



Norwegian University of  
Science and Technology

# Active and passive damping systems for vibration control of metal machining equipment

Ole Christian Hermanrud

Master of Science in Mechanical Engineering

Submission date: May 2017

Supervisor: Bjørn Haugen, MTP

Co-supervisor: Olav Egeland, IPK  
Dan Östling, Sandvik Teeness AS

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



NTNU - NORGES TEKNISK-  
NATURVITENSKAPELIGE UNIVERSITET  
INSTITUTT FOR PRODUKTUTVIKLING  
OG MATERIALER

**MASTEROPPGAVE VÅR 2016**  
**FOR**  
**STUD.TECHN. OLE CHRISTIAN HERMANRUD**

**System for aktiv og passiv viberasjonsdemping av metallbearbeidingsverktøy**

***Active and passive damping systems for vibration control of metal machining equipment.***

Ved metallbearbeiding er maskineringsutstyret ofte utsatt for vibrasjoner og resonans. Dette er ofte holdt innenfor akseptable toleranse ved hjelp av passive dampingssystemer. I samarbeid med Sandvik Teeness så ønsker en å utforske egenskapene til noen aktive dempingsmetoder og systemer.

En sentral del av oppgaven vil være å bygge fysiske demonstratorer for ett eller flere aktive dampingssystem og teste disse ved bruk av ristebord hos Sandvik Teeness. De fysiske testresultatene skal sammenlignes med relevante numeriske modeller.

Vurdering av egnethet for forskjellige aktuatorer og sensorer vil være en del av oppgaven. En ønsker også å vurdere om Arduino er egnet som prosessor for reguleringssystemet for de aktuelle frekvensområdene.

**Formelle krav:**

Senest 3 uker etter oppgavestart skal et A3 ark som illustrerer arbeidet leveres inn. En mal for dette arket finnes på instituttets hjemmeside under menyen masteroppgave (<https://www.ntnu.no/web/ipm/masteroppgave-ved-ipm>). Arket skal også oppdateres en uke før innlevering av masteroppgaven.

Risikovurdering av forsøksvirksomhet skal alltid gjennomføres. Eksperimentelt arbeid definert i problemstilling skal planlegges og risikovurderes innen 3 uker etter utlevering av oppgavetekst. Konkrete forsøksvirksomhet som ikke omfattes av generell risikovurdering skal spesielt vurderes før eksperimentelt arbeid utføres. Risikovurderinger skal signeres av veileder og kopier skal inngå som vedlegg til oppgaven.

Besvarelsen skal ha med signert oppgavetekst, og redigeres mest mulig som en forskningsrapport med et sammendrag på norsk og engelsk, konklusjon, litteraturliste, innholdsfortegnelse, etc. Ved utarbeidelse av teksten skal kandidaten legge vekt på å gjøre teksten oversiktlig og velskrevet. Med henblikk på lesning av besvarelsen er det viktig at de nødvendige henvisninger til korresponderende steder i tekster, tabeller og figurer anføres på begge steder. Ved bedømmelse legges det stor vekt på at resultater er grundig bearbeidet, at de oppstilles tabellarisk og/eller grafisk på en oversiktlig måte og diskuteres utførlig.

Besvarelsen skal leveres i elektronisk format via DAIM, NTNUs system for Digital arkivering og innlevering av masteroppgaver.

Kontaktpersoner:

Dr. Ing. Dan Östling, Sandvik Treeness AS  
Professor Olav Egeland, IPK



Torgeir Welo  
Instituttleder



Bjørn Haugen  
Faglærer



NTNU  
Norges teknisk-  
naturvitenskapelige universitet  
Institutt for produktutvikling  
og materialer

## *Acknowledgements*

I would like to express my gratitude to my supervisor Bjørn Haugen for his supervision and help during this thesis. His contributions have been much needed and greatly valued. He made sure that the paper would be my own work, but steered me in the right direction whenever he thought I needed it. He was also very understanding during the difficult times while working with the thesis.

I would also like to thank Dan Østling for his input and ideas. He has played an important role in enabling me to finish the thesis. All the employees at Teeness have been very helpful and interested in finding the best solution.

Finally, I must express my very profound gratitude to my parents and to my spouse for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Ole Christian Hermanrud

# **Active and passive damping systems for vibration control of metal machining equipment**

**Ole Christian Hermanrud**

## **Abstract**

Passive damping is used to reduce vibrations of machining tools. The drawback with the passive damping system is that it is adjusted to damp out vibrations in one restricted frequency range. Active damping uses real time measurements to dampen the vibrations and is not tuned for one specific frequency range. Teeness AS wants to be able to investigate different ways of regulating an active damping system on a physical test bench and to investigate if an Arduino is suited for regulating the system.

This paper presents the design development of the physical test bench, simulations of the designed test bench and an electrical setup with Arduino as the regulating controller.

Actuators were evaluated and purchased. The test bench was designed to fit the actuators using Siemens NX as CAD program. The components of the test bench were produced by Teeness AS. Simulations of the designed test bench were conducted using FEDEM to investigate the dynamics of the system. Requirements for the Arduino were investigated and regulating scripts were coded, tested and evaluated.

The simulated test bench was able to significantly reduce the vibrations investigated. However there were some behaviours of the simulated test bench that differ from the measurements made of the physical test bench. This was most likely a result of the modeling of actuators. The results from the simulations give an indication of the general behaviour of the test bench, but the use of exact values should be avoided. To improve the validity of the simulations, further work with the simulation of the actuators should be conducted. Using an Arduino as a control mechanism seems promising, but further work needs to be done to reduce the noise of the sensor data and the regulating script presented in this paper should be improved.

# **Active and passive damping systems for vibration control of metal machining equipment**

**Ole Christian Hermanrud**

## **Sammendrag**

Passiv demping er brukt for å redusere vibrasjoner på metal bearbeidende verktoy. Ulempen med passive dempesystem er at systemet er justert for å dempe ut vibrasjoner rundt en gitt frekvens. Aktiv demping på den andre siden bruker nåtids data for å regulere et system som demper ut vibrasjoner og er ikke avhengig av å bli justert for en bestemt frekvens. Teeness AS ønsker å undersøke forskjellige måter å regulere et aktivt dempesystem på en fysisk testbenk og å undersøke om det er mulig å bruke Arduino som reguleringsmekanisme.

Denne oppgaven presenterer utviklingen av den fysiske testbenken, simuleringer av den modellerte testbenken og et elektrisk oppsett der en Arduino er brukt for å regulere systemet.

Aktuatorer ble evaluert og anskaffet. Deretter ble testbenken designet rundt aktuatorene ved å bruke Siemens NX som modelleringsverktøy. Delene modellert ble produsert av Teeness AS. Simuleringer av den modellerte testbenken ble gjennomført i FEDEM for å undersøke de dynamiske egenskapene til testbenken. Krav til Arduino ble undersøkt og skript ble laget og evaluert.

Den simulerte testbenken kunne redusere de undersøkte vibrasjonene signifikan. Det var noe avvik fra simuleringer og resultater fra testing av den fysiske testbenken. Avviket oppstår mest sannsynlig grunnet modellering av aktuator ikke er slik som i virkeligheten. Eksakte verdier fra simuleringer blir da ikke vektlagts, men fokus er på den generelle oppførselen. Videre arbeid med modellering av aktuatorer er anbefalt for få mere representative simuleringer. Å bruke Arduino som kontroll mekanisme viker lovende. Videre arbeid med å redusere støy og forbedring av script må gjennomføres for å gi en sikker konklusjon på om Arduino er egnet eller ikke.

# Contents

<b>Acknowledgements</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>Sammendrag</b>	<b>iii</b>
<b>List of Figures</b>	<b>vi</b>
<b>List of Tables</b>	<b>viii</b>
<b>Abbreviations</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Theory</b>	<b>3</b>
2.1 Arduino boards . . . . .	3
2.2 Hook's law . . . . .	5
2.3 Harmonic motion . . . . .	6
2.4 Newtons second law . . . . .	7
2.5 Dynamics modeling of system . . . . .	8
2.5.1 Spring . . . . .	8
2.5.2 Damper . . . . .	10
2.5.3 Forced harmonic vibrations . . . . .	12
2.5.4 Support motion . . . . .	15
2.5.5 System with two degrees of freedom . . . . .	17
2.5.6 Rayleigh damping . . . . .	19
2.6 Piezoelectric actuators . . . . .	19
<b>3 Test bench and virtual model setup</b>	<b>24</b>
3.1 Choosing the actuator . . . . .	26
3.2 Electrical setup . . . . .	28
3.3 Mechanical setup . . . . .	30
3.3.1 Main mass . . . . .	31
3.3.2 Counter mass . . . . .	32
3.3.3 Rubber sandwich . . . . .	33
3.3.4 Spring . . . . .	36

3.4	The simulation setup . . . . .	39
3.4.1	Rubber modeling . . . . .	41
3.4.2	The actuator . . . . .	45
<b>4</b>	<b>Tests, calibrations and scripts</b>	<b>50</b>
4.1	Noise in the circuit . . . . .	51
4.1.1	Noise between accelerometer and Arduino . . . . .	51
4.1.2	Noise in the rest of the circuit . . . . .	52
4.2	Time used by functions in script . . . . .	53
4.3	Accelerometer . . . . .	55
4.4	Scripts used in the Arduino Due . . . . .	62
4.4.1	Frequency and amplitude estimation . . . . .	62
4.4.2	Stroke of actuator . . . . .	63
4.4.3	Regulating output signal . . . . .	64
4.4.4	Initiating the signal . . . . .	67
4.5	Adjusting properties in the simulations . . . . .	69
<b>5</b>	<b>Results</b>	<b>73</b>
5.1	Active damping simulations . . . . .	73
5.1.1	Amplitude of stack displacement . . . . .	74
5.1.2	Phase specter resulting in vibration reduction . . . . .	76
5.1.3	Change in frequency affects the delay specter . . . . .	77
5.1.4	Effect of change in amplitude . . . . .	77
5.2	Active damping regulation . . . . .	79
5.2.1	Measurement of frequencies . . . . .	79
5.2.2	Reading the frequency with Arduino . . . . .	80
5.2.3	Calculating the acceleration with Arduino . . . . .	80
5.2.4	Applying regulating signal in correct phase . . . . .	81
<b>6</b>	<b>Discussion</b>	<b>84</b>
6.1	Simulations . . . . .	84
6.2	Test bench as demonstrator for active damping . . . . .	87
6.3	Arduino as control mechanism . . . . .	90
<b>7</b>	<b>Conclusion and Further Work</b>	<b>93</b>
<b>A</b>	<b>Appendix</b>	<b>95</b>
	<b>Bibliography</b>	<b>99</b>

# List of Figures

2.1	Layout of the Arduino Due board . . . . .	3
2.2	PWM used to represent a quasi-sine wave form . . . . .	4
2.3	Acceleration-frequency relationship with $420\mu\text{m}$ amplitude . . . . .	8
2.4	Different types of damping models[1] . . . . .	9
2.5	Model of mass-spring system . . . . .	9
2.6	Model of mass-spring-damper system . . . . .	10
2.7	Vector relationship between spring-, damping force and force caused by acceleration . . . . .	13
2.8	Phase shift between applied force and oscillation of mass . . . . .	14
2.9	Amplified force in a mass-spring-damper system with damping coefficients from 0.01 to 0.1 . . . . .	15
2.10	Model of a mass-spring-damper system with oscillating base . . . . .	15
2.11	Model of a two mass-spring system . . . . .	17
2.12	Typical hysteresis of Piezoelectric stack[2] . . . . .	21
2.13	Typical relationship between applied force and stroke of a piezoelectric actuator[2] . . . . .	21
2.14	Working curve of piezoelectric stack with mass load[2] . . . . .	22
3.1	Simple model of the dynamic representation . . . . .	25
3.2	Actuator without protective cover . . . . .	26
3.3	Simplified layout of the electrical circuit . . . . .	28
3.4	Simple model of the test bench . . . . .	30
3.5	Design of the main mass . . . . .	32
3.6	Design of the rubber sandwich . . . . .	35
3.7	Illustration of distance the bolt can be screwed . . . . .	37
3.8	Physical test bench . . . . .	38
3.9	Components in the FEDEM simulation . . . . .	39
3.10	Deformations from simulation in NX, fixed bottom and free displacement in x-y plane on top . . . . .	42
3.11	Deformations from simulation in NX, fixed bottom and fixed in x-y plane on top . . . . .	42
3.12	Deformations in eigenmode analysis with RBE2 elements . . . . .	44
3.13	Deformations in eigenmode analysis with RBE3 elements . . . . .	44
3.14	Compression of rubber . . . . .	45
3.15	Render of the flexure . . . . .	46

3.16	Comparing displacements of the sphere. Flexure modelled as one part compared with eight parts . . . . .	46
3.17	FEDEM model of the flexure in eight parts . . . . .	47
3.18	Displacement of sphere with force and displacement regulation using values from data sheet. . . . .	48
4.1	Script for measuring the time consumption for code instructions . . . . .	53
4.2	Signal from ADXL335 at 1g acceleration at various frequencies . . . . .	56
4.3	Signal from ADXL335 while mounted on shaker with 10Hz frequency and varying acceleration . . . . .	56
4.4	Signal from ADXL335 while mounted on shaker with 50Hz frequency and varying acceleration . . . . .	57
4.5	Signal from ADXL335 while mounted on shaker with 100Hz frequency and varying acceleration . . . . .	57
4.6	Signal from ADXL335 while mounted on shaker with 150Hz frequency and varying acceleration . . . . .	58
4.7	Relationship between acceleration and analog read value . . . . .	59
4.8	Relationship between analogRead()-value amplitude and frequency . . . . .	60
4.9	Script calculating the frequency . . . . .	63
4.10	Voltage displacement relationship of actuators . . . . .	65
4.11	Script for the $\omega t$ -sine representation . . . . .	66
4.12	Script for the matrix-element sine representation . . . . .	67
4.13	Script determining when to initiate the regulating signal . . . . .	68
4.14	Results from Vibration View, frequency sweep at 0.1g acceleration without damping components . . . . .	69
4.15	Results from FEDEM, frequency sweep at 0.1g acceleration without damping components before adjustments . . . . .	70
4.16	Results from FEDEM, frequency sweep at 0.1g acceleration without damping components after adjustments . . . . .	71
4.17	Results from Vibration View, frequency sweep at 0.1g acceleration with damping components . . . . .	71
4.18	Results from FEDEM, frequency sweep at 0.1g acceleration with damping components . . . . .	72
5.1	Amplitude for main body with varying amplitudes regulating the piezoelectric stack-function . . . . .	75
5.2	Fraction of phase vs. amplification of vibrations . . . . .	76
5.3	Displacement of main mass with and without damping by actuators . . . . .	77
5.4	Specter of phase delay with vibration reduction for different frequencies . . . . .	78
5.5	Maximum deviation from applied frequency fro oscilioscope and estimated frequency by Arduino . . . . .	81
5.6	Error occurring as a result of time used by code instructions. . . . .	82
5.7	Measured maximum error compared to calculated maximum error. . . . .	83
6.1	Maximum acceleration exerted by the shaker which the actuators can dampen optimally . . . . .	89

# List of Tables

2.1	Arduino boards with key properties . . . . .	5
3.1	Properties of PK2FVF1 actuator[3] . . . . .	27
3.2	Properties of PK2FVP1 stack[3] . . . . .	27
3.3	Values used for calculating thickness of counter mass . . . . .	33
3.4	Cross section area dimensions for rubber . . . . .	34
3.5	Material properties for different rubbers . . . . .	34
3.6	Material properties used for the parts used in the simulations . . . . .	39
3.7	Deviation in properties used in parts in simulation compared with data sheet	48
4.1	Data from noise test of the ADXL326 accelerometer . . . . .	52
4.2	Data from noise test of the ADXL335 accelerometer . . . . .	52
4.3	Time in $\mu\text{s}$ used by code instructions . . . . .	54
4.4	Tests conducted to investigate accelerometer data precision . . . . .	55
4.5	The percentage change of analogRead()-value due to increasing acceleration	59
4.6	$R^2$ values for the different accelerations . . . . .	61
4.7	Applied voltage and corresponding displacement of actuators . . . . .	64
4.8	analogWrite()value, arduino voltage, amplified voltage . . . . .	64
4.9	Delay in matrix-element script and corresponding frequency. . . . .	66
5.1	Input for sine function governing the base excitation function(SI units) .	74
5.2	Frequencies and corresponding minimum and maximum phase delay . .	78
5.3	Reduction(negative values) and gain(positive values) of displacement amplitude . . . . .	79
5.4	Acceleration calculated with Arduino . . . . .	81

# Abbreviations

<b>PWM</b>	Pulse Width Modulation
<b>FBD</b>	Free Body Diagram
<b>PZT</b>	Lead Zirconate Titanate
<b>DOF</b>	Degrees of Freedom
<b>MBS</b>	Multi Body Simulations
<b>DAC</b>	Digital to Analog Converter

# Chapter 1

## Introduction

Vibrations are a common problem during machining and reduces productivity and performance of the machine tools which also contributes to reduce the tool life[4][5]. Reducing these vibrations by damping is a common way to address the problem. Passive damping methods have been used for a long time, the first tuned mass damper was patented in 1909[6]. The most common passive dampers for damping the amplitude of the resonance vibrations are tuned mass dampers and vibration absorbers[1]. The drawback of these dampers is that they are adjusted for vibrations around a given frequency. Structures with more than one natural frequency can only be optimized for one of the frequency, the other frequencies will not be optimally damped. Therefore active damping and a combination of passive and active damping have been used [7]. Active damping most commonly use piezoelectric motors[4][8][7][9][10] or voice coil motors to generate the force used to dampen vibrations[5][11]. Different regulating techniques have been used to control the actuators and can be very complex[7][4][11]. Since the frequencies of the vibrations in machining tools are usually high, the sensors and regulating system have to be responsive and precise. Most laboratory equipment with millisecond accuracy and real time operations tend to be expensive. Arduino boards on the other hand have become cheap, accurate and precise. The development and major improvements of the Arduino boards and the compatible sensors have made them suitable for some laboratory uses[12].

The goal of this paper is, in collaboration with Teeness AS, to design and produce a test bench used to investigate and demonstrate active damping. The paper also explore the possibility if the system can be regulated using an Arduino as the control mechanism.

A test bench for active damping is designed using the Siemens PLM software NX and produced at Teeness AS in Trondheim. The test bench is designed as a structure of two

masses, one large and one small. The large mass is placed on a rubber cylinder fastened to a shaker. The base excitation of the shaker causes vibrations to the large mass and actuators, fastened to the large mass, regulate the displacement of the small mass to dampen the vibrations of the large mass.

Scripts on an Arduino Due board were used to investigate whether the regulation of the active damping system could be performed with satisfying results. The data used in the scripts are read from Arduino compatible components.

Multi body simulations(MBS) are simulations that are composed of elastic bodies and a relation described between them. The test bench consists of many parts and MBS are therefore used to investigate the properties. FEDEM was the chosen MBS program used for simulating the virtual the bench. Components were investigated and modified by them selves and later joined together in simulations containing all parts.

# Chapter 2

## Theory

### 2.1 Arduino boards

Arduino boards are cheap multipurpose micro controller boards that are designed to be user friendly[13]. On the boards there are pins to connect to components as sensors or actuators. The digital pins are I/O pins, meaning that they have two states, on and off. While the I/O pin is on, a current is sent through the pin usually, 5V or 3.3V, depending on the board and its configurations. The I/O pins can also measure voltages where the signal is read as either "High"(1) when there is a measurable current or "Low"(0) otherwise.

