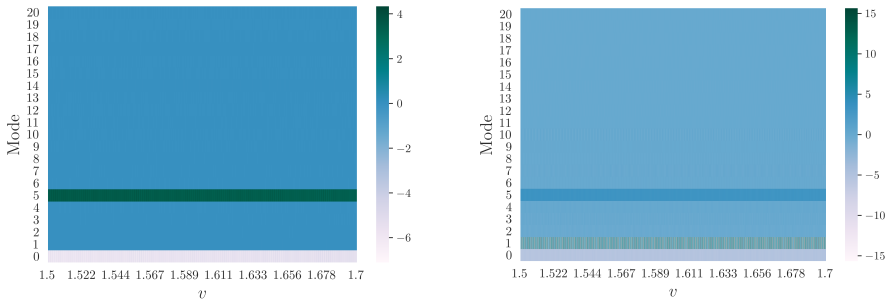
(a) Decreasing. **(b)** Increasing.

Figure F.4: Heatmaps for increasing and decreasing runs, in the slider velocity range 1.08 to 1.22.



(a) Decreasing. **(b)** Increasing.

Figure F.5: Heat maps for increasing and decreasing runs, in the slider velocity range 0.8 to 0.66. (Run specifications: Appendix A.1 and A.7).



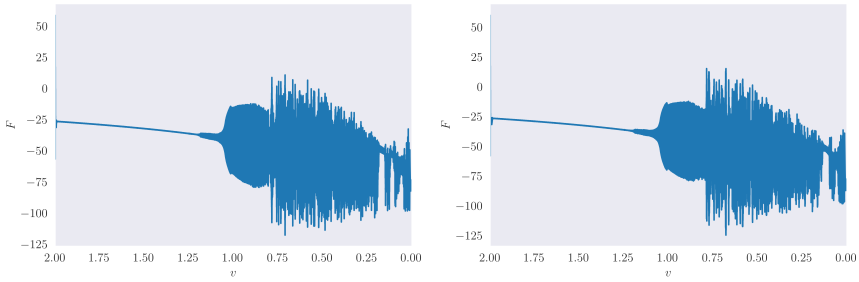
(a) Decreasing.

(b) Increasing.

Figure F.6: Heat maps for increasing and decreasing runs, in the slider velocity range 1.5 to 1.7. (Run specifications: Appendix A.1 and A.7.)

Appendix - G

130 Blocks



(a) Seed 30.

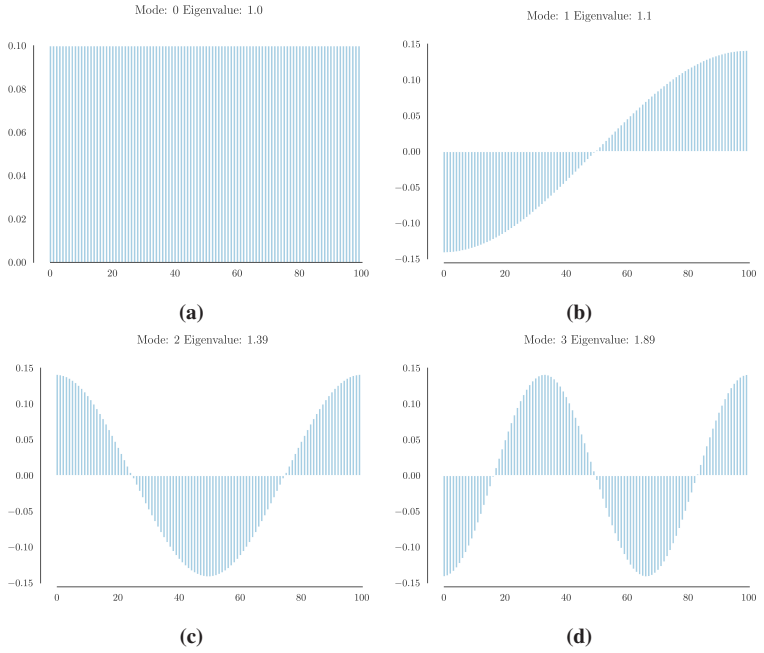
(b) Seed 140.