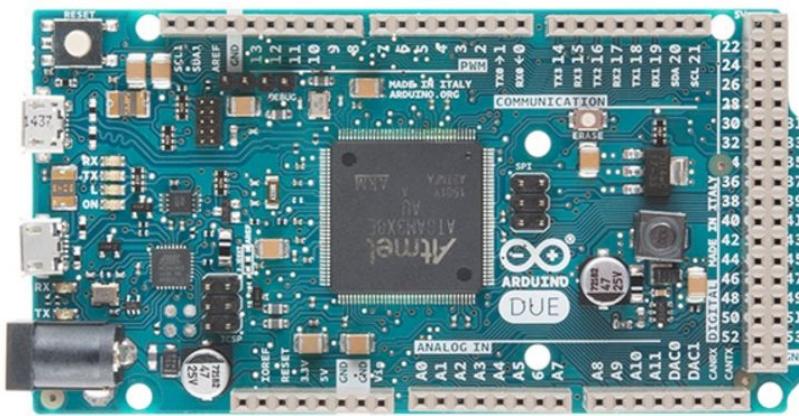


FIGURE 2.1: Layout of the Arduino Due board.

The analog input pins can measure the current and return a value between 0 and the highest number the resolution of the analog to digital converter(ADC) allows. This value is function of the bit resolution of the board and is given by equation 2.1.

$$\text{Resolution} = 2^{\text{bits}} \quad (2.1)$$

Some pins have the possibility of using pulse width modulation(PMW) and can give an average output between 0V and maximum output voltage of the board. The resolution for these regulations are also determined by equation 2.1.

PWM is a technique using digital means to represent analogue values. The current is switched on and off at a high frequency and the relationship between the on/off time determines the average current. With this control mechanism a quasi-sine wave form can be generated[14]. An analog signal would give the desired voltage as a continuous signal and not a signal that alters between 0V and the maximum voltage the board can deliver.

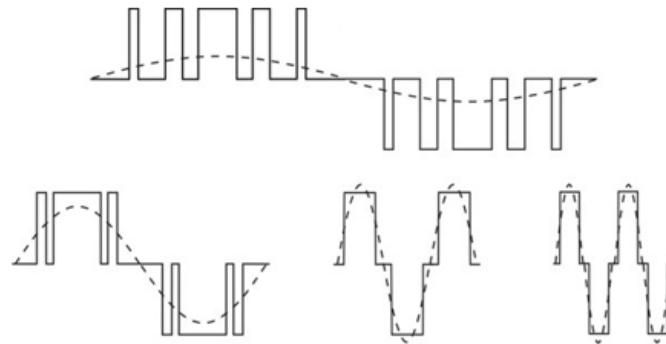


FIGURE 2.2: PWM used to represent a quasi-sine wave form.

There are over 20 different original Arduino boards and countless clones. Table 2.1 shows some of the Arduino boards available and some of their key properties[15]. The CPU speed determines how many instructions per second the Arduino is capable of executing. The flash memory (program space), is where the Arduino sketch is stored, while the SRAM (static random access memory) is where the sketch creates and manipulates variables when it runs[16]. A sketch is the name that Arduino uses for a program. It is the code that is uploaded to and run on an Arduino board[17]. "Analog in/out" in the table shows how many digital to analog converter pins the board has. For the Arduino Due these pins provide true analog outputs with 12-bits resolution (4096 levels) with the `analogWrite()` function and not a value represented by using PWM. The DAC output range is actually from 0.55V to 2.75V on the Due.

TABLE 2.1: Arduino boards with key properties

Name	CPU	Analog in/out	Bit resolution	SRAM[kB]	Flash[kB]
Uno	16 MHz	6/0	8	2	32
Due	84 MHz	12/2	12	96	512
Mega	16 MHz	16/0	8	8	256
Mini	16 MHz	8/0	8	2	32

Arduino uses a scripting language that is based on C/C++ and open source libraries with scripts are available on-line[18]. There is a large open source community, where troubleshooting and examples of scripts are posted. Some instructions in the codes in the Arduino language such as "micros()", which returns the number of microseconds since the Arduino started running, has a resolution dependent on the CPU speed.

There are many Arduino compatible components and modules available, making prototyping convenient as there is a large variety of components to choose from. There are analog devices which are passive, and there are modules that have small processors that process data before sending the data to the Arduino board. Signals from components are sent digitally, analogue or via serial communication.

## 2.2 Hook's law

Using springs, the relationship between applied force and the displacement of a homogeneous material was discovered to be dependent on the stiffness of the material [19]. Hook's law states that the force by a solid (in compression or tension) is proportional to the product of the body's stiffness and its displacement. This can be generally expressed by equation 2.2[20].

$$\text{Force} = \text{Stiffness} * \text{Displacement} \quad (2.2)$$

Hook's law is valid for linear elastic materials. The unload curve of these materials follow the load curve, while no plasticity has occurred. Equation 2.2 is valid for both compression and tension as long as the material has linear elastic behaviour. The direction of the force by the body is in the opposite direction of applied displacement relative to its equilibrium.

Hooke's law can be written as

$$\sigma = E\epsilon \quad (2.3)$$

Where  $\sigma$  is the stress,  $E$  is the Young's modulus and  $\epsilon$  is strain. The relationship between the axial force,  $F$ , on a solid and the stress is shown below[21].

$$F = \sigma A \quad (2.4)$$

Where  $A$  is the cross section area of the body. Strain is defined as the length of the deformation relative to the original length.

$$\epsilon = \frac{\delta}{L} \quad (2.5)$$

Combining equations 2.3, 2.4 and 2.5 yields equation 2.6[22].

$$F = \frac{\delta EA}{L} \quad (2.6)$$

Equation 2.6 shows the relation between compression- or tension force needed to deform a body with a constant cross section area. This means that the material stiffness can be expressed by

$$k = \frac{EA}{L} \quad (2.7)$$

The pressure needed to compress a solid body with the displacement  $\delta$  will then be expressed with

$$p = \frac{F}{A} = \frac{AE\delta}{AL} = \frac{E\delta}{L} \quad (2.8)$$

The relationship between the modulus of elasticity and the shear modulus in isotropic materials is given by equation 2.9.[23]

$$G = \frac{E}{2(1 + \nu)} \quad (2.9)$$

## 2.3 Harmonic motion

When the motion of an object is repeated regularly in equal intervals of time  $\tau$ , its motion is considered as periodic. The time for one repetition of the motion is called the period, denoted here by  $\tau$ . Its reciprocal is called the frequency,  $f$ .

$$f = \frac{1}{\tau} \quad (2.10)$$

Harmonic motion is the simplest form of periodic motion [24] and can be expressed by

$$x = A \sin(2\pi \frac{t}{\tau}) \quad (2.11)$$

The angular velocity is denoted with  $\omega$  and measured in radians per second.

$$\omega = 2\pi f \quad (2.12)$$

By combining equation 2.12 and 2.10 we get the equation for harmonic motion

$$x = A \sin(\omega t) \quad (2.13)$$

with its derivatives which describe the velocity and the acceleration of the harmonic motion.

$$\dot{x} = \omega A \cos(\omega t) \quad (2.14)$$

$$\ddot{x} = -\omega^2 A \sin(\omega t) = -\omega^2 x \quad (2.15)$$

From this we can see that the acceleration is heavily affected by the angular velocity and thereby the frequency of the harmonic motion. The relationship between acceleration and frequency is represented in figure 2.3. Angular velocity is used to describe the frequency using radians pr second.

## 2.4 Newtons second law

The force needed to accelerate an object is proportional to the product of the objects mass and the acceleration. Thus;

$$F = ma \quad (2.16)$$

Where F represents force, m represents mass, and a represents the acceleration.

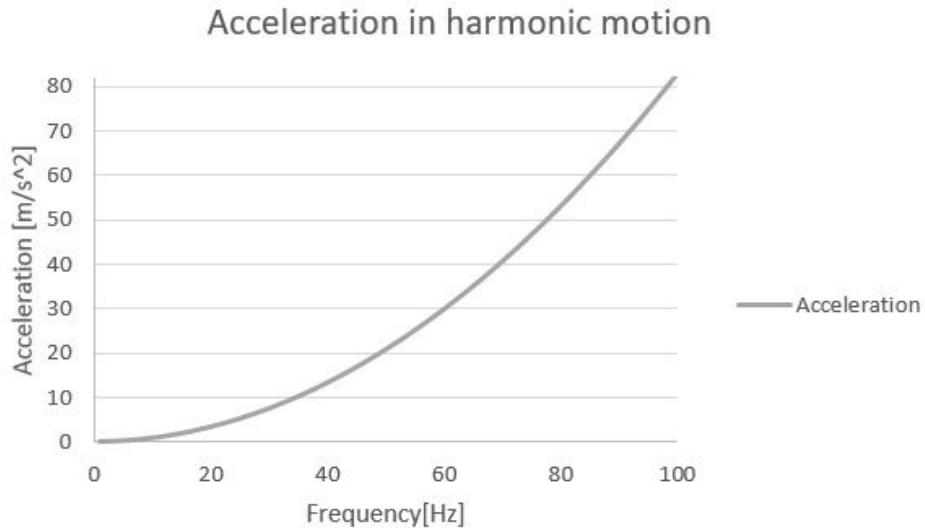


FIGURE 2.3: Acceleration-frequency relationship with  $420\mu\text{m}$  amplitude

## 2.5 Dynamics modeling of system

There are many different ways to regulate vibrations of a mass[1]. The motion of the body in a mass-spring-damper system is dependent on its mass, spring stiffness, damping coefficients and applied forces. Figure 2.4 shows different ways to regulate the oscillations in the main mass by damping. In the representation shown in the figure, it is assumed that the ground has a harmonic single degree of freedom(DOF) motion and there are no external forces. The figure shows graphs of the amplification of a relative motion of the large mass, M, compared to the motion of the ground(y-axis) and the frequency of the oscillation of the support structure(x-axis). The peaks of the relative displacement occur on the natural frequency/frequencies of the systems.

### 2.5.1 Spring

Springs follow Hook's law and the force-displacement relation by a spring is given by equation 2.17

$$F = kx \quad (2.17)$$

Where F represents force, k represents the spring stiffness, and x represents the displacement[25].

When looking at a mas spring system with no damper or external forces acting on the system, we can obtain the equations of motion from the free body diagram(FBD). The

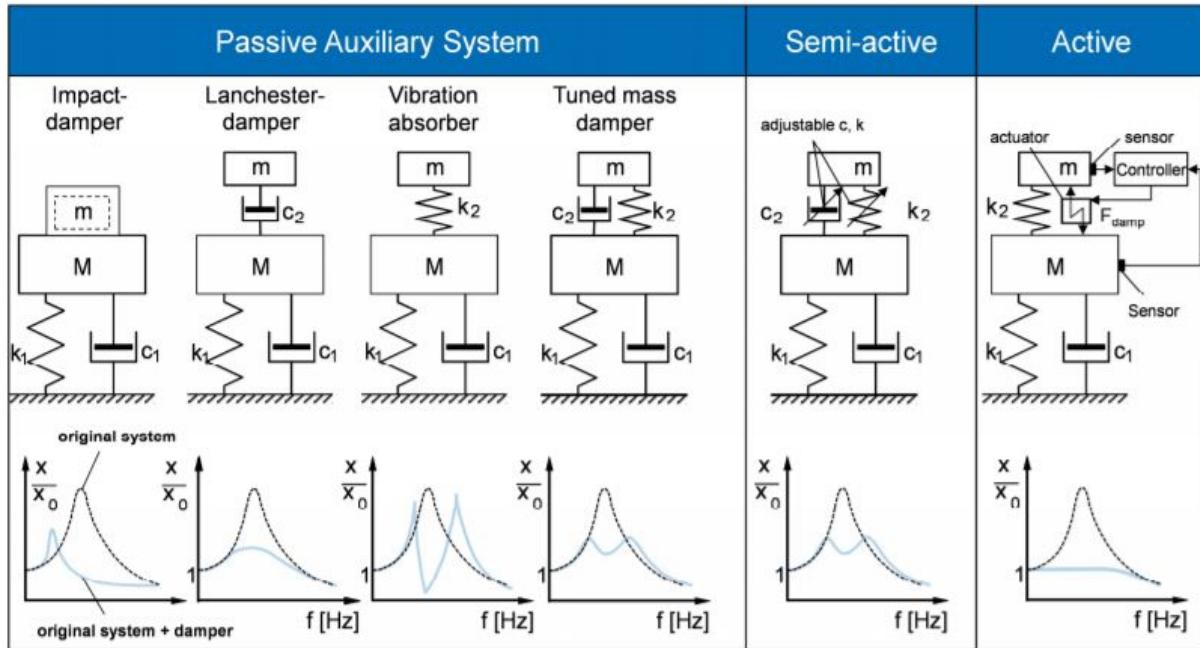


FIGURE 2.4: Different types of damping models[1]

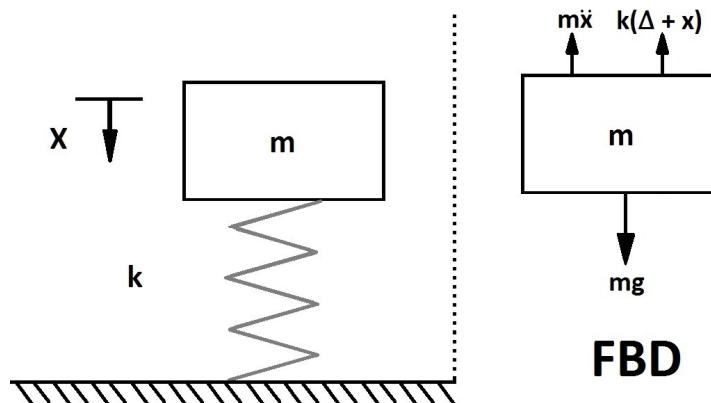


FIGURE 2.5: Model of mass-spring system

equation of motions is obtained by balancing the forces acting on the body. When the mass is stationary, the gravitational force,  $mg$  is equal to the reaction force by the spring given by equation 2.17, thus;

$$k\Delta = mg \quad (2.18)$$

When the displacement of the mass relative to its stationary position is denoted by  $x$ , and using Newton's second law of motion 2.16 we obtain the equation of motion by looking at the FBD.

$$m\ddot{x} = mg - k(\Delta + x) \quad (2.19)$$

Subsituting equation 2.18 into eqation 2.19 we obtain

$$m\ddot{x} = -kx \quad (2.20)$$

By including equation 2.15 we get

$$-m\omega^2x = -kx \quad (2.21)$$

solving for  $\omega$

$$\omega = \sqrt{\frac{k}{m}} \quad (2.22)$$

Equation 2.22 gives the natural frequency when there is no external forces on the single degree of freedom(DOF) system, like the mass-spring system shown in figure 2.5. A degree of freedom represents the mode of motion a system can move. In general a unconstrained system can move in 6 DOFs, 3 translational and 3 rotational.

### 2.5.2 Damper

Figure 2.6 shows a mass-spring-damper system with its FBD on the right. Equation 2.18 is considered in figure 2.6 and both the  $mg$  and the  $k\Delta$  term are removed from the FBD since they cancel each other out.

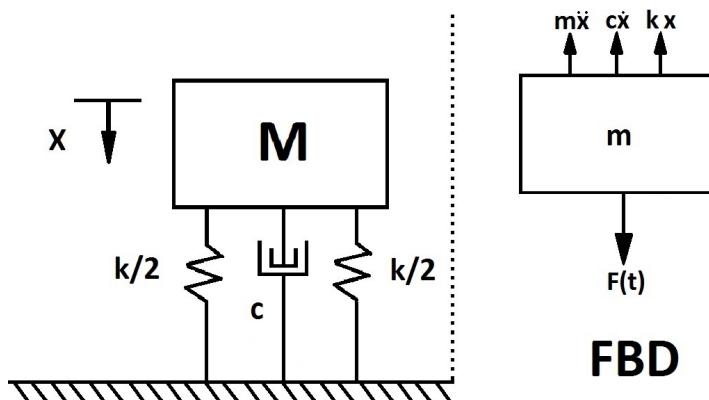


FIGURE 2.6: Model of mass-spring-damper system

The damping force is expressed by equation 2.23. The direction of the damping force is in the opposite direction of the motion.

$$F_d = c\dot{x} \quad (2.23)$$

This shows that the force is linearly dependent on the velocity scaled by the damping coefficient,  $c$  [24]. The symbol for a damper is a dash pot as shown between the springs in figure 2.6. The equation of motion for the mass-spring-damper system is given by equation 2.24.

$$F(t) = kx + c\dot{x} + m\ddot{x} \quad (2.24)$$

The equation for the free-damped vibration is given by the solution of the homogeneous differential equation 2.25.

$$0 = kx + c\dot{x} + m\ddot{x} \quad (2.25)$$

By assuming the solution and its derivatives to be given by 2.26,

$$x = e^{st}, \dot{x} = se^{st}, \ddot{x} = s^2e^{st} \quad (2.26)$$

and implementing this in equation 2.25 we get

$$0 = ke^{st} + cse^{st} + ms^2e^{st} \quad (2.27)$$

This holds for every value of  $t$ , when

$$0 = s^2 + \frac{c}{m}s + \frac{k}{m} \quad (2.28)$$

and is known as the characteristic equation for the system. The two roots to the characteristic equation is given by

$$s_{1,2} = -\frac{c}{2m} \pm \sqrt{\left(\frac{c}{2m}\right)^2 - \frac{k}{m}} \quad (2.29)$$

Where the general solution is given by the equation

$$x = Ae^{s_1 t} + Be^{s_2 t} \quad (2.30)$$

By combining the expressions for  $s_1$  and  $s_2$  from equation 2.29 to 2.30 we can rewrite 2.30.

$$x = e^{-\frac{c}{2m}t} (Ae^{\sqrt{((\frac{c}{2m})^2 - \frac{k}{m})}t} + Be^{-\sqrt{((\frac{c}{2m})^2 - \frac{k}{m})}t}) \quad (2.31)$$

By investigating equation 2.31 we can see that when the term  $(\frac{c}{2m})^2 - \frac{k}{m}$  is positive, the exponents in the large parenthesis in equation 2.31 are real numbers and therefore equation 2.31 can only yield positive values. The first term in equation 2.31  $e^{-\frac{c}{2m}t}$  is a decreasing function when  $t$  increases. When the  $(\frac{c}{2m})^2 - \frac{k}{m}$  term is positive, the magnitude of the decreasing exponent in the function is reduced and the system is referred to as a over damped system[26].

If the exponent in equation 2.31 is complex, the equation has both positive and negative values, meaning oscillation will occur. This case is referred to as under damped.

If  $(\frac{c}{2m})^2 - \frac{k}{m}$  is 0, the motion is described by  $e^{-\frac{c}{2m}t}$  and the amplitudes A and B. This case is called critical damped and occurs when

$$\left(\frac{c}{2m}\right)^2 = \frac{k}{m} \quad (2.32)$$

The damping coefficient yielding critical damping is therefor given by

$$c_c = 2m\omega_n = 2\sqrt{km} \quad (2.33)$$

Damping can then be expressed in relation to the critical damping and called damping ratio and is denoted with  $\xi$ . Materials have an internal damping that is often denoted by using a %-based value.

$$\xi = \frac{c}{c_c} \quad (2.34)$$

### 2.5.3 Forced harmonic vibrations

As seen from previous subsection the FBD in figure 2.6 show us that the equation of motion can be expressed by by equation 2.35 [25]. During forced harmonic vibrations function describing the external forces  $F(t)$  is not equal to 0.

$$F(t) = kx + c\dot{x} + m\ddot{x} \quad (2.35)$$

The function describing the forced harmonic vibrations is expressed with an amplitude of  $F_0$  and an angular velocity is given by  $\omega t$ , the differential equation describing the motion of the FBD will be 2.36

$$F_0 \sin(\omega t) = kx + c\dot{x} + m\ddot{x} \quad (2.36)$$

We can assume the particular solution to be of the form[24].

$$x = X \sin(\omega t - \phi) \quad (2.37)$$

Where  $X$  is the amplitude,  $\phi$  is the phase and  $x$  expresses the displacement. The velocity is given by 2.38 and the acceleration is given by 2.39.

$$\dot{x} = \omega X \cos(\omega t - \phi) \quad (2.38)$$

$$\ddot{x} = -\omega^2 X \sin(\omega t - \phi) \quad (2.39)$$

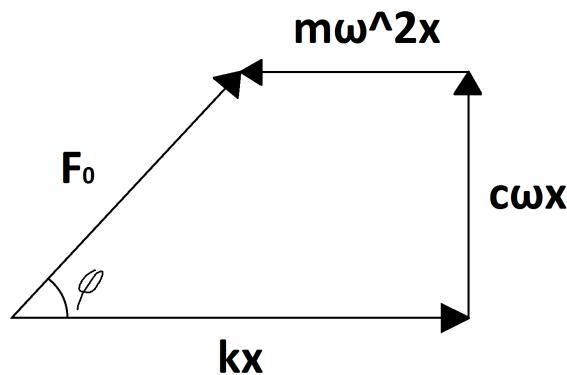


FIGURE 2.7: Vector relationship between spring-, damping force and force caused by acceleration

The vector relationship of the force components of  $F_0$  in equation 2.36 when inserting 2.37, 2.38 and 2.39 is shown in figure 2.7. The Pythagorean theorem gives the relationship between the force and displacement/velocity/acceleration. We can see that the angle  $\phi$  is determined by the relationship between  $kx - m\omega^2$  and  $c\omega$ . The phase represents a delay

in the oscillation of the mass applied force. Figure 2.8 shows the motion of the mass and the applied force with a 45 degree phase shift, which is when  $c\omega$  is equal to  $kx - m\omega^2$ .

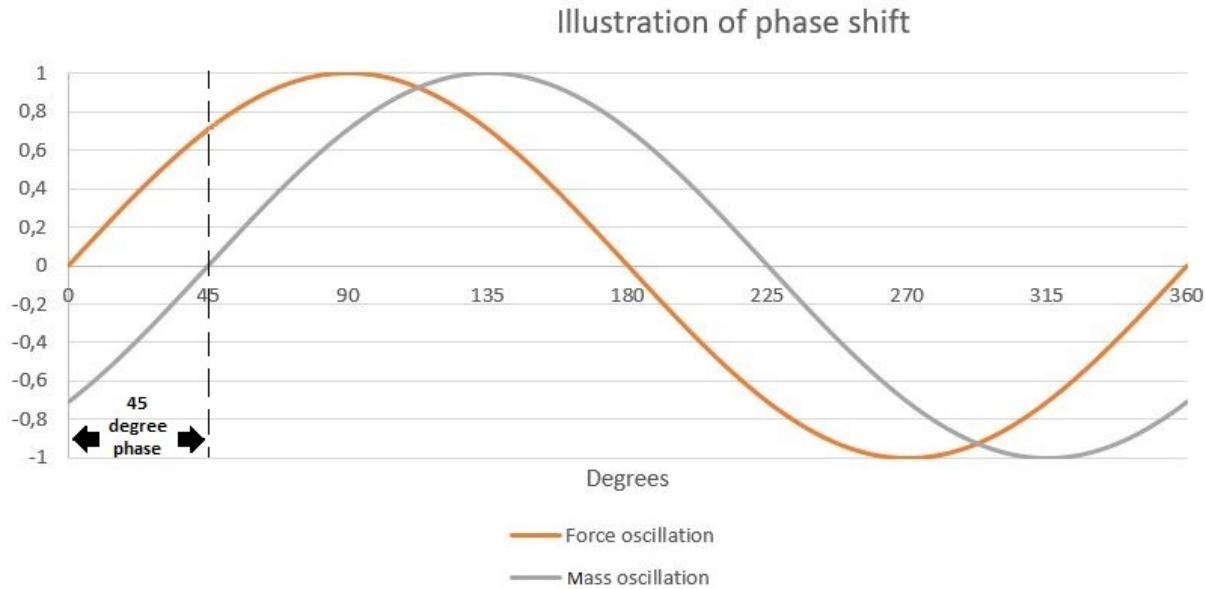


FIGURE 2.8: Phase shift between applied force and oscillation of mass

$$F_0^2 = (X((k - m\omega^2))^2 + (c\omega)^2) \quad (2.40)$$

$$F_0^2 = X^2(k - m\omega^2)^2 + X^2(c\omega)^2 \quad (2.41)$$

$$F_0 = X \sqrt{(k - m\omega^2)^2 + (c\omega)^2} \quad (2.42)$$

$$X = \frac{F_0}{\sqrt{(k - m\omega^2)^2 + (c\omega)^2}} \quad (2.43)$$

Dividing the numerator and denominator by  $k$ , we obtain:

$$X = \frac{\frac{F_0}{k}}{\sqrt{(1 - \frac{m\omega^2}{k})^2 + (\frac{c\omega}{k})^2}} \quad (2.44)$$

When including equations 2.34, 2.33 and 2.22 and moving the  $\frac{F_0}{k}$  term over to the left side of the equation to make the equation non-dimensional we get

$$\frac{kX}{F_0} = \frac{1}{\sqrt{(1 - (\frac{\omega}{\omega_n})^2)^2 + (2\xi\frac{\omega}{\omega_n})^2}} \quad (2.45)$$

This equation describes the amplification of the force in the spring in proportion to the harmonic excited force on the mass. In other words this shows the amplified force on the mass due to the systems natural frequencies.

Equation 2.45 is shown in the figure 2.9 below with different damping ratios.

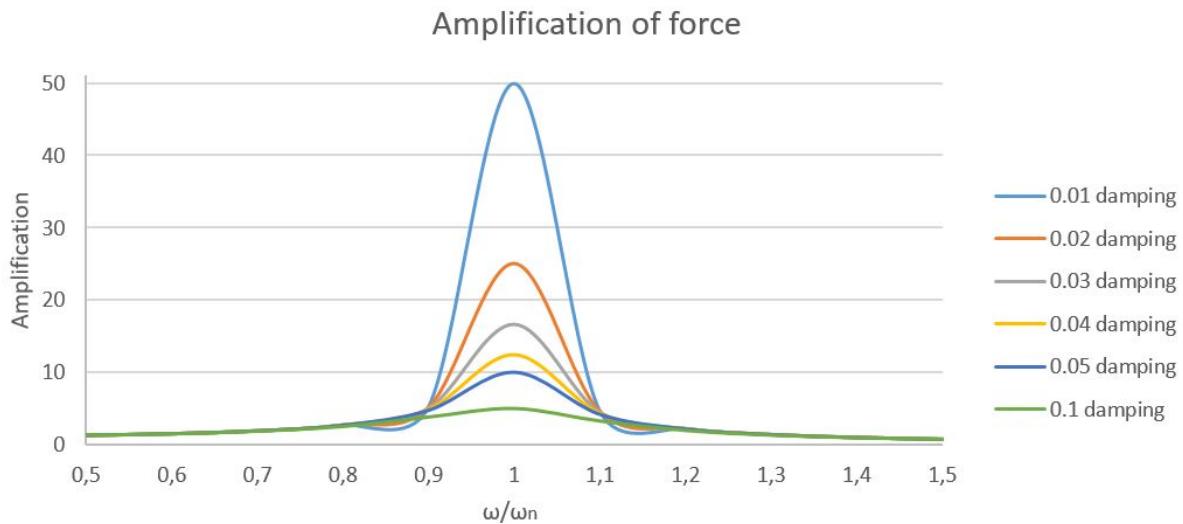


FIGURE 2.9: Amplified force in a mass-spring-damper system with damping coefficients from 0.01 to 0.1

#### 2.5.4 Support motion

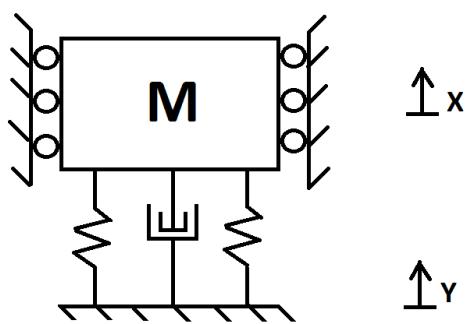


FIGURE 2.10: Model of a mass-spring-damper system with oscillating base

When the ground in which the mass is supported by is moving, the mass also will move. The motion of the mass relative to the ground will be given by equation 2.46.

$$z = x - y \quad (2.46)$$

The case where the motion of the ground is harmonic and is given by equation

$$y = Y \sin(\omega t) \quad (2.47)$$

$$\dot{y} = \omega Y \cos(\omega t) \quad (2.48)$$

$$\ddot{y} = -\omega^2 Y \sin(\omega t) \quad (2.49)$$

Using the FBD of the model, we can obtain the equation of motion.

$$m\ddot{x} = -k(x - y) - c(\dot{x} - \dot{y}) \quad (2.50)$$

Combining equations 2.50, 2.46 and 2.49 we can rewrite the equation to 2.51

$$m\ddot{z} + c\dot{z} + kz = m\omega^2 Y \sin(\omega t) \quad (2.51)$$

Equation 2.51 is similar to equation 2.36. The differences are that the variable x describing the motion of the body is substituted with z describing the relative distance between the body and the oscillating base.  $F_0$  is substituted for  $m\omega^2 Y$ . Making the same assumption as discussed in section 2.5.3, the amplitude is given by equation 2.52.

$$Z = \frac{m\omega^2 Y}{\sqrt{(k - m\omega^2)^2 + (c\omega)^2}} \quad (2.52)$$

The steady state amplitude can be derived by using exponential form to describe the motion and substituting into equation 2.50. This leads to the amplitude given by equation 2.53.

$$\left| \frac{X}{Y} \right| = \sqrt{\frac{k^2 + (\omega c)^2}{(k - m\omega^2)^2 + (c\omega)^2}} \quad (2.53)$$

### 2.5.5 System with two degrees of freedom

In a system with two masses connected by springs as shown in figure 2.11, the motion of one mass is dependent on the other. Once again by looking at the FBD we can obtain the equation of motion for both of the masses.

$$m_1 \ddot{x}_1 + (k_1 + k_2)x_1 - k_2x_2 = 0 \quad (2.54)$$

$$m_2 \ddot{x}_2 - k_2x_1 + k_2x_2 = 0 \quad (2.55)$$

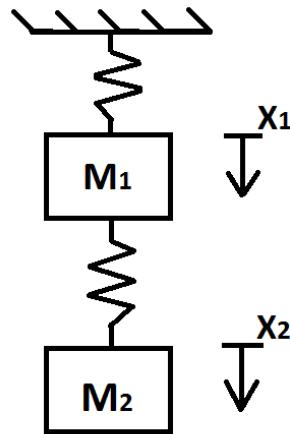


FIGURE 2.11: Model of a two mass-spring system

Writing equations 2.54 and 2.55 in matrix form yields equation 2.56

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{Bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{Bmatrix} + \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (2.56)$$

The displacements  $x$ , is then denoted as shown below.

$$\begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \sin(\omega t - \phi) \quad (2.57)$$

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{Bmatrix} = \omega \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \cos(\omega t - \phi) \quad (2.58)$$

$$\begin{Bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{Bmatrix} = -\omega^2 \begin{Bmatrix} a_1 \\ a_2 \end{Bmatrix} \sin(\omega t - \phi) \quad (2.59)$$

Combining the equations above gives

$$-\omega^2 \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \end{Bmatrix} \sin(\omega t - \phi) + \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \end{Bmatrix} \sin(\omega t - \phi) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (2.60)$$

Dividing by  $\sin(\omega t - \phi)$  gives

$$-\omega^2 \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \end{Bmatrix} + \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \end{Bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (2.61)$$

Collecting the terms will give

$$\begin{bmatrix} -\omega^2 m_1 + k_1 + k_2 & -k_2 \\ -k_2 & -\omega^2 m_2 + k_2 \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \end{Bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (2.62)$$

Solving for  $a_1$  and  $a_2$

$$\det \begin{vmatrix} -\omega^2 m_1 + k_1 + k_2 & -k_2 \\ -k_2 & -\omega^2 m_2 + k_2 \end{vmatrix} = 0 \quad (2.63)$$

Giving the characteristic equation 2.64

$$(-\omega^2 m_1 + k_1 + k_2)(-\omega^2 m_2 + k_2) - (-k_2)(-k_2) = 0 \quad (2.64)$$

simplifying this equation gives

$$-\omega^4(m_1 m_2) + -\omega^2(m_2 k_1 + m_2 k_2 + m_1 k_2) + k_1 k_2 = 0 \quad (2.65)$$

To find the natural frequency we have to find the roots of equation 2.65. When the two natural frequencies are found, we can values in equation 2.62. We can then calculate the amplitude ratio by dividing the amplitudes with each other.

### 2.5.6 Rayleigh damping

Rayleigh damping, or proportional damping, consists of two proportional relations, mass- and stiffness proportional damping. Rayleigh damping is given by equation 2.66.

$$C = \alpha_1 m + \alpha_2 k \quad (2.66)$$

Where  $\alpha_1$  and  $\alpha_2$  are constants given by equations 2.67 and 2.68,  $m$  is the mass and  $k$  the stiffness.  $\lambda_i$  are the damping ratios for the selected vibration modes which are expressed with the internal damping given in equation 2.34. This is used in FEDEM where needed theory is provided in the FEDEM theory guide.

$$\alpha_1 = \frac{2\omega_1\omega_2}{\omega_2^2 - \omega_1^2}(\lambda_1\omega_2 - \lambda_2\omega_1) \quad (2.67)$$

$$\alpha_2 = \frac{2(\lambda_1\omega_2 - \lambda_2\omega_1)}{\omega_2^2 - \omega_1^2} \quad (2.68)$$

## 2.6 Piezoelectric actuators

Piezoelectrics either produce a voltage in response to mechanical stress, direct mode, or a physical displacement as a result of an applied electrical field, indirect mode. Due to this property, piezoelectric materials are used in both actuators and sensors. One of the most common piezoelectric materials is lead-zirconate-titanate (PZT).

The piezoelectric effect, which can be described through a coupling the electric field equations with the strain tensor of HookeâŽs law that is shown below[8][9].

$$\begin{bmatrix} D \\ \epsilon \end{bmatrix} = \begin{bmatrix} e^\sigma & d^d \\ d^c & s^E \end{bmatrix} \begin{bmatrix} E \\ \sigma \end{bmatrix} \quad (2.69)$$

Here  $D$  is the electric displacement vector,  $\epsilon$  is the strain vector,  $E$  is the applied electric field vector,  $\sigma_m$  is the stress vector,  $e_{ij}^\sigma$  is the dielectric permittivity,  $d_{im}^d$  and  $d_{jk}^c$  are the piezoelectric coefficients, and  $s_{km}^E$  is the elastic compliance.

There are 3 main categories for PZT actuators: low voltage, high voltage, and ring actuators. For any given height,  $h$ , of an actuator, a stacked device can produce a displacement proportional to the number of stacks, as described by equation 2.70:

$$\Delta h = nd_{33}V \quad (2.70)$$

Here  $n$  is the number of stacks and the change in height is proportional to the number of stacks. These stacked structures require a low driving voltage and have a quick response time whilst having a high force generation. The driving voltage for one stack and the response time is the same as the whole stack structure because the stacks are coupled in parallel.

Low voltage actuators tend to have operating voltages below 200 V. These devices are monolithic stacks devices, meaning that the stacked structure is produced through sintering rather than gluing individual layers. They have an electrical capacitance on the order of a few  $\text{pF}$  and tend to have a high elastic modulus.

PZTs are often used where precision is key and are also capable of producing high forces while the stiffness is high. Piezoelectric actuator have been used by NASA to stabilize instruments during take off.

The hysteresis loop of a open-loop system will be similar to the one shown in figure 2.12[2]. The three different curves show the hysteresis loop with different maximum voltages applied. The hysteresis will be affected by the load applied and environmental conditions.

The force the actuator can generate and the stroke is dependent on the load applied. The graph below shows a typical relationship between the stroke and the force applied on the actuator.

When no forces are applied on the actuator, the maximum stroke is achieved when applying the maximum operating voltage. This stroke length is referred to as the free stroke  $\Delta L_{FS}$ . The block force,  $F_{Block}$  is defined as the force needed to be applied to compress the actuator back to its original length after the maximum operational voltage is applied. The stiffness of the actuator can be calculated by using the free stroke and the blocking force[2].

$$k_A = \frac{F_{Block}}{\Delta L_{FS}} \quad (2.71)$$

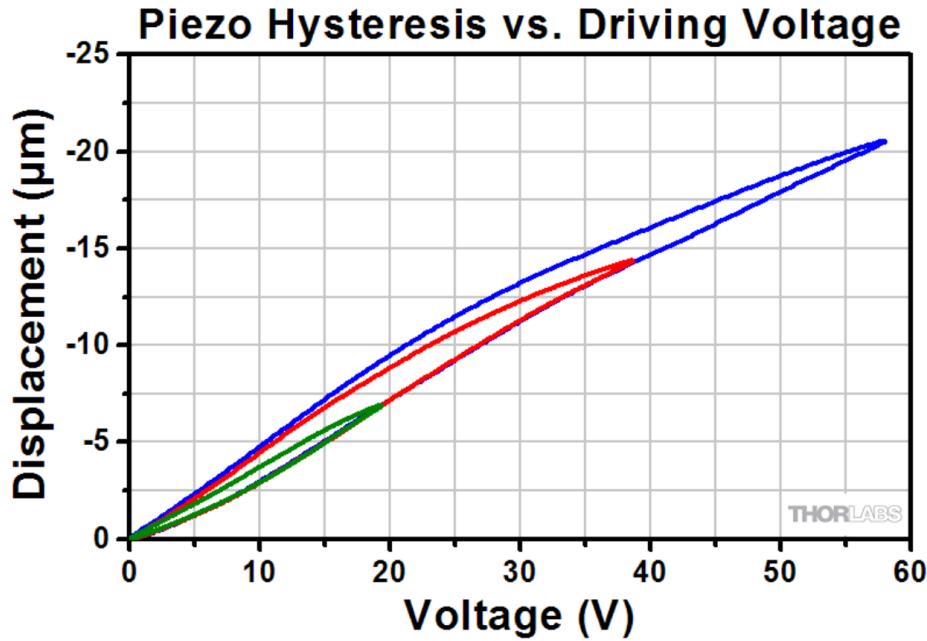


FIGURE 2.12: Typical hysteresis of Piezoelectric stack[2]

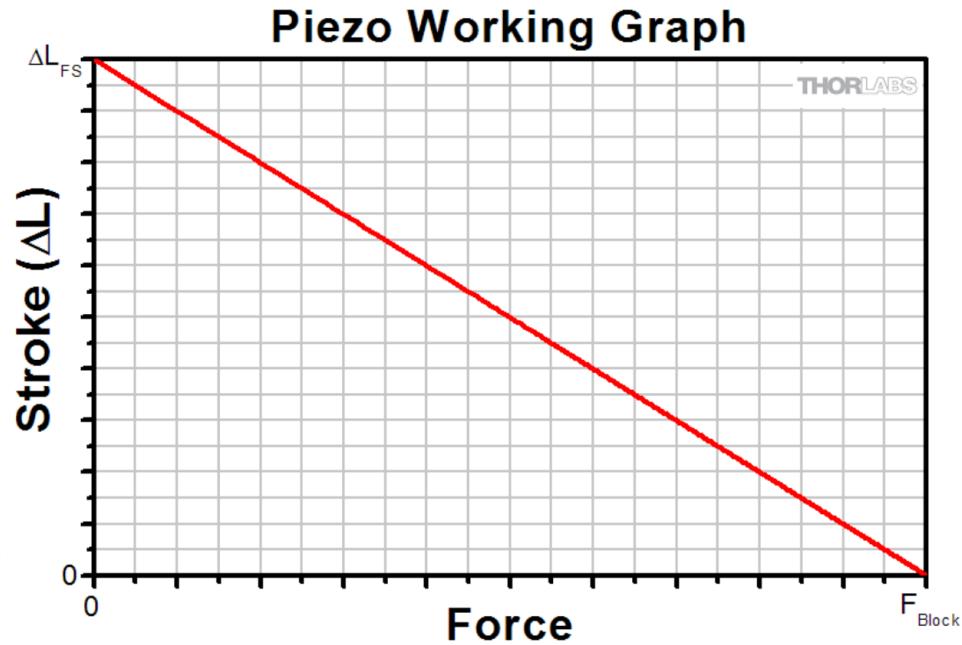


FIGURE 2.13: Typical relationship between applied force and stroke of a piezoelectric actuator[2]

The stroke and the force of the actuator is dependent on the load applied. If the load applied on the actuator behaves like a spring, the length of the stroke will be given by equation 2.71 and the force will be given by equation 2.73

$$\Delta L \approx \Delta L_{FS} \left( \frac{k_A}{k_A + k_L} \right) \quad (2.72)$$

$$\Delta F_{eff} = F_{Block} \left( \frac{k_L}{k_A + k_L} \right) \quad (2.73)$$

We see that the stroke is reduced when the stiffness of the load spring increases. This is because of the force the spring exerts back on the piezo.