Figure G.1: Simulations ran with 130 blocks.

Appendix - H

The BK Model

H.1 Mode Shapes of the BK Model



(a) **(b)**

(c) **(d)**

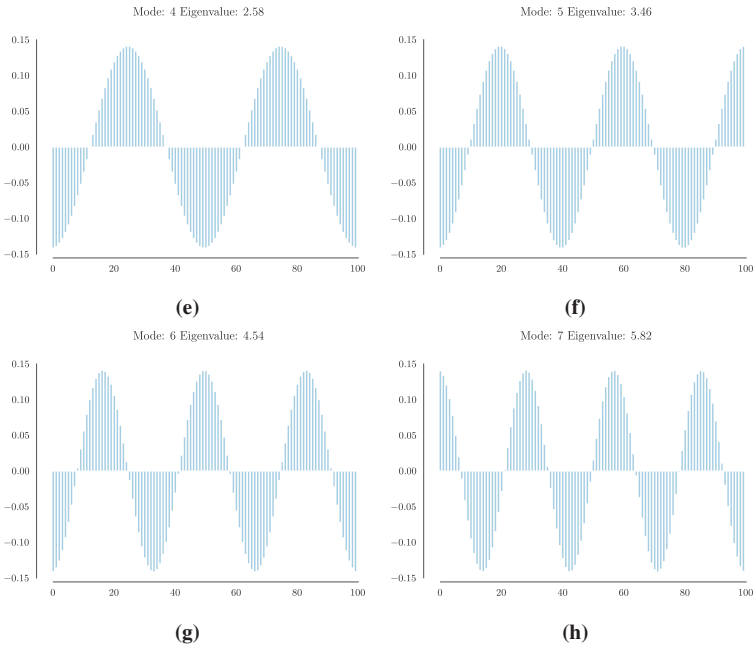
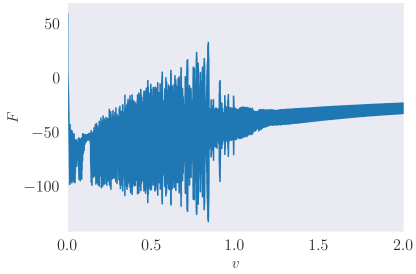
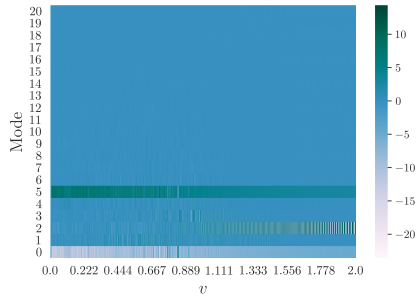


Figure H.1: Bar plot of the first 8 eigenmodes for the BK model, where each value in the eigenvector corresponding to each mode is plotted as a bar.

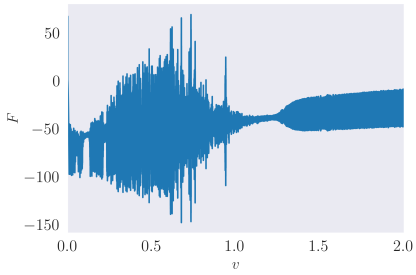
H.2 Comparing BK and BKP for Seed 100



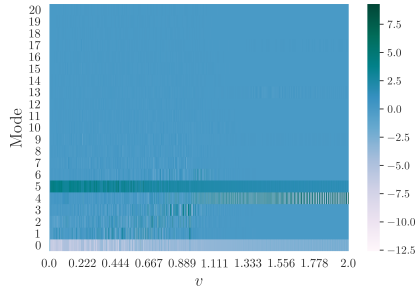
(a) BKP.



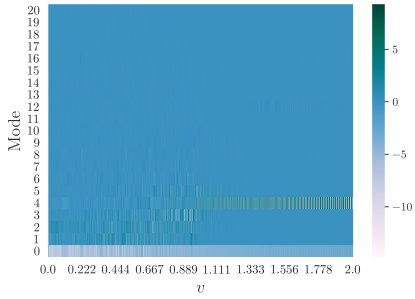
(b) BKP.