If a mass is used as a load on the actuator, a stroke independent force is applied to the actuator. The steady state force applied would be  $m^*g$ . The work curve is shown in 2.14.

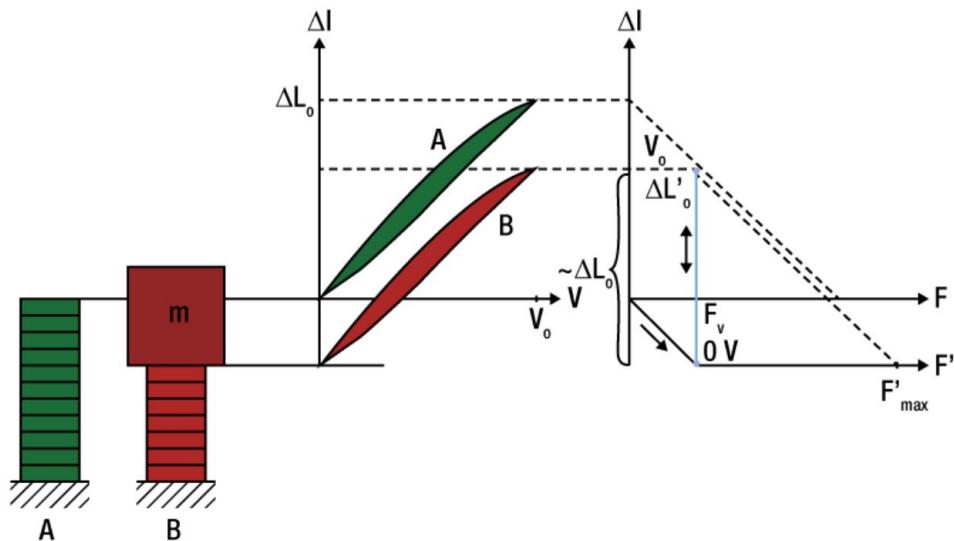


FIGURE 2.14: Working curve of piezoelectric stack with mass load[2]

In most systems the performance is going to be a combination of these cases. The spring stiffness of the preloaded spring load is recommended not to exceed a  $k_L \leq \frac{k_A}{10}$ .

A piezo system is usually operated in a frequency below its resonant frequency. The resonant frequency of the PZT is mainly dependent on the length of the stack and the mass of the preload. The equation below show how the resonant frequency is dependent on the preload mass.

$$f'_0 = f_0 \sqrt{\frac{\frac{m}{3}}{\frac{m}{3} + M}} \quad (2.74)$$

Where M is the preloaded mass, m is the mass of the actuator and  $f_0$  is the resonant frequency with no load. For primarily dynamic applications, the lowest resonant frequency of the piezo with preload must be well above the driving frequency of the piezo actuator[27]. The time it takes the piezo to expand is given by

$$t \approx \frac{1}{3f_0} \quad (2.75)$$

The maximum velocity of the actuator is dependent on the peak current of the power supply used to regulate the actuator. The acceleration response is dependent on the slew rate of the power supply.

The necessary current to drive the actuator in a sine motion is given by the formula 2.76

$$I_{Max} = \pi f C V_{pp} \quad (2.76)$$

The maximum power needed for the same operation is given by

$$P_{Max} = \pi f C V_{pp}^2 \quad (2.77)$$

Where C is the capacitance,  $V_{pp}$  is the peak-to-peak voltage and f is the frequency.

PZTs are brittle and have a low tensile strength. The stacks are not designed to withstand lateral, transverse or bending forces. Therefore most of the actuators have a mounting system for external loads that ensures that only compression is possible.

The driving power source has to be able to output the required voltage at the required slew rate and with the desired current. The slew rate is given by

$$\text{Slew rate} = \frac{\delta V}{\delta t} = \frac{I_{Max}}{C} \quad (2.78)$$

The maximum bandwidth of the amplifier is then given by

$$f_{Max} = \frac{I_{Max}}{\pi V_{PP} C} \quad (2.79)$$

# Chapter 3

## Test bench and virtual model setup

Firstly an actuator that is suitable to regulate the counter mass with desired speed and precision was chosen. The actuator had to be able to be regulated by simple means so that an Arduino board could be used as the control unit.

Secondly sensors and equipment that were used to drive the motors were chosen. The equipment to drive the actuator must satisfy the demands of the actuators.

After the components were chosen, the test bench was designed using Siemens NX. The design was made considering the chosen components and how to mount them. Machine drawings of the designed parts were made for production.

The virtual parts in NX were given material properties, meshed and used for MBS to investigate the dynamic and damping properties of the test bench.

The dynamic properties of the test bench as a whole were the main dimensioning criteria. Some off the criteria for the design are listed below.

1. The test bench was desired to have a natural frequency of around 100Hz-200Hz.
2. The mass of the test bench was desired to be fairly small and set to under 4kg.
3. Actuators have to be able to regulate the vibrations of the measured body.
4. The test bench has to be easily mounted on the shaker at Teeness' laboratory.

The general design of the test bench is a two mass system where each mass has one DOF in the same direction. The main mass was connected to a shaker by a rubber cylinder acting

like a spring-damper pair. The main mass will vibrate when the supporting structure vibrates. The vibrations of the main mass are the vibrations the active damping system is designed to reduce. A smaller mass, called the counter mass from here on out, was connected to the main mass with actuators between the masses.

The motion of the small mass was regulated by the voltage input to the actuator regulated by an Arduino board. The vibrations caused by regulating the actuators and thereby the counter mass is meant to counteract the vibrations of the main mass caused by the supporting structure. A simple representation of the model is shown in figure 3.1

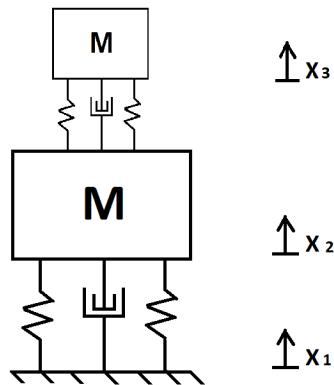


FIGURE 3.1: Simple model of the dynamic representation

To generate the support motion a shaker provided by Teeness is used. The shaker used in this thesis is a "Data Physics V350/DSA1-1k". Teeness use this shaker to measure the quality and performance of their products as well as for research and development. The shaker is capable to generate oscillations up to 4000Hz, which is more than sufficient for the use in this thesis. The shaker is regulated by a software called Vibration View. The program instructs the shaker to generate vibrations by either defining acceleration and frequency or displacement and frequency. Vibration View regulates the shaker and reads data from an accelerometer mounted on the oscillating part of the shaker. The data read from this accelerometer is used by Vibration View to tune the behaviour of the shaker.

### 3.1 Choosing the actuator

The test bench was modelled to fit the actuators regulating the counter mass. Therefore it was preferable that the actuator had a simple design with options for mounting. The desired frequencies in question are about 100Hz-200Hz, therefore the actuator had to be able to operate under these frequencies, preferably with a safety margin. The actuator was preferred to be fairly small, so that the actuator is easy to handle and that the designed test bench could be small. Reliability was also an important factor for ensuring an efficient regulating system. Furthermore the actuator had to provide a sufficient force to dampen the oscillations of the main body when it was subjected to the vibrations caused by the shaker. The actuator had to be able to be regulated by an Arduino and therefore have a simple regulating mechanism or a simple driver that could use the signals from the Arduino. The price was also important because the test bench was a prototype that had no guarantee for being successful.

With the criterion above in mind, an amplified piezoelectric actuator was chosen. The piezoelectric motors used in the test bench are three PK2FVF1 amplified piezoelectric actuator, shown in figure 3.2, from Thorlabs. This actuator type is fairly cheap and has all the desired properties mentioned above. The actuator consists of a piezoelectric stack module and a flexure. The piezoelectric stack changes its length when voltage is applied as described in section 2.6. The flexure increases the travel range through a lever amplification system. The increase in travel distance comes at the cost of reducing the maximum blocking force of the actuator. Key data of the actuator is presented in the table 3.1.



FIGURE 3.2: Actuator without protective cover

The properties of the piezoelectric stack which generates the force is given in table 3.2.

TABLE 3.1: Properties of PK2FVF1 actuator[3]

Property	Value
Drive voltage range	Maximum: 0-75 V
Displacement (Free Stroke) av 75V	$420\mu m + 15\%$
Maximum push force in motion direction	100N
Maximum pull force in motion direction	5N

TABLE 3.2: Properties of PK2FVP1 stack[3]

Property	Value
Drive voltage range	Maximum: 0-75 V
Displacement (Free Stroke) av 75V	$44.8\mu m + 15\%$
Maximum push force in motion direction	1000N
Capacitance	$16.5 \mu F$

As seen in the table 3.2, the driving voltage range is from 0V to 75V. Arduino boards can only deliver up to 5V. Therefore, an amplifier to amplify the signal from the Arduino became a necessity. To drive these actuators the amplifier had to be able to deliver sufficient current and be able to alter between these voltage levels as fast or faster than the Arduino. By rearranging equation 2.79, using the data in the data sheet for the actuator and assuming we want to be able to regulate the actuators up to 200Hz we can calculate the required current.

$$I_{Max} = f_{Max} \pi V_{PP} C = 200\text{Hz} \pi 75\text{V} 16.5\mu F = 0.78A \quad (3.1)$$

The power needed to drive the actuator is given by equation 2.77

$$P_{Max} = \pi 200 16.5\mu F (75V)^2 = 58.3\text{VA} \quad (3.2)$$

## 3.2 Electrical setup

The sensors have to be fast, since the oscillations investigated are at a frequency up to 200Hz. The desired number of measurements for each oscillation is 20. The sensors must therefore be able to output values every  $250\mu s$  with a satisfying resolution. Both precision and accuracy are desired, where precision is the most important. The script used to interpret data can adjust for the errors in the potential accuracy when sufficiently calibrated. The sensors also had to be Arduino compatible so that sensors and the Arduino can communicate. The sensors had to be small so that they could be mounted on the test bench without significantly changing the dynamic properties or design space.

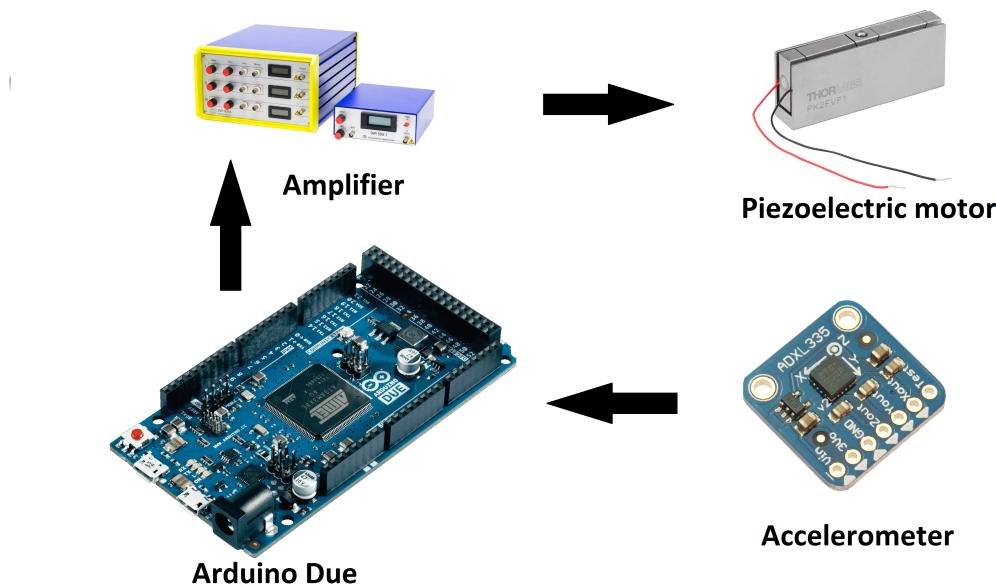


FIGURE 3.3: Simplified layout of the electrical circuit

Two different accelerometers were chosen as sensors for the test bench, the ADXL335 and the ADXL326. Both accelerometers are analog accelerometers meaning that they send out a voltage continually while powered. The voltage varies between 0 and the voltage used to power the sensor(1.8V-3.6V). The ADXL335 can measure accelerations from -3g to 3g, while the ADXL326 can measure accelerations from -16g to 16g.

The chosen Arduino used for controlling the active damping system was the Arduino Due. The cost of this board is 36 euro compared to the 20 euro the original Arduino(UNO). The Properties that sets the Due apart from the other board is that it has analog outputs and a faster CPU than the other boards. This means that the calculations and actions of the Arduino board is faster compared to the other boards. Further the resolution of the "micros()" - function is  $1\mu s$  compared to the resolution of UNO which returns values

of multiples of  $4\mu s$ . By having analog outputs the signal sent from the Arduino will be a continuous voltage output and not a PWM as shown in figure 2.2. Since the output from the Arduino has to be amplified to drive the actuators, a continuous signal seems like a more reliable choice of regulation.

The amplifier used in the circuit is a "Piezomechanic LE 200/150-1". The data sheet for the amplifier could not be found on the web page[28], so there is some uncertainty to its performance. The output voltage from the amplifier was measured with a multimeter before applied to the actuators. The slew rate/bandwidth is unknown for this amplifier. The demands for the amplifier are given 3.1 and 3.2.

The ADXL335 accelerometer was mounted on the main mass and was connected to the Arduino with the jumper wires that came with the official Arduino beginners kit. The ADXL326 accelerometer was used to read the acceleration of the counter mass and therefore mounted on the counter mass. The Arduino processes the signals from the ADXL335 accelerometer and sends out a signal that regulates the actuator via the DAC0 pin on the board. Since the DAC pins on the Arduino only can output voltages between 0.55V and 2.75V the signal has to be amplified. The amplifier is adjusted by manually adjusting the gain of the signal and the offset level. The output of the amplifier is directly coupled to the three actuators using a modified coax cable.

### 3.3 Mechanical setup

The model for testing the possibilities of active damping was designed using the computer software Siemens NX. Machine drawings were made in NX for the production of the designed parts which were made in the machine park at Teeness. The designs were made with proper clearances and tolerances. To make calculations easier, the representation of the test bench was a two mass spring and damper system as shown in figure 3.4. The main mass represented by  $M_1$  in the figure vibrates as a result of the oscillation of the support structure. The vibrations of the main mass represent the vibrations of a tool during machining. The goal is to reduce the vibrations of the main mass by regulating the motion of the counter mass,  $M_2$ . During testing these vibrations are generated by a shaker which the test bench is mounted on.

The main mass was designed to be made from left over material from the machine park. The spring-damper pair between the ground and the main mass shown in figure 3.4 is a simplified representation of the rubber sandwich in the designed test bench. The rubber used in the test bench was of the same type of rubbers that Teeness use in their dampers. The other spring-damper pair in figure 3.4 between the small and large mass represents the actuators.

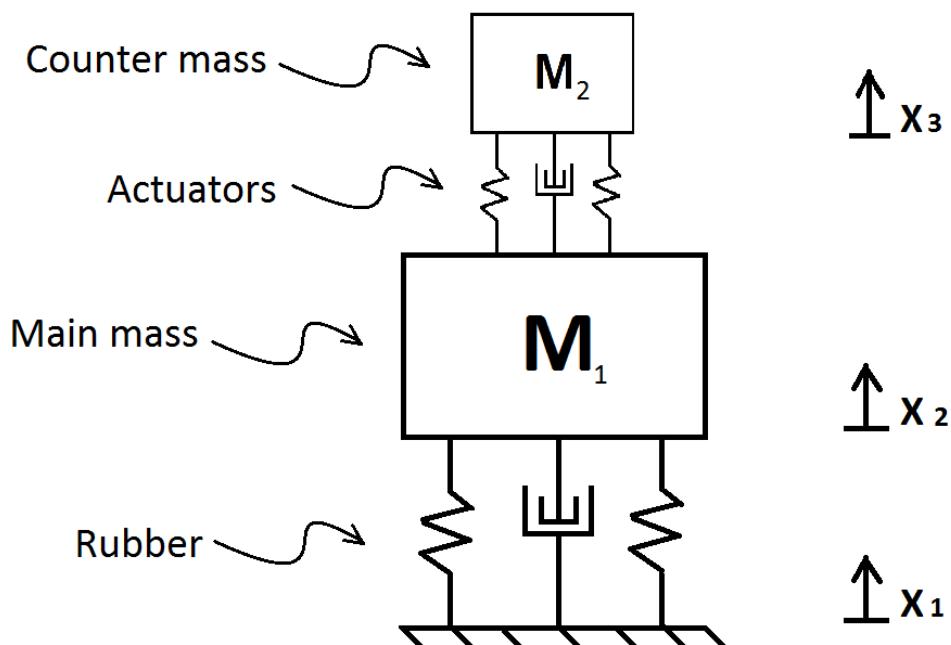


FIGURE 3.4: Simple model of the test bench

The actuators have a tungsten-carbide spherical contact on the top center of the motor, which is shown in figure 3.2. The forces applied by the actuator is designed to go through this

sphere. Since the force from each motor goes through the contact point connecting the sphere and the counter mass, balancing of the counter mass could be challenging with one motor. With one motor the counter mass would be free in all 3 rotational DOFs and extra boundary conditions would have to be made to keep the counter mass from sliding off. Three motors were chosen to make sure that the motion of the counter mass is translational and kept from sliding off, assuming all three motors behave identically.

Two accelerometers were mounted on the test bench. One on the counter mass and one on the main mass. The accelerometer on the main mass sends data which is used to regulate the actuators. The accelerometer on the top mass is there to enable the surveillance the acceleration of the counter mass.

### 3.3.1 Main mass

The main mass was made from left over material found at the workshop at Teeness. This was done to keep the costs low. Since Teeness AS specialise in damping tools used in metal working machines, including turning machines, they have bolts of both steel and aluminum which they use for production and testing. Their largest scrap bolts available were 100mm in diameter and this diameter was used to restrict the design space. The bolt used for the main mass was turned from a initial diameter of 100mm to 95mm, this was done to ensure the bolt was circular. With a cross section that is circular the CNC-machining is independent of the rotation of the cross section of the bolt and less time for calibrating the CNC-machine is needed.