(c) BK.

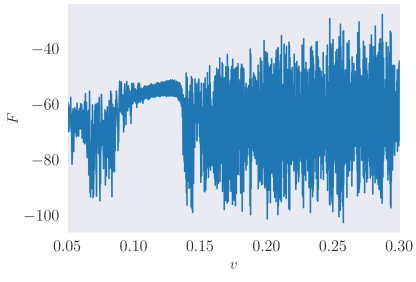


(d) BK with mode shapes from BKP.

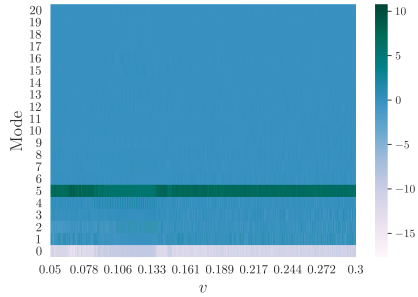


(e) BK with mode shapes from BK.

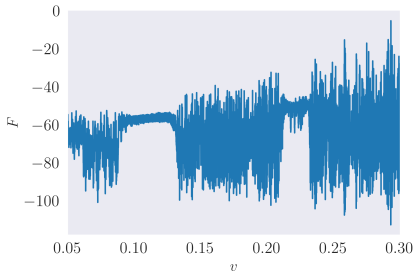
Figure H.2: Increasing slider velocity from 0 to 2.



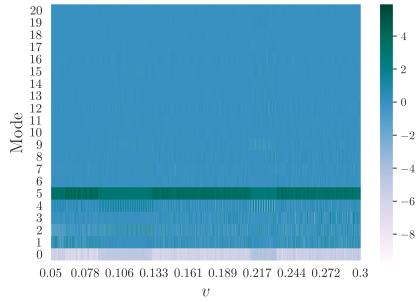
(a) BKP.



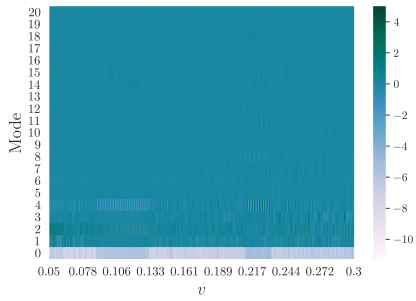
(b) BKP.



(c) BK.

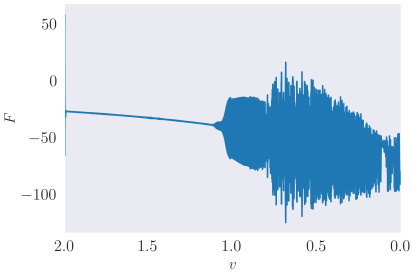


(d) BK with mode shapes from BKP.

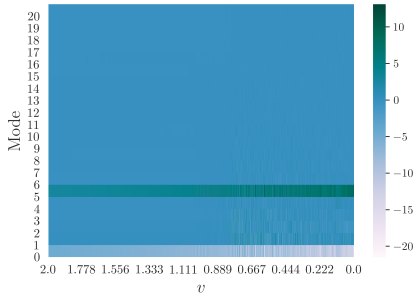


(e) BK with mode shapes from BK.

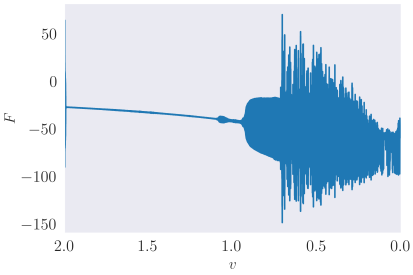
Figure H.3: Increasing slider velocity from 0.05 to 0.3.



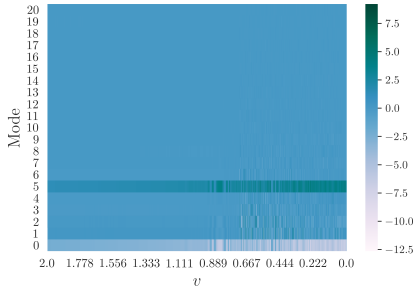
(a) BKP.