Three slots were milled for the actuators. The design enables the actuator to slide in the slots, making it possible to regulate the distance from the center of the main mass and the sphere of the actuator. Each actuator have two threaded M2.5 holes on the bottom of the protective cover which were used for mounting the actuator to the main mass. Countersunk holes were drilled in the main mass in order to fasten the cover of the actuators to the main mass. The actuators can be mounted in three different places in the slot with different radii from the center of the main mass to the sphere of the actuator. The depth of the slots was made shallower than the height of the actuator. This was done so that the sphere on the actuator in its fully retracted position is higher than the main body. This was done to ensure that the counter mass does not collide with the main mass when in use. In the center of the main mass a M8 threaded hole was designed. This was done to be able to fasten a bolt that tightens a spring holding the counter mass in place. Three countersunk holes were made in the main mass for M5 screws to be able to fasten

to the rubber sandwich to the main mass. Two threaded holes were mad on the side of the main mass to mount the ADXL335 accelerometer.

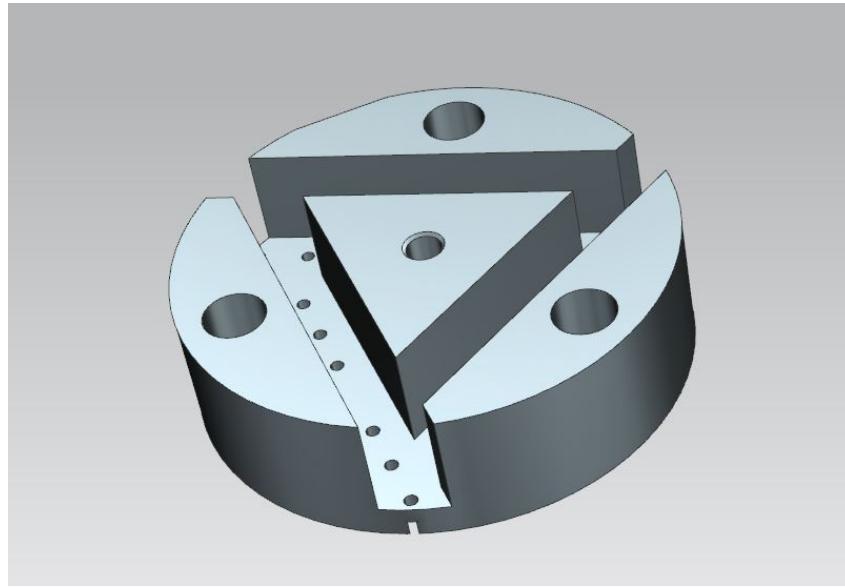


FIGURE 3.5: Design of the main mass

### 3.3.2 Counter mass

The counter mass should be approximately  $\frac{1}{15}$  of the main mass, which would be about 0.1kg. By looking at equation 2.74 we can see that a low mass is needed to keep resonant frequency of the actuator high. As mentioned in section 2.6 the resonant frequency of the actuators with the counter mass is calculated below, with the assumption that the mass is evenly distributed between the three actuators. The resonant frequency of the actuator is over three times the maximal driving frequency and therefore considered to meet the requirements mentioned in section 2.6.

$$f_0' = 1000 \sqrt{\frac{\frac{0.07}{3}}{\frac{0.07}{3} + 0.033}} = 642\text{Hz} \quad (3.3)$$

The counter mass was made from excess material. The dimensions of the bolts available were the same as the bolts described for the main mass. The outer dimension was machined down to a diameter of 95mm. A hole with a diameter of 8.5mm in the center of the counter mass was made so that the M8 bolt tightening the spring holding the counter

mass in place could slide freely. The thickness of the top mass was calculated considering using steel and aluminum and was calculated from equation 3.4.

$$t = \frac{m}{\rho\pi(r_o^2 - r_i^2)} \quad (3.4)$$

Where t is the thickness, m is the mass,  $\rho$  is the material density,  $r_o$  is the outer radius and  $r_i$  is the inner radius. All in SI units.

TABLE 3.3: Values used for calculating thickness of counter mass

Desired mass	Inner radius	Outer raduis	Density steel	Density aluminium
0.1 kg	0.00425 m	0.0475 m	8000 kg/m <sup>3</sup>	2700 kg/m <sup>3</sup>

If the counter mass was made of steel disc would have a thickness of 1.8mm as shown in equation 3.5.

$$\frac{0.1}{8000\pi(0.0475^2 - 0.00425^2)} = 1.8mm \quad (3.5)$$

The thickness of the aluminum plate would be 5.3mm as shown in equation 3.6.

$$\frac{0.1}{2700\pi(0.0475^2 - 0.00425^2)} = 5.3mm \quad (3.6)$$

The ADXL326 accelerometer is to be mounted on the counter mass. A threaded M3 hole through the thickness of the counter mass is used to mount the accelerometer. The threaded hole should be at least 0.8 to 1.5 times the diameter of the screw[29]. The minimum length of the threaded hole is therefor  $0.8*3mm=2.6mm$ . This means that the minimum thickness of the plate is 2.6mm and therefore aluminum is chosen as the material for the counter mass.

### 3.3.3 Rubber sandwich

The rubber cylinder for the test bench can be made in different sizes and thicknesses. The rubber is made at Teeness by vulcanizing the rubber in a sandwich structure with rubber in the middle and metal discs on top and bottom. The metal discs have to fit the molds used for the vulcanising process. Therefore the dimensions of the rubber sandwich was restricted by the dimensions of the molds. One key factor of the design is that the rubber sandwich can easily be changed and replaced with another sandwich. The physical properties as the natural frequency and damping ratio of the test bench

can thereby be changed when substituting the rubber sandwich with a rubber sandwich with other properties. This enables the test bench to be used for investigating different dynamic systems. Table 3.4 shows the properties of the chosen mold. The possible thicknesses produced of the rubber varies with an increment of 1mm.

TABLE 3.4: Cross section area dimensions for rubber

Outer diameter	Inner diameter	Area	Thickness
95 mm	33 mm	6233 mm <sup>2</sup>	2-24 mm

table 3.4 shows different material properties of rubbers which can be produced at Teeness. The relationship between the shear modulus and Young's modulus is given by equation 2.9 and the value of  $\nu$  is assumed to be 0.49, which is common for rubbers[23].

TABLE 3.5: Material properties for different rubbers

Name of rubber	G-modulus	E-modulus
EPDM 25	0.39 MPa	1.16 MPa
EPDM 40	0.70 MPa	2.09 MPa
EPDM 50	1.15 MPa	3.43 MPa
EPDM 70	5.00 MPa	14.90 MPa

The eigenfrequency of the test bench was initially dimensioned to be 100Hz. To obtain this eigenfrequency, equation 2.65 was solved via the "goal seek" function in Excel. The  $m_1$  mass used in the equation was 1.6kg. This value was determined by measuring the mass of the virtual representation of the main mass in NX, while material properties of steel from the material library were applied. The mass of the counter mass,  $m_2$  was measured the same way as  $m_1$  while applying material properties of aluminum and found to be 0.1kg.  $k_2$  in the equation represents the stiffness of the actuators in the displacement direction. The stiffness for one actuator was calculated by using Hooke's law, using data from the data sheet presented in table 3.1.

$$k = \frac{F}{\delta} = \frac{100N}{420e10^{-6}m} = 238095 \frac{N}{m} \quad (3.7)$$

Since three actuators were used for controlling the motion of the counter mass, the stiffness is three times the stiffness in equation 3.7, which leads to  $k_2$  is  $714285 \frac{N}{m}$ .

The value for  $k_1$  that gave resonance at 100Hz for the system was  $k_1 = 670650 \frac{N}{m}$ . When using the dimensions given in table 3.4 and the mechanical properties of EPDM 40 in

table 3.5 in a rearranged form of equation 2.7, be obtain the desirable length of the rubber profile to be 19.4mm.

$$L = \frac{AE}{k} = \frac{0.006233m^2 2090000 \frac{N}{m^2}}{670650 \frac{N}{m}} = 0.0194m = 19.4mm \quad (3.8)$$

The rubbers produced at Teeness have a minimum thickness of 2mm with a thickness increment of 1mm. Therefor rounding 19.4 to the closest whole number, we get 19 which will be the length of the rubber in mm. This will lead to a slightly higher stiffness in the rubber compared to its calculated length of 19.4mm. This will result in a slightly higher natural frequency, which can bee seen from equation 2.65.

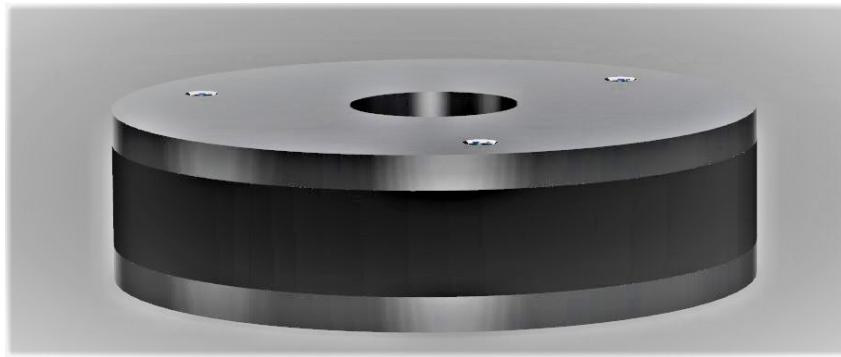


FIGURE 3.6: Design of the rubber sandwich

While assuming that the shaker, which is the supporting structure of the test bench, oscillates at the test bench's natural frequency with a constant acceleration, the acceleration of the main mass will be amplified. With the assumption of that the damping of the rubber is between 3-5% we can use equation 2.45 to see that the amplification ratio is between 10.0 and 16.7.

The disc on the top of the rubber has three threaded holes in which bolts going through the main mass is fastened with M5 bolts. The plate itself is only 5mm thick and therefore the length of threaded area is equal to the diameter of the screw, which is almost the minimum thickness recommended for the threaded length for screw connections [29]. The bottom disc of the rubber sandwich also has threaded holes so that it can be joined to the first coupling disk with M5 screws. The coupling between the shaker and the rubber sandwich is made out of two discs with a diameter of 120mm. These discs were connected to each other with M8 screws. The top plate has a M5 threaded hole to mount the reference accelerometer used by Vibration View to tune the shaker.

A key property of the rubber sandwich is that it is easily changed. If it is desired to change the stiffness or the damping ratio of the rubber one can simply change the rubber sandwich.

### 3.3.4 Spring

If the counter mass was placed on top of the actuators without being fastened to the rest of the test bench, the counter mass would be detached from the actuators when exceeding an lateral acceleration above 1g. Considering Newtons second law 2.16, the maximum force each of the actuator could apply before detachment would be given by the equation below 3.9.

$$F = mg = 9.81 \frac{m}{s^2} \frac{0.1kg}{3} = 0.33N \quad (3.9)$$

Therefore some modifications were needed to enable the actuators to use its full force. Two of the most intuitive ways of doing this is to either increase the weight of the counter mass or to add a force to the top mass to keep it in place. It was important to keep the resonant frequency well above the driving frequency of the piezoelectric actuator. Therefore the counter mass was designed with low mass when considering the effect it has on the resonant frequency given by equation 2.74. Therefore a spring was used to make sure that there was no separation between the counter mass and the sphere of the actuators. Gluing the counter mass to the spheres was considered, but was not done because it was a too permanent solution for a prototype, where adjustments of the test bench can be expected.

The spring applies a force compressing the counter mass to the actuator mounted on the main mass. A relatively soft spring is desired, so that the force remains fairly constant while the spring is being compressed by the actuators pushing the counter mass from the main mass.

The length of the threaded hole in the center of the main mass is at 26.2mm. The rubber sandwich has a inner diameter of 33mm, which allows the bolt to be screwed through the main mass down the center of the rubber sandwich. This results in that the bolt can be screwed  $26.2+19+2*5\text{mm}= 55.2\text{mm}$  from the top to the main mass before hitting the next body. A M8 screw would then need to be fastened at least 6.4mm in the main mass

to satisfy the 0.8 times diameter criteria[29]. This leads to that the remaining 48.8mm can be used to adjust the compression of the spring as shown in figure 3.7.

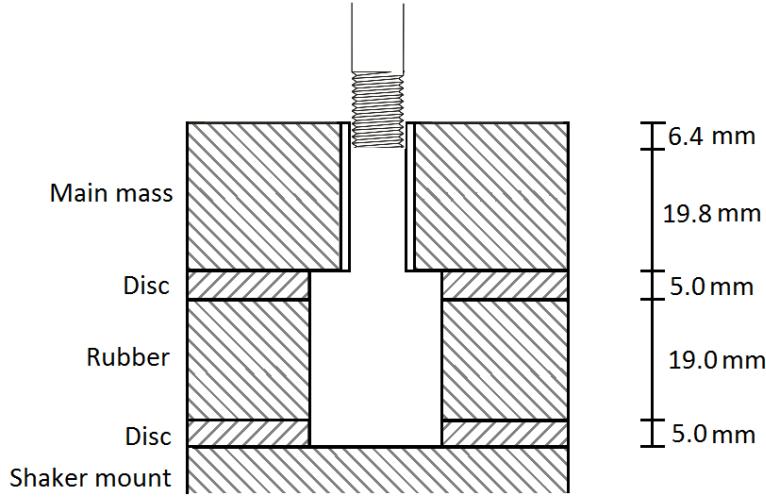


FIGURE 3.7: Illustration of distance the bolt can be screwed

The total blocking force of the three actuators combined is at 300N. A spring with stiffness  $6700\frac{N}{m}$  was used in the test bench to hold the counter mass in place. This would mean that the screw had to compress the spring 22.4mm to pretension the spring with half of the blocking force(150N). This is shown in equation 2.17.

$$x = \frac{F}{k} = \frac{150N}{6700\frac{N}{m}} = 0.0224m \quad (3.10)$$

The ISO standard for the pitch of the threads of an M8 are 1.25mm. This leads to that the bolt tightening the spring has to be turned 18 times to compress the spring with 22.4mm. This would also result in the force form the spring by tightening the bolt with one revolution would be 8.4N.

The reduction of the stroke is given by equation 2.72. Inserting values from 3.7 gives the new stroke length to be

$$L = 420\mu m \frac{238095}{238095 + 6700} = 409\mu m \quad (3.11)$$

This means that the difference in force by the spring when fully retracted compared to fully expanded is 2.73.

$$\Delta F = 409\mu m6700 \frac{N}{m} = 2.73N \quad (3.12)$$

The produced test bench mounted on the shaker is shown in figure 3.8.

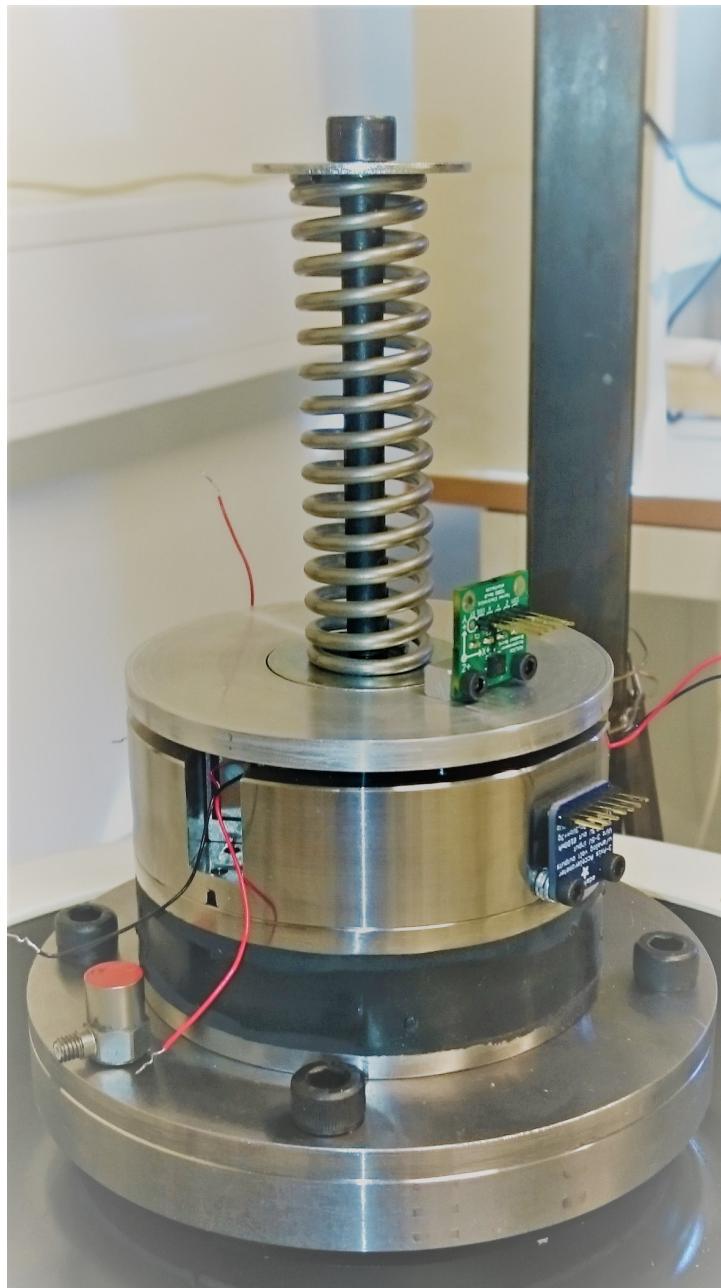


FIGURE 3.8: Physical test bench

### 3.4 The simulation setup

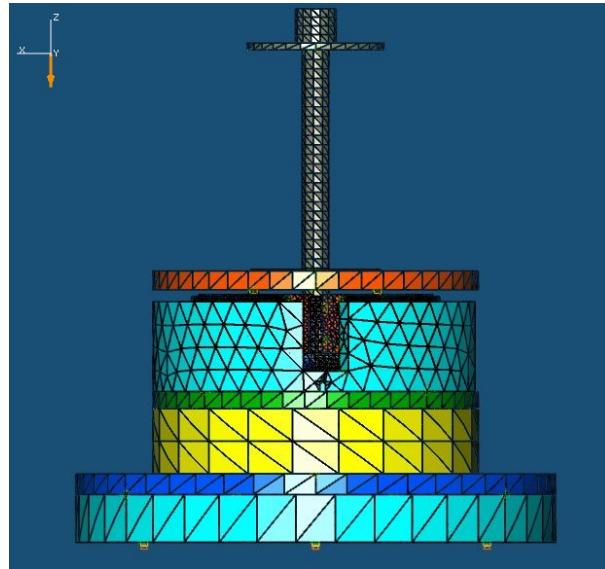


FIGURE 3.9: Components in the FEDEM simulation

The simulations were used to investigate the dynamic properties of the test bench and to investigate how the regulation of the actuators affect the reduction of the vibrations of the main mass. Comparing results from the simulations to the data sheets and results from physical test was done to improve the simulation representations of the virtual test bench.

The CAD parts used the simulations were the parts described in section 3.3. These were designed and assembled using NX. The parts were then meshed with CTETRA(10) elements, given material properties and nodes with connections to the part. All parts were given the mechanical properties of steel, with exception of the rubber(yellow in figure 3.9) and the aluminium counter mass(orange in figure 3.9). The key material properties for the three different material is shown in table 3.6. The nodes were added and connected to surfaces of the parts. This was done to connect the parts in the FEDEM model. A NASTRAN file for each part in the assembly was exported from NX and loaded into the model in FEDEM.

TABLE 3.6: Material properties used for the parts used in the simulations

Material	Density	E-modulus	$\nu$
Steel	7.83e-06 kg/mm <sup>3</sup>	206.94 GPa	0.29
Aluminium	2.71e-06 kg/mm <sup>3</sup>	68.98 GPa	0.33
Rubber	1.20e-06 kg/mm <sup>3</sup>	1.23 MPa	0.49

FEDEM(Finite Element Dynamics in Elastic Mechanisms) is a MBS program designed for solving dynamic problems. Imported parts are treated as individual objects and the relationship between these object are determined by interactions between triads set by the user. Triads are connected to a node on a part and some connections between triads demand that the triads have the exact same coordinates. The interaction between the triads can be determined by either joints, gears, springs or dampers.

A virtual spring was placed between the top of the bolt(grey in figure 3.9) and was given the stiffness of  $6700 \frac{N}{m}$ . In the physical test bench the spring is compressed by screwing the bolt with the spring between the washer and the counter mass. This was modeled in FEDEM by setting of initial stress free length to be longer than the initial length of the spring in the model. This would be the same as defining the original length of the spring to be the stress free length while the length of the spring in the model represents the length of the compressed spring. The added stress free length to preload the spring was calculated to be 0.0224m in equation 3.10.

The steel parts in the model were connected by RBE2 connections in a rigid joint in FEDEM. Rigid joints are joints between two triads at the same coordinate. The joint have all the relative translational- and rotational DOFs between the triads fixed, hence a rigid connection between the two triads.

To represent the motion of the shaker, free joints were modeled to be at the bottom most plate in the numerical model. Free joints are joints between two triads where all 6 DOFs are initially free, but can be altered by the user. This choice allows the user to set the relationship between triads to be free, fixed, prescribed or by a spring-damper relationship. The motion of the shaker is modeled as a free joint where the only DOF that is not fixed is in the z-direction(orange arrow) shown in figure 3.9. This motion is determined by a mathematical formula describing the oscillation of the virtual shaker.

Virtual sensors were added to triads to generate graphs of displacements, and to acquire acceleration data of the simulation. A control editor in FEDEM has to be used when requesting output from the sensors. The editor has different components that can be used to create a control system. These components can be used to model properties of the regulation system, such as delays and restrictions given by time response.

Rayleigh damping is used to simulate the internal proportional damping. Both stiffness proportional and mass proportional damping was calculated for all parts using equations 2.67 and 2.68.

### 3.4.1 Rubber modeling

The rubber between the main mass and the shaker has a great influence on the dynamics of the system. The stiffness of the rubber affects the eigenfrequency of the test bench and is represented as  $k_2$  in equation 2.65. The damping of the rubber determines how much the amplification of the force the main mass has compared to the base. How the rubber deforms is also essential for the models used in the FEDEM simulations to represent the real test bench. Firstly simulations were conducted to investigate the stiffness added by the transverse constraints of the rubber body. Secondly the deformation mode of the rubber was investigated in FEDEM. RBE2 and RBE3 connections used to couple the center node of the rubber to the surface area gave different deformation modes. Thirdly the Young's modulus of the material in the FEDEM model was adjusted to represent the stiffness of the transverse constrained rubber body with the selected RBE connection.

#### Stiffness of the rubber sandwich

The rubber used in the test bench is vulcanized between two plates during production. When compressed or elongated along one axis it deforms by increasing or decreasing the cross section of the middle of the thickness of the rubber. Because the rubber is fixed in the x- and y-plane shown in figure 3.11, structural stiffness is added to the rubber. This means that the rubber 2.2 not only is dependent on the material stiffness shown in equation 2.7. The stiffness of the rubber body is larger than what equation 2.7 would yield, since the equation does not consider structural stiffens. Two simulations using NX were conducted to determine the added stiffness due to the transverse constraints of the rubber.

Two simulations to investigate the added structural stiffness of the rubber due to the transverse constraints were conducted. In the first simulation the rubber was fixed in the z-translation, but free to move in the x-y plane on the bottom and a load represented by pressure was added to the top. The pressure needed to deform the rubber 0.1mm was calculated by using equation 2.8 and inserting material properties from table 3.5 and using the cross section area from table 3.4. The equation with the values described above is shown in equation 3.13.

$$\frac{2.086 \frac{N}{mm^2} * 0.0001m}{0.019m} = 0.01098 \frac{N}{mm^2} \quad (3.13)$$

The displacement resulting from the pressure in this case was 0.1mm as expected from equation 3.13.

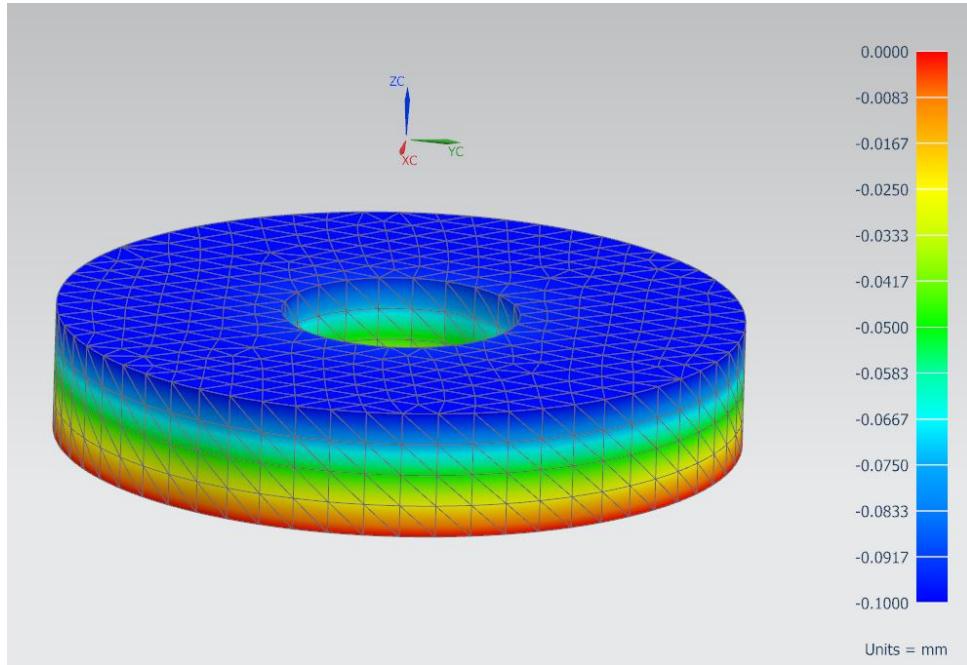


FIGURE 3.10: Deformations from simulation in NX, fixed bottom and free displacement in x-y plane on top

The second simulation was conducted while the bottom was fixed in all translations and the cross section at the top was fixed in the x-y plane. This was done so that the simulated rubber was constrained in a similar way like it would be in real life when it was vulcanised to the rigid plates. The deformations from this simulation are shown in figure 3.11.

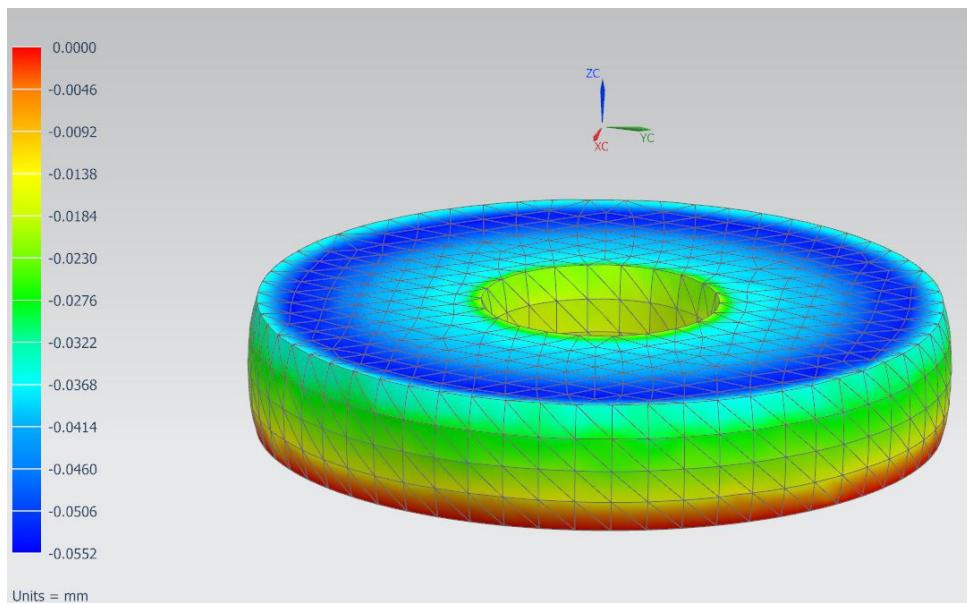


FIGURE 3.11: Deformations from simulation in NX, fixed bottom and fixed in x-y plane on top

The simulated displacement in z-direction here was 0.0552mm compared to the 0.1mm in the first simulation. This will imply that the stiffness of the body is almost doubled due to the added structural stiffness.

The stiffness of the rubber body was then calculated using the values found in the simulations. The total force on the body is 68.4N as shown in equation 3.14.

$$F = pA = 0.01098 \frac{N}{mm^2} * 6233mm^2 = 68.4N \quad (3.14)$$

The stiffness of the rubber body can then be calculated by

$$k = \frac{F}{\delta} = \frac{68.4N}{0.0000552m} = 1239130 \frac{N}{m} \quad (3.15)$$

This compared to the originally calculated stiffness of  $670650 \frac{N}{m}$ . This would result in a natural frequency of 136Hz by using equation 2.65. The stiffness found in equation 3.15 was further used to adjust the Young's modulus of the virtual rubber part exported to the FEDEM simulation.

### **Deformation mode of the rubber sandwich**

Eigenmode simulation in FEDEM were conducted to determine whether to use RBE2 or RBE3 connections in the FEDEM simulations to connect nodes to the cross section surfaces. The goal for these simulations were to see how the rubber deformed when being subjected to oscillations at its natural frequency. The eigenmode describing the deformation of the rubber when RBE2 connections were used gave a geometrical deformation of the rubber similar to the geometrical deformation in the NX simulation.

The deformations in the eigenmode analysis shown in figure 3.13 show a twisting motion that occurs when using RBE3 connections. This is not a likely behaviour when the the only force is generated by the oscillations of the shaker. RBE2 connections will be used for connecting the rubber as a result for the eigenmode analysis.

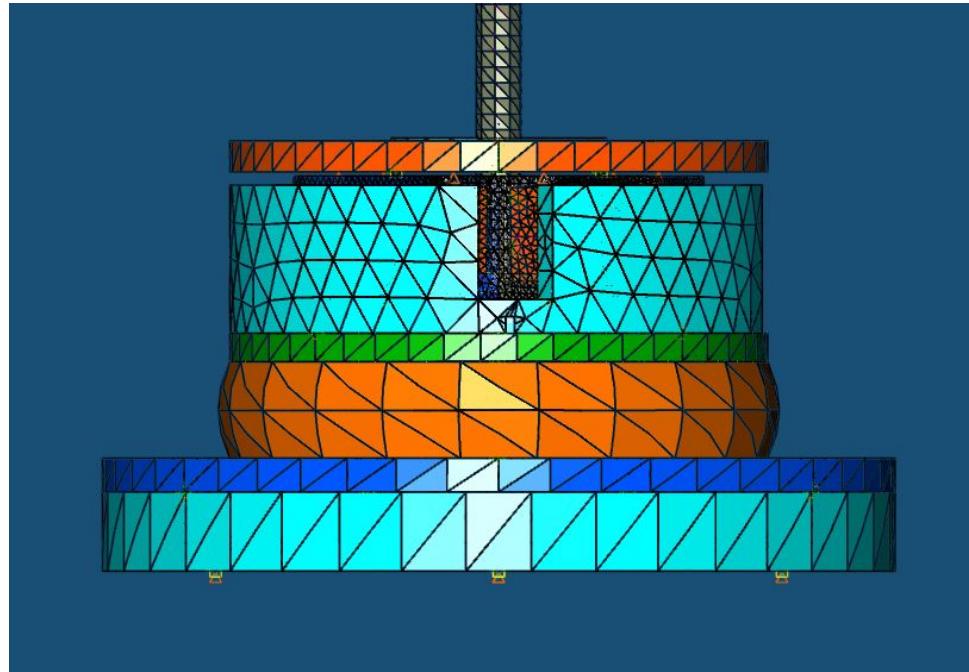


FIGURE 3.12: Deformations in eigenmode analysis with RBE2 elements

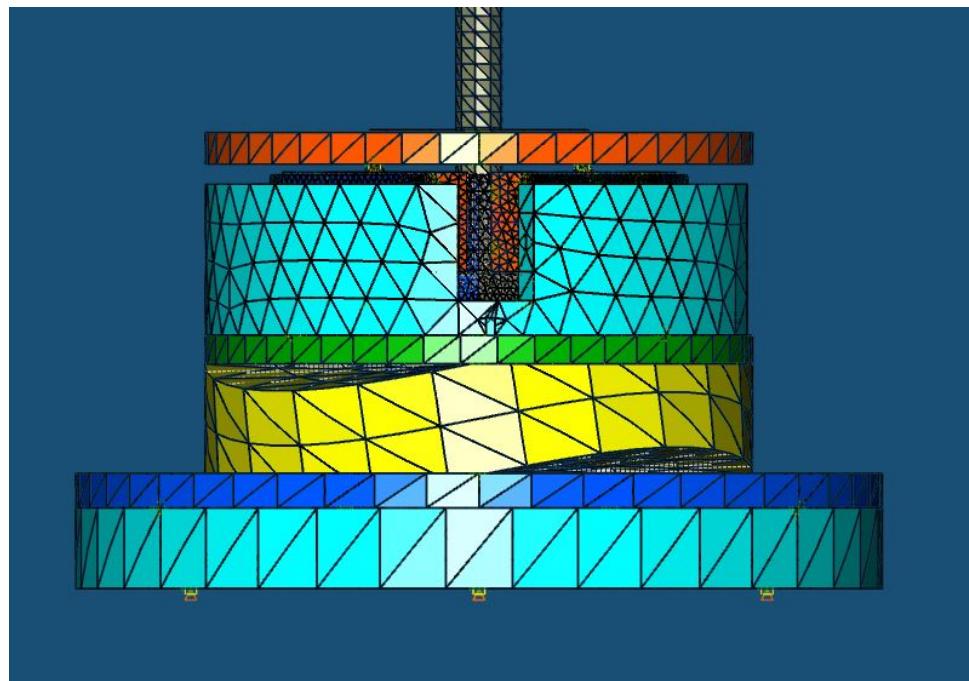


FIGURE 3.13: Deformations in eigenmode analysis with RBE3 elements

### Material properties change of rubber

Because the RBE2 connection affect the structural stiffness of the rubber, simulations were conducted to investigate the stiffness of the rubber in FEDEM. A center point in the center of the rubber cylinder was made on both the top and bottom plane of the rubber. These points were connected with RBE2 connections to the cross section surface

of the rubber cylinder. The bottom point was held fixed while a force of 1239N was applied perpendicular to the cross section surface. The deformation of the rubber would be 1mm when the stiffness of the rubber is  $1239130 \frac{N}{m}$  as calculated in equation 3.15. The deformation was measured as the relative displacement between the top- and bottom node mentioned above. The Young's modulus of the rubber body was changed until the simulated deformation of the rubber was 1mm. This was done so that the stiffness of the rubber in the FEDEM model was the same as the stiffness as found in the NX simulations. The stiffness was reduced from 2.086MPa to 1.231MPa to obtain the desired body stiffness of the rubber. Figure 3.14 show the deformation in FEDEM with its original Young's modulus in blue and the deformation with the adjusted Young's modulus.

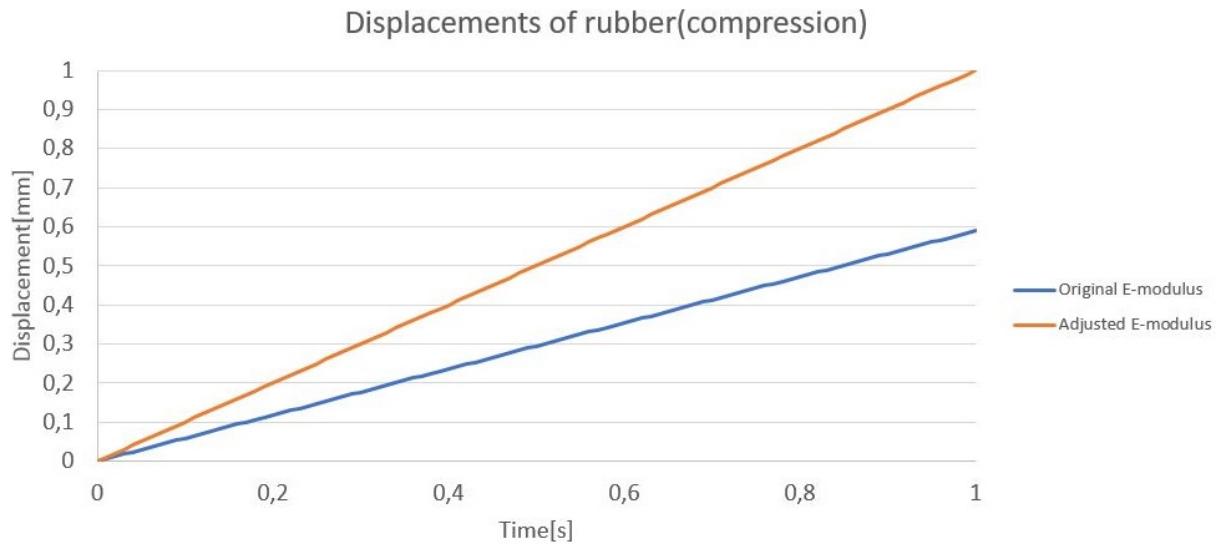


FIGURE 3.14: Compression of rubber.

### 3.4.2 The actuator

Initially the model for the actuator was an imported STEP file from Thorlabs web-page [3] to get the correct geometry and dimensions for the flexure and the cover. Some minor changes were made in the simulations, such as uniting the ball on the top to the midst part of the flexure by reducing the whole made for the ball. This was done to be able to treat the actuator as one part. Initially, the ball in the STEP file was floating loose in the model with no contact to the flexure. The piezoelectric stack was removed from the model to enable a force/displacements to be applied to the flexure without implications form the stiffness of the designed stack.

When the actuator was simulated as one part the geometrical representation of the part are kept constant during the simulation. This resulted in a linear relationship between

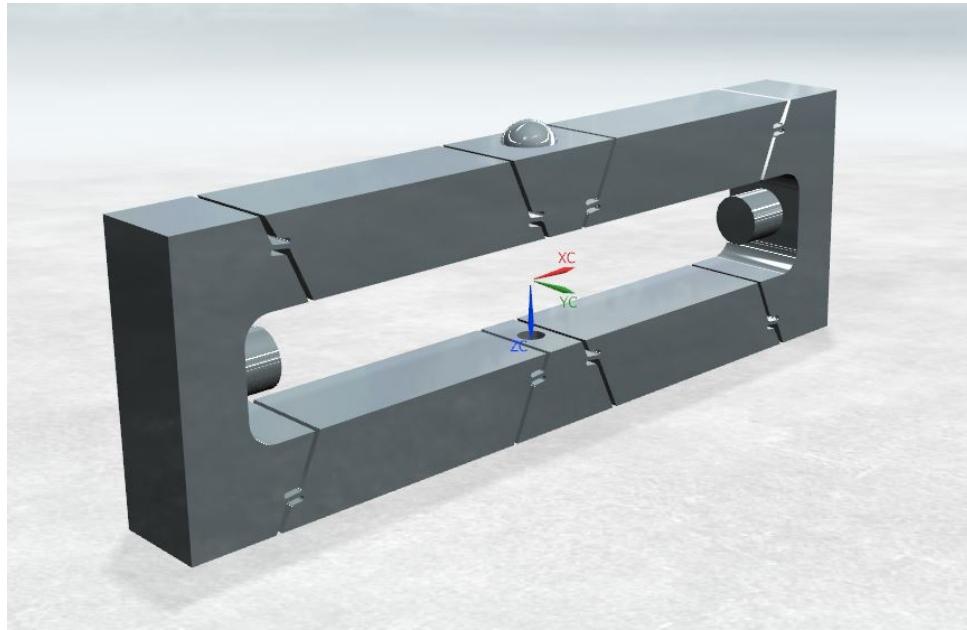


FIGURE 3.15: Render of the flexure

the simulated displacement of the stack and the sphere. The geometrical changes that occur during the displacements of the actuator are not included when the actuator is modeled as one part. This error can be eliminated if the flexure is divided into parts. Due to this the flexure was divided in to 8 parts and used for all further simulations. Figure 3.16 shows the difference of the displacement for the top sphere on the flexure when it is modelled as one part and eight parts.