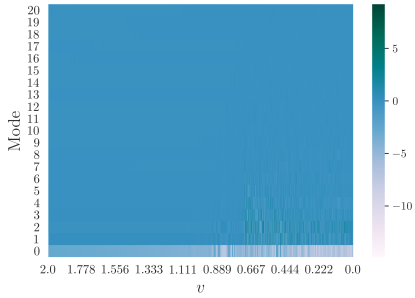
(b) BKP.



(c) BK.

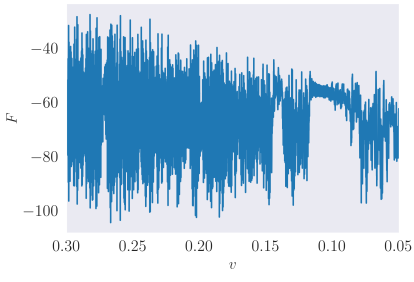


(d) BK with mode shapes from BKP.

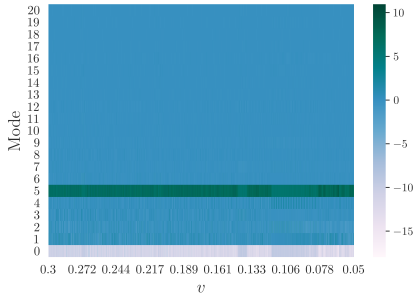


(e) BK with mode shapes from BK.

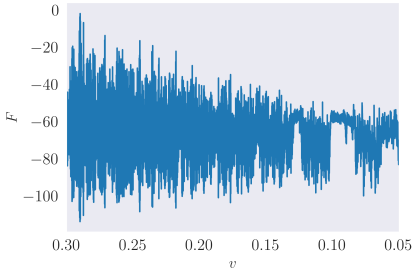
Figure H.4: Decreasing slider velocity from 2 to 0.



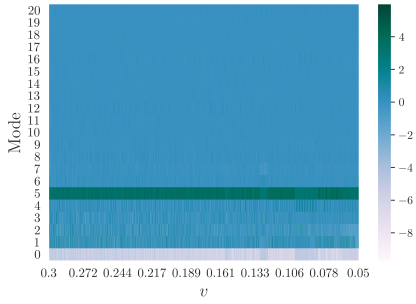
(a) BKP.



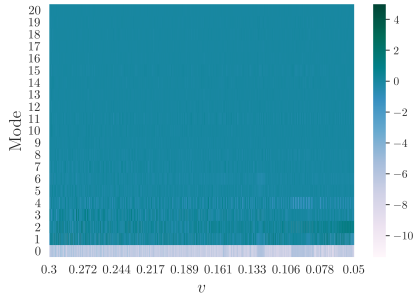
(b) BKP.



(c) BK.



(d) BK with mode shapes from BKP.



(e) BK with mode shapes from BK.

Figure H.5: Decreasing slider velocity from 0.05 to 0.3.

Appendix - I

Rayleigh Damping

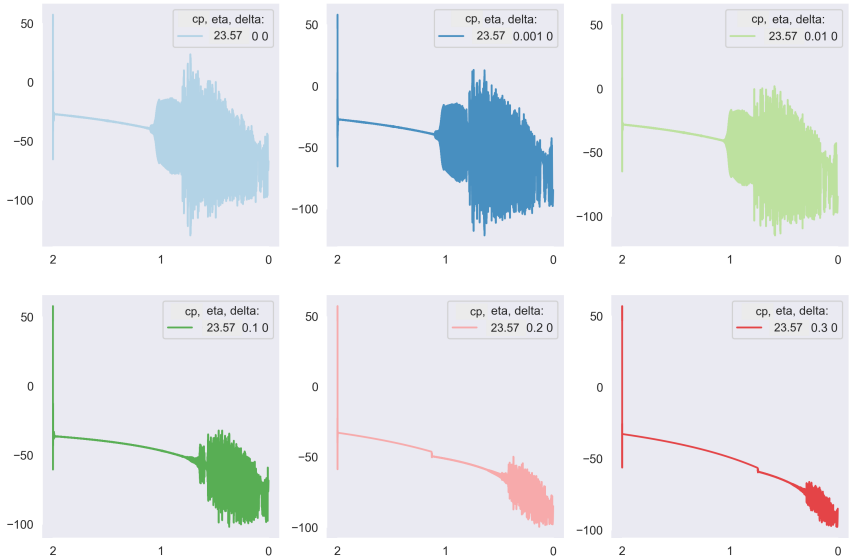


Figure I.1: Different values of the mass proportional damping constant η for seed 100.