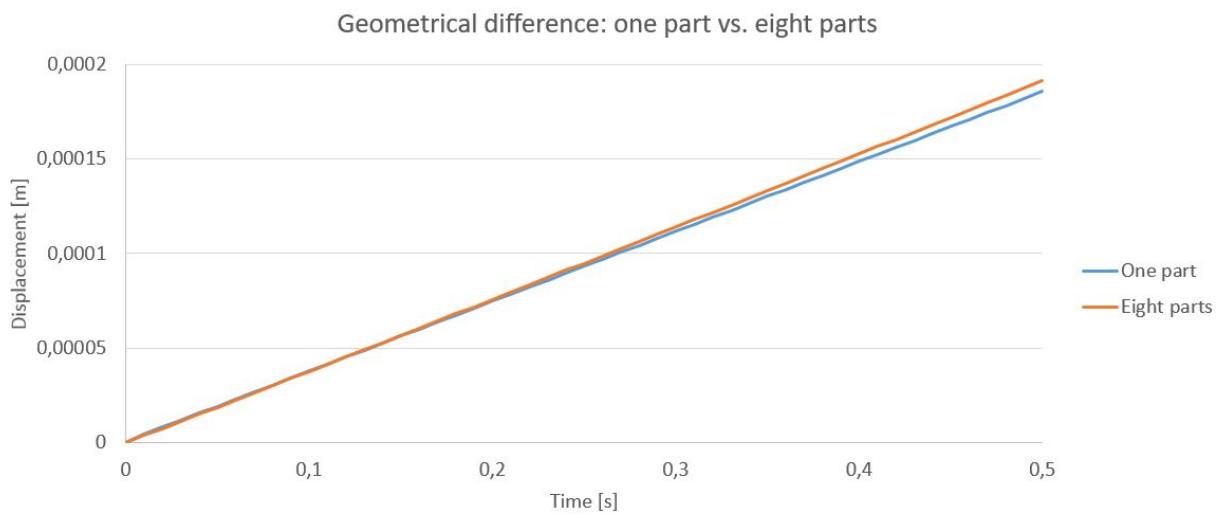


FIGURE 3.16: Comparing displacements of the sphere. Flexure modelled as one part compared with eight parts

In NX nodes were connected to the parts by RBE2 or RBE3 connections to the surface of the cross section where the flexure was divided. The model was initially made with

RBE3 connections, but errors in FEDEM arose due to negative pivots in matrices caused by the RBE3 connections. These errors were removed when the RBE3 connections were substituted with RBE2 connections.

One node was placed in the center of the cross section of the cut surface where the flexure was divided. Rigid joints were made between the nodes and additional restrictions on the triads on the nodes were added. The additional restrictions allowed translation for the triads only in the z direction, as it would in real life applications. This was done to reduce computational errors and to avoid unwanted translations.

The material used in the flexure was not defined in the data sheet for the actuator. Therefore the predefined material "Steel" in NX was chosen, with the properties presented in table 3.6.

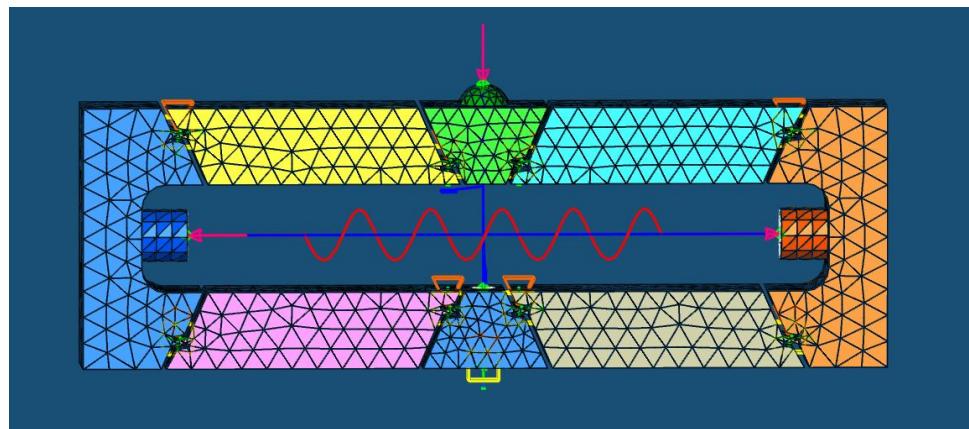


FIGURE 3.17: FEDEM model of the flexure in eight parts

The initial model of the stack and flexure was made by using the properties in the data sheet for the piezoelectric stack. The piezoelectric stack was modelled as a spring and a force exerted on the cylindrical end-cup and screw. The spring in the model had the stiffness of 22 321 429.0 N/m. This was calculated by equation 2.71 by entering the properties from the data sheet presented in table 3.1.

$$K_a = \frac{1000N}{44.8\mu m} * 10^6 \frac{\mu m}{m} = 22321429 \frac{N}{m} \quad (3.16)$$

Two simulations were conducted using the properties from the data sheet to the actuator and piezoelectric stack. The first simulation gradually applied the maximum force the stack could provide of 1000N and to the end cup. After the maximum force was applied to the end cup a force equal to the maximum blocking force of 100N was applied to the sphere compressing the actuator towards its original position.

The second simulation conducted gradually changed the stress free length of the spring with the stiffness given in equation 3.16 representing the piezoelectric stack. The displacement of the stack is set to be  $44.8\mu m$ . After the full displacement for the simulated piezoelectric stack was applied, the blocking force was gradually applied in the same way as in the first simulation.

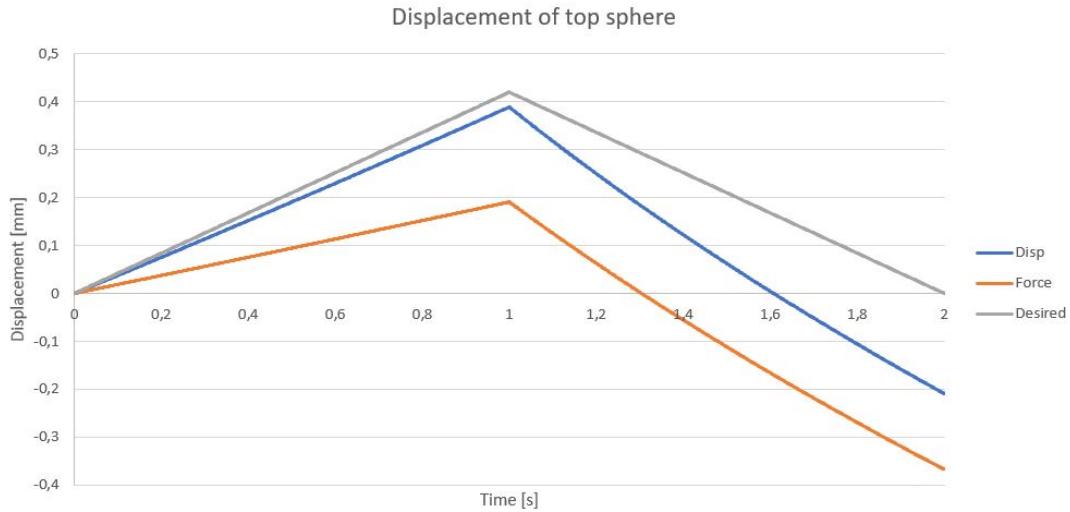


FIGURE 3.18: Displacement of sphere with force and displacement regulation using values from data sheet.

Figure 3.18 shows the simulated displacement of the sphere when conducting these two simulations and the desired results. The desired result shows a displacement of the actuator sphere of  $420 \mu m$  when the simulated stack is elongated. When the blocking force is applied the sphere returns to its original position, as it should according to the definition of the blocking force used in [2]. Since neither of the approaches shown in figure 3.18 described above gave the desired results, the properties for the spring stiffness and the displacement of the simulated piezoelectric stack were changed. The desired properties for actuator stiffness and actuator displacement in the model were the dimensioning criteria, the stiffness and the displacement of the modeled piezoelectric stack were changed to achieve this. The parameter changes made are shown in table 3.7 .

TABLE 3.7: Deviation in properties used in parts in simulation compared with data sheet

Parameter	Value in data sheet	Value in model
Displacement of Piezoelectric stack	$44.8\mu m$	$46.7\mu m$
Stiffness of piezoelectric stack	$22321429 \frac{N}{m}$	$66800000 \frac{N}{m}$

The relations of the stack displacement and the sphere displacement can be approximated to be linear as shown in figure 3.16. By using the stack displacement value shown in table 3.7 and remembering that the maximum displacement of the actuator is simulated to be  $420\mu m$  we can obtain the amplification factor,  $A_f$  between the stack displacement and the sphere displacement.

$$A_f = \frac{420\mu m}{46.7\mu m} = 8.99 \quad (3.17)$$

# Chapter 4

## Tests, calibrations and scripts

Several tests and were conducted to investigate how the coupling between the Arduino and the accelerometer should be, how the Arduino should be programed, and how the simulations should be adjusted to have dynamic properties that are similar to the physical test bench. A list over the tests and calibrating measures are listed below.

1. Noise in the circuit was measured with different hook up configurations between the Arduino and other components to find the best configuration in section [4.1](#).
2. The time different commands in the scrips used were investigated to point out what commands to be avoided in the scripts in section [4.2](#).
3. Data from the accelerometers were investigated to find the relations between the measured voltage and acceleration in section [4.3](#).
4. Scripts were developed and physical relations were investigated to be able to make a suitable way regulate the actuators in section [4.4](#).
5. Data from physical tests of the test bench using Vibration View and the shaker provided by Teeness were conducted to adjust the properties in the simulations in section [4.5](#).

## 4.1 Noise in the circuit

First the electrical circuit was set up. The accelerometers were connected to the Arduino and the Arduino was connected to the amplifier. Noise in different connections between the accelerometer and the Arduino investigated in three different configurations. Accelerometer data was read by the Arduino while the accelerometer was completely stationary. To investigate the noise in the circuits from the Arduino to the actuator, signals from the Arduino were sent via the amplifier and back, where the noise was measured.

### 4.1.1 Noise between accelerometer and Arduino

Two different ADXL335 accelerometers and two different ADXL326 accelerometers were tested and compared in different configurations. The accelerometers were fixed and during 10 seconds the Arduino read values from the accelerometer and printed the values via the serial port. These variation of these values were then used to compare the different connections of the accelerometers to the Arduino to choose the desired configuration.

#### Configuration 1

Wires from a Cat 5 cable were directly soldered on the accelerometer and the other end in a 8P8C- male contact. The male end was then connected via a female end of the same type of contact and placed into the analogue pins of the Arduino. This was thought to be a good choice of connecting the sensors to the Arduino since the cables were designed for transferring signals.

#### Configuration 2

Pins were soldered onto the accelerometer and the jumper-wires that followed the Arduino kit were cut in two. A wire was soldered between the two jumper-wire halves to enable for the Arduino to be further from the shaker. This was done so the Arduino could be placed on a table near by and fixture to hold the Arduino is not necessary.

#### Configuration 3

Pins were soldered onto the accelerometer and jumper-wires that followed the Arduino kit were used between the pins soldered on the accelerometer and the Arduino. These wires are 70mm long and therefor the Arduino had to be close to the test bench when it was used. A fixture had to be made to hold the Arduino when the wires are shorter than 500mm, due to the size of the shaker and the placing of the work bench.

## Comparing configurations

Key results from the noise test are given in the tables 4.1 and 4.2 shown below.

TABLE 4.1: Data from noise test of the ADXL326 accelerometer

ADXL 326			
	Configuration 1	Configuration 2	Configuration 3
Maximum value	2022	2038	2042
Minimum value	1994	2024	2038
Maximum deviation	28	14	4
Average value	2003	2031	2040
Standard deviation	2.94	1.52	0.92

TABLE 4.2: Data from noise test of the ADXL335 accelerometer

ADXL 335			
	Configuration 1	Configuration 2	Configuration 3
Maximum value	2378	2370	2365
Minimum value	2312	2346	2360
Maximum deviation	32	12	5
Average value	2324	2358	2363
Standard deviation	3.25	1.5	1.00

These results demonstrate that configuration 3 gives the least noise from the measurements. A mount for the Arduino was made to enable the use of the short jumper-wires and this configuration was used throughout this paper.

### 4.1.2 Noise in the rest of the circuit

The actuators are regulated by the Arduino via an amplifier. The wires both form the Arduino to the amplifier and the wires from the amplifier to the actuator are coax cables. These cables were used in the setup for measuring noise in the amplifier circuit. Signals were sent from the DAC pins on the Arduino to the amplifier. The signals were then amplified with a factor of one and no offset(input signal=output signal) and read by the A0 pin of the Arduino. The signals sent from the DAC pin were compared to the signals read in the A0 pin. The test showed that no significant noise could be detected.

A second test where the output signal from the Arduino was reduced to half compared with the first test and setting the amplify setting to 2, the same output signal was measured and without significant noise.

## 4.2 Time used by functions in script

The speed of the code is important to ensure rapid output signals to generate smooth signals at high frequencies. Therefore a script was run on the Arduino Due to measure the time each instruction used in the script. This is done so that the speed of the scripts can be improved and to investigate if there are some instructions in the code that should be avoided due to time consumption. The script used to measure the time is shown in figure 4.1.

```

1 int y=0;
2 int Tid1=0;
3 int Tid2=0;
4
5 void setup(){
6     Serial.begin(115200);
7     Tid1=micros();
8 }
9
10 void loop(){
11     while(Tid2<1000000){
12         Tid2= micros()-Tid1;
13         // Insert code to be timed here
14         y=y+1;
15     }
16     Serial.println(y);
17 }
18

```

FIGURE 4.1: Script for measuring the time consumption for code instructions

The script outputs the number of rounds the loop can run within one second. The output of the code was then used to calculate the time spent by each code by using equation 4.1.

$$\text{Time used by instruction in } \mu\text{s} = \frac{1000000\mu\text{s}}{\text{number of rounds}} \quad (4.1)$$

The time used for some investigated codes are presented in table 4.3. When the loop ran with no other function in it(NONE), the code spent  $2\mu\text{s}$ . The "Time added pr cycle" shows the time added as a result of including the function.

From this we can see that the `Serial.print()` command uses  $83\mu\text{s}$  pr character sent via the serial communication. The `analogRead()`-values with a 12 bit resolution consist of up to 4 characters leading to  $83*4\mu\text{s}(332\mu\text{s})$  spent for printing one output vale. For this reason

TABLE 4.3: Time in  $\mu\text{s}$  used by code instructions

Function	Cycles	Time pr cycle	Time added pr cycle
NONE	508500	2	0
Serial.print(pr character)	11797	85	83
if	508500	2	0
while	485000	2	0
analogRead	161328	6	4
analogWrite	182396	5	4
digitalWrite	283401	4	2
sin	238334	4	2
sin(float)	19564	51	49
$\pm$	511445	2	0
$\pm$ (float)	502001	2	0
$*$ /	511445	2	0
$*$ / (float)	264667	4	2

the Serial.print() command should be avoided when possible. Using float numbers in the sine function also reduces the speed of the script and should be avoided.

## 4.3 Accelerometer

The accelerometers used are the ADXL335 and the ADXL326. The ADXL335 accelerometer is mounted on the main mass and the data from this accelerometer is used by the Arduino to regulate the actuators. This makes the data from the ADXL335 imperative for the regulation system. The ADXL326 accelerometer on the other hand is used to measure accelerations of the counter mass and not used for the regulation of the actuators. Therefor the acceleration data from the main mass was deemed more important than the acceleration of the counter mass. Therefor the focus of the testing was on the ADXL335 accelerometer.

A series of tests were conducted to investigate the performance of the ADXL335 accelerometer. Tests with frequencies ranging from 10Hz-150Hz and with 1, 2 and 3 times the acceleration of the gravity was applied to the accelerometer. A list of the tested parameter combinations is shown in table 4.4. The accelerations and the frequencies were input in Vibration View and the Arduino compatible accelerometer was mounted on the shaker.

TABLE 4.4: Tests conducted to investigate accelerometer data precision

<b>Acceleration</b>	<b>Frequency</b>
1g	10 Hz
1g	50 Hz
1g	100 Hz
1g	150 Hz
2g	10 Hz
2g	50 Hz
2g	100 Hz
2g	150 Hz
3g	10 Hz
3g	50 Hz
3g	100 Hz
3g	150 Hz

The raw data was logged for 10 seconds while the shaker applied the acceleration at the given frequency. The test at 10Hz giving an acceleration equal to 3 times of the gravity, could not be preformed because of a restriction of the translation length of the shaker.

The comparison of the values of the accelerometer output while varying the frequency and acceleration was set to 1g is shown in figure 4.2. The graph shows that there is

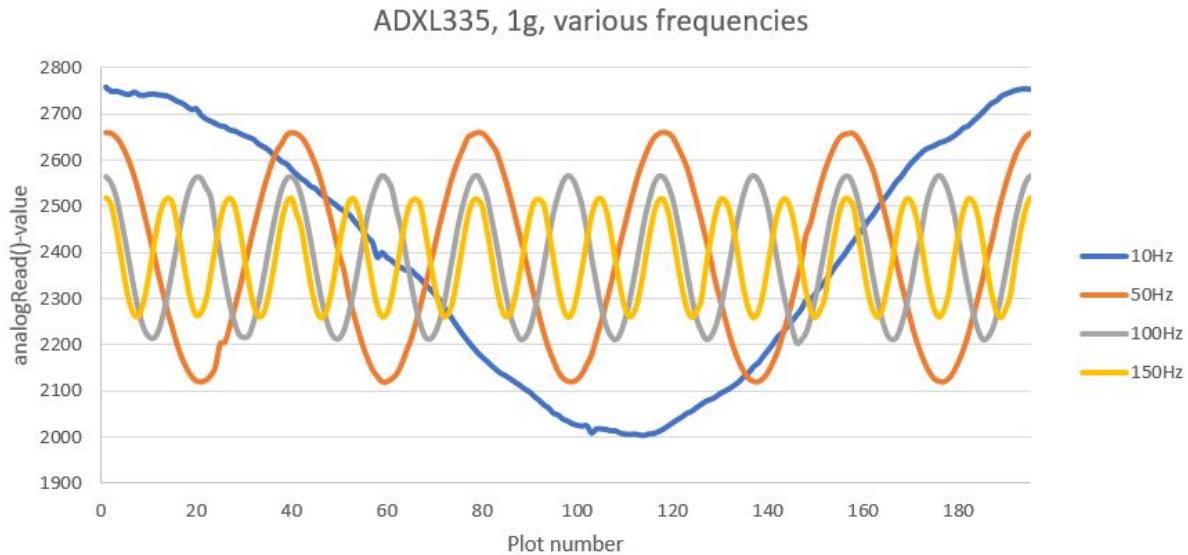


FIGURE 4.2: Signal from ADXL335 at 1g acceleration at various frequencies

an unexpected deviation of the amplitude of the analogRead()-value. The amplitude of the analogRead()-values should theoretically be the same magnitude independent of the frequency.