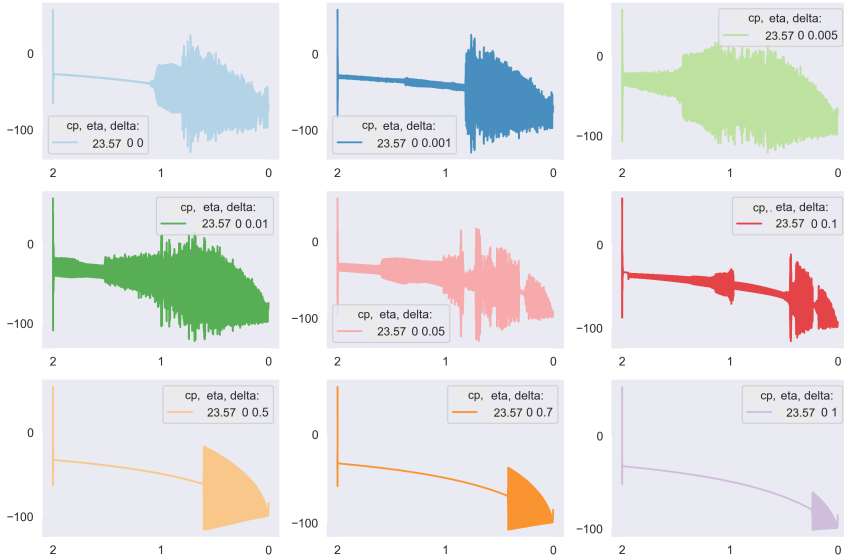


Figure I.2: Stiffness proportional damping for different values of δ for seed 100.

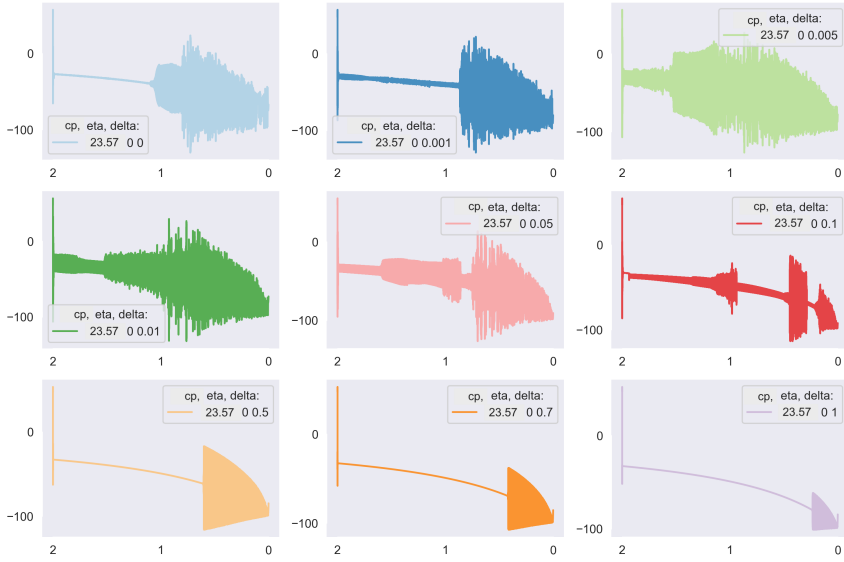


Figure I.3: Stiffness proportional damping for different values of δ for seed 103.

Thea Martine Marstrander

Investigation of the Burridge-Knopoff-Pad Model

NTNU
Norwegian University of
Science and Technology