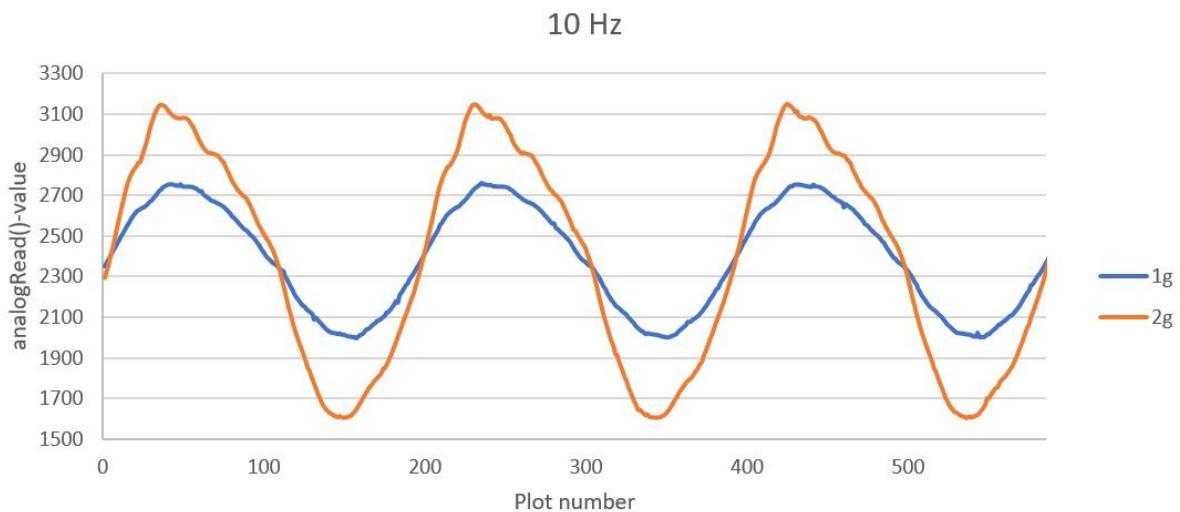


FIGURE 4.3: Signal from ADXL335 while mounted on shaker with 10Hz frequency and varying acceleration

Figures 4.3, 4.4, 4.5 and 4.6 show the analogRead()-values compared to the plot number of the series. The time increment between each plot is the same for every plot and should be about  $332 \mu s$  between each point. By looking at the amplitudes of these graphs we can see that the amplitude is fairly proportional of the acceleration. This relationship is shown in 4.7. We can also see that the analogRead()-values do not make a smooth sine curve as intended at 10Hz. This is presumably because the shaker was not designed

to work at low frequencies and long travel ranges and therefore has trouble supplying a consistent acceleration. The data from 10Hz measurements are therefore considered to be unreliable.

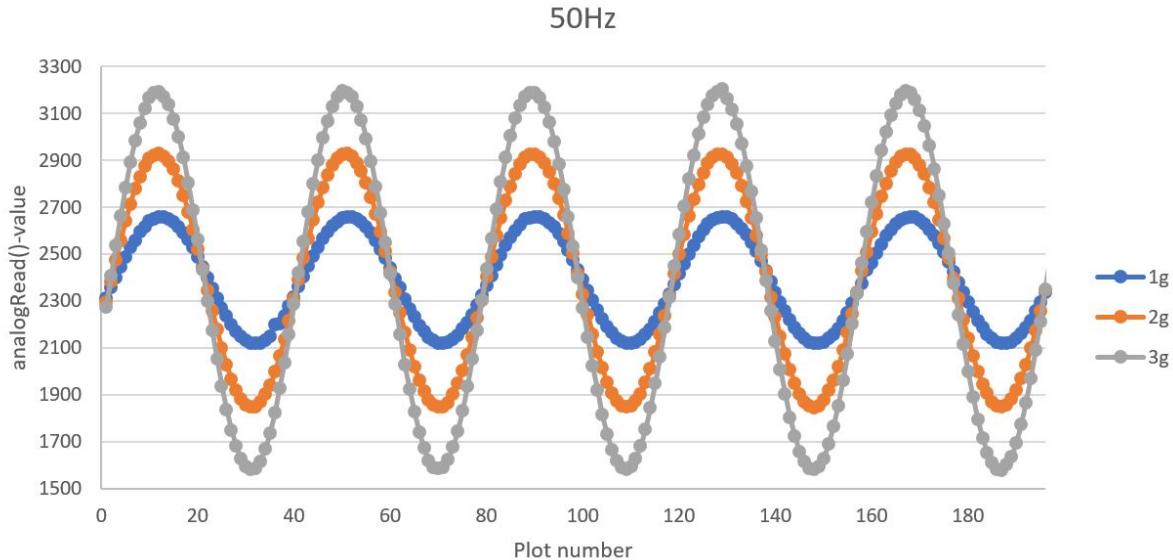


FIGURE 4.4: Signal from ADXL335 while mounted on shaker with 50Hz frequency and varying acceleration

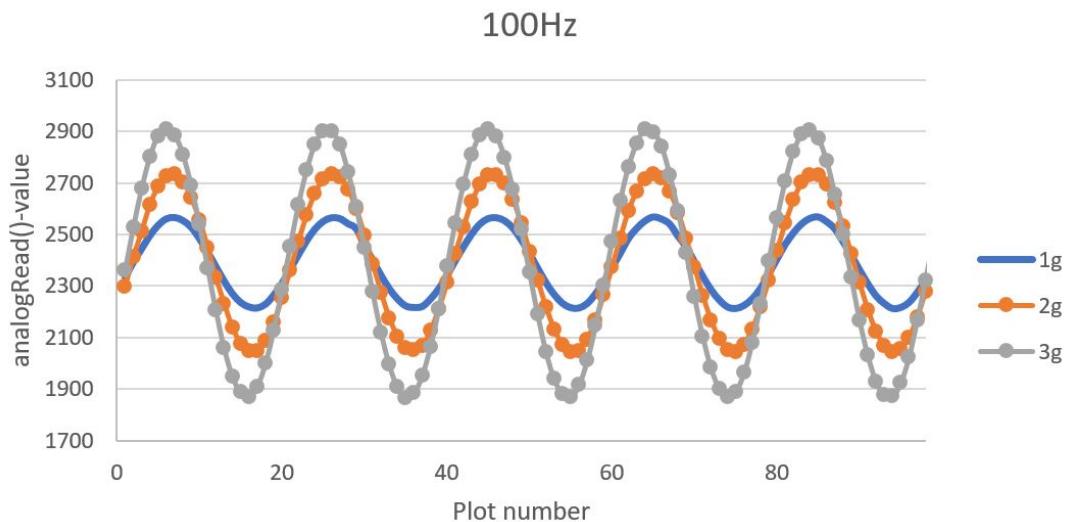


FIGURE 4.5: Signal from ADXL335 while mounted on shaker with 100Hz frequency and varying acceleration

The number of plots measured per period at the high frequencies are about 19 at 100Hz and 13 at 150Hz. The main restriction for the number of plots pr time interval is the serial communication between the Arduino and the computer as can bee seen from table 4.3. This will lead to that the Arduino can read values from the accelerometer much

faster than the data shown in figures 4.3, 4.4, 4.5 and 4.6 when serial communication is not used.

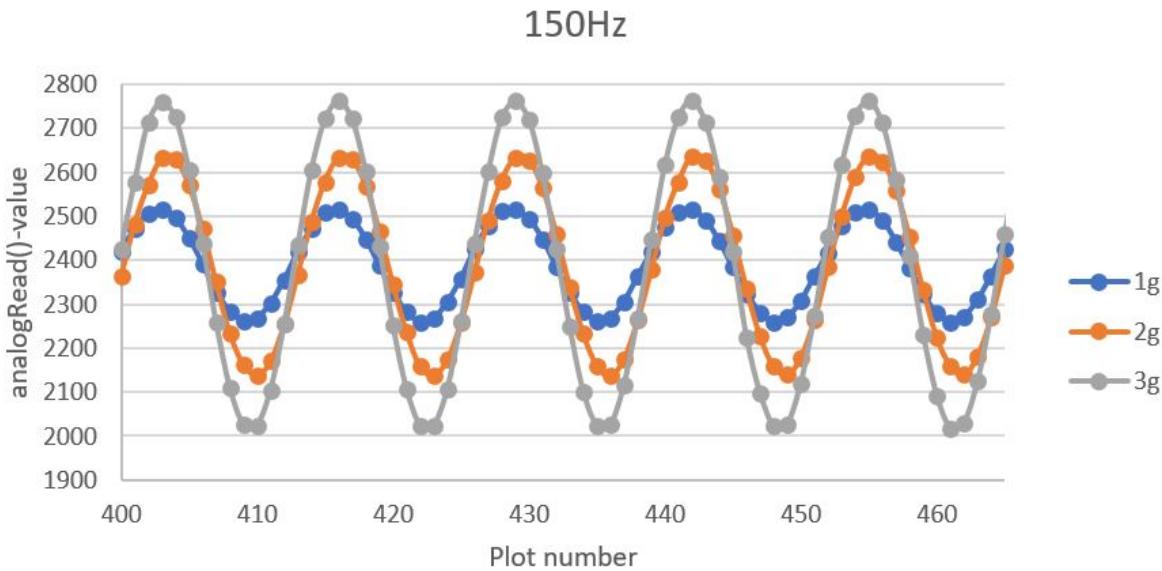


FIGURE 4.6: Signal from ADXL335 while mounted on shaker with 150Hz frequency and varying acceleration

Data obtained from the tests conducted in table 4.4 were analysed and used to describe the behaviour of the accelerometer. First the maximum and minimum value of each series were calculated by taking the average of the 10 highest and lowest values respectively. The average was found by averaging all values in the series. The peak amplitude,  $A_p$  was calculated by using equation 4.2.

$$A_p = \frac{(average - minimum) + (maximum - average)}{2} \quad (4.2)$$

The peak amplitude was then normalised with respect to the acceleration applied by the shaker. Plotting the normalised analogRead()-value vs the applied acceleration gave figure 4.7. The ideal accelerometer would have had a constant normalised amplitude independent of both magnitude of the acceleration and the frequency. This is not the case of the ADXL335 accelerometer used in this test.

The slope of the lines representing the different frequencies in figure 4.7 indicate that the measured average amplitude of the analogRead()-data from the accelerometer is dependent on the magnitude of the acceleration. Since the slope is negative, the measured normalised average amplitude is lower at high accelerations. The average change in

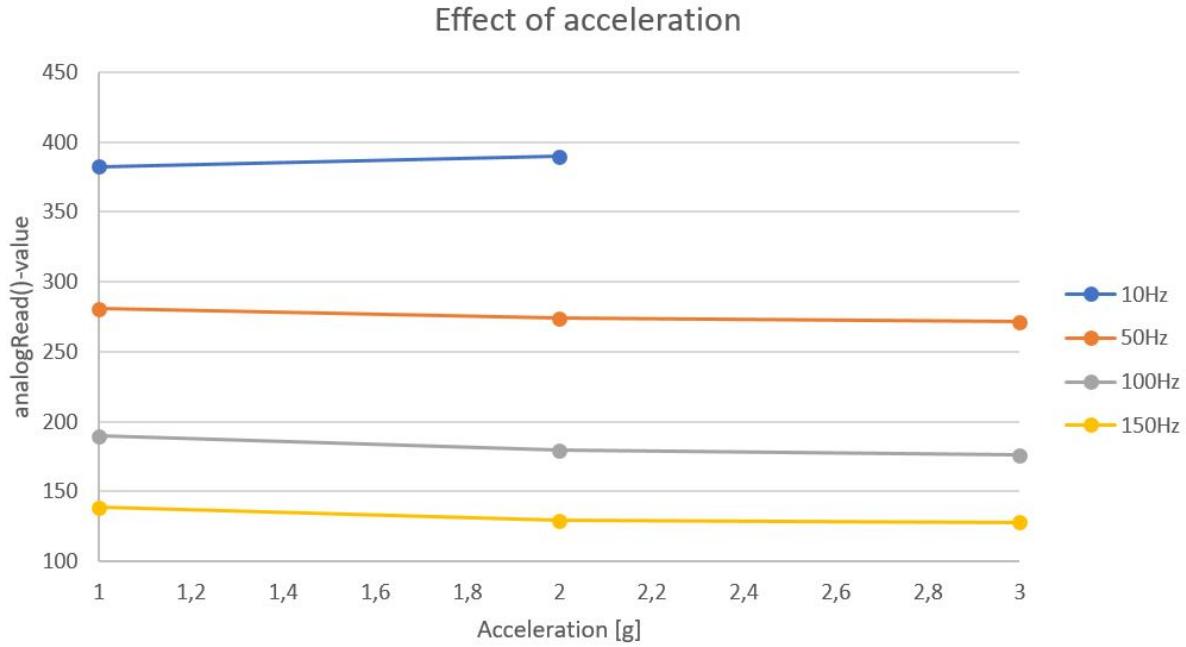


FIGURE 4.7: Relationship between acceleration and analog read value

percentage of the average normalised amplitude per 1g increase in applied acceleration of the accelerometer is given in the table 4.5.

TABLE 4.5: The percentage change of analogRead()-value due to increasing acceleration

Frequency	AnalogRead()-value change pr 1g increase
10	1.9%
50	-1.6%
100	-3.6%
150	-3.8%

The vertical shift of the lines in the graphs shown in figure 4.7 show that the amplitude is affected by the frequency of the motion. The amplitude of the analogRead()-values representing 1g acceleration is reduced to about 36% when increasing the frequency from 10Hz to 150Hz.

By plotting the normalised average amplitude value compared to the frequency we obtain the graph in figure 4.8. This shows that the measured amplitude is highly dependent on the frequency. Figure 4.8 is made by plotting the average analogRead()-amplitude representing 1g acceleration against the frequencies applied. This was done for the three different acceleration magnitudes applied. A curve was made representing the average values for the three different curves shown in fig 4.8. This curve was used to generate a

polynomial function in Excel to describe the relationship between the applied frequencies and the analogRead()-amplitude representing 1g acceleration.

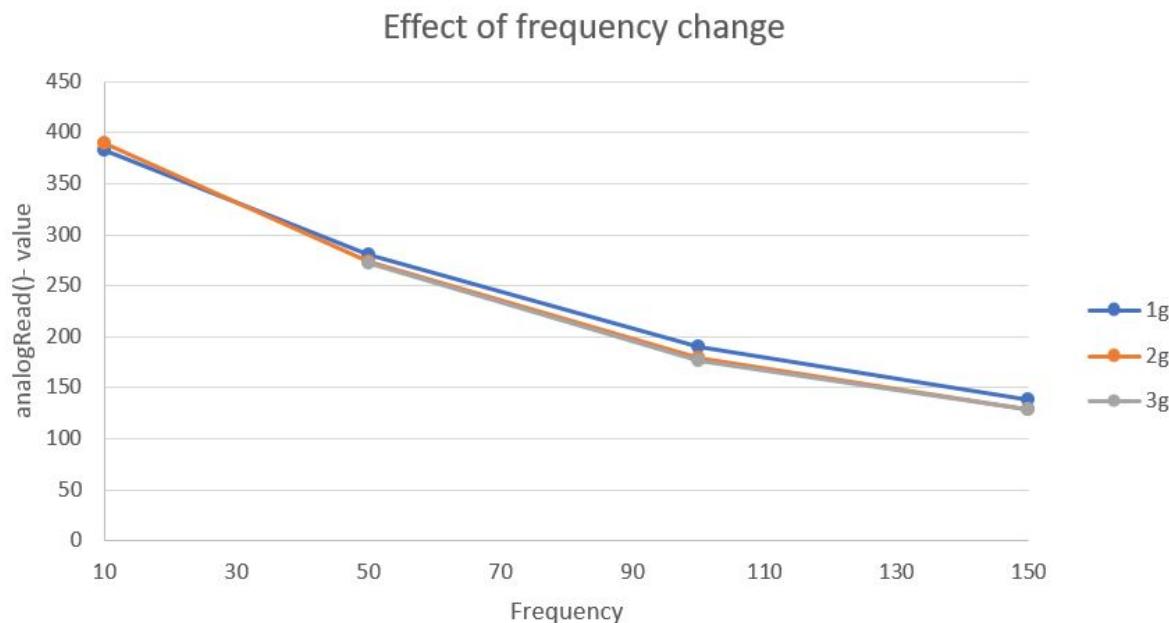


FIGURE 4.8: Relationship between analogRead()-value amplitude and frequency

The polynomial expression describing the relationship is shown in equation 4.3 where  $y$  is the acceleration measured in a factor of the gravity acceleration and  $x$  represents the amplitude of the analogRead()-values.

$$y = 0.01x^2 - 3.443x + 420.78 \quad (4.3)$$

The R-square value for equation 4.3 was calculated for all three curves and is shown in table 4.6. The R-squared value describes how much of the variation of the plot that can be described by the function. Since the amplitude is dependent on the frequency, the relationship between measured accelerometer data and output voltage regulating the actuators gets more complex. The relationship between frequency and amplitude must be implemented to the code to get a relationship between the analogRead()-values and the applied acceleration.

Assuming that equation 4.3 holds from 0 to 200Hz, the minimum value of the amplitude representing 1g acceleration would be found by deriving the equation and finding the frequency where the lowest amplitude occurs and then plotting the value of the frequency to represent x in equation 4.3. This results in the minimum amplitude of the analogRead()-values representing 1g to be 124. The minimum resolution for 1g using the script will then be 0.008g. The maximum noise detected form the analogRead()-value while reading data from the accelerometer was 5. This will then result in that the noise can distort the acceleration readings of up to 0.04g.

TABLE 4.6:  $R^2$  values for the different accelerations

Acceleration	$R^2$
1	0.997
2	0.999
3	0.996

## 4.4 Scripts used in the Arduino Due

The signals from the accelerometers had to be treated to create an output from the Arduino that regulates the actuators to dampen the vibrations. This was done by creating scripts that interpret the accelerometer data to determine an output signal.

The amplitude, minimum, maximum and average of the analogRead()-values had to be measured and calculated to further be used to calculate the frequency of the input signal. These values were put in equation 4.3 to obtain the estimation of the acceleration. The relationship between the output signal from the Arduino and the displacement of the actuator was calculated. The output signal algorithm has to be chosen and lastly a script to determine when to initiate the regulating signal in the desired phase was used.

The signal output from the Arduino is then sent via the amplifier to the actuator. It is therefor important that the signal from the Arduino is consistent and precise. Deviations from the Arduino will also be amplified and will affect the regulation of the actuator and therefor the damping properties of the active system.

### 4.4.1 Frequency and amplitude estimation

To be able to regulate the actuators with the correct signal, the data from the accelerometer had to be used to estimate the frequency and amplitude. The data has to be converted to determine the amplitude and frequency of the motion of the accelerometer to be able to generate the signal regulating the actuators. The Arduino must also be able to ensure that the motion of the counter mass has the desired phase compared to the main mass. The script used for obtaining the frequency and amplitude follows the following steps:

1. Obtain minimum and maximum analogReadd()-values from the accelerometer.
2. Calculate the average analogRead()-value.
3. Measure time when the analogRead()-value is below average.
4. Measure time when the analogRead()-value is above average.
5. Calculate time of one cycle form time spent over and below the average.
6. Calculate frequency of cycle.

7. Average 20 measurements.
8. Adjust amplitude of analogRead()-value according to equation 4.3

A visual representation of how the frequency is obtained is shown in figure 4.9. The frequency of the cycles averaged is calculated by using formula 4.4.

$$\text{Frequency} = \frac{1000000\mu\text{s}}{\text{Time high} + \text{time low}} \quad (4.4)$$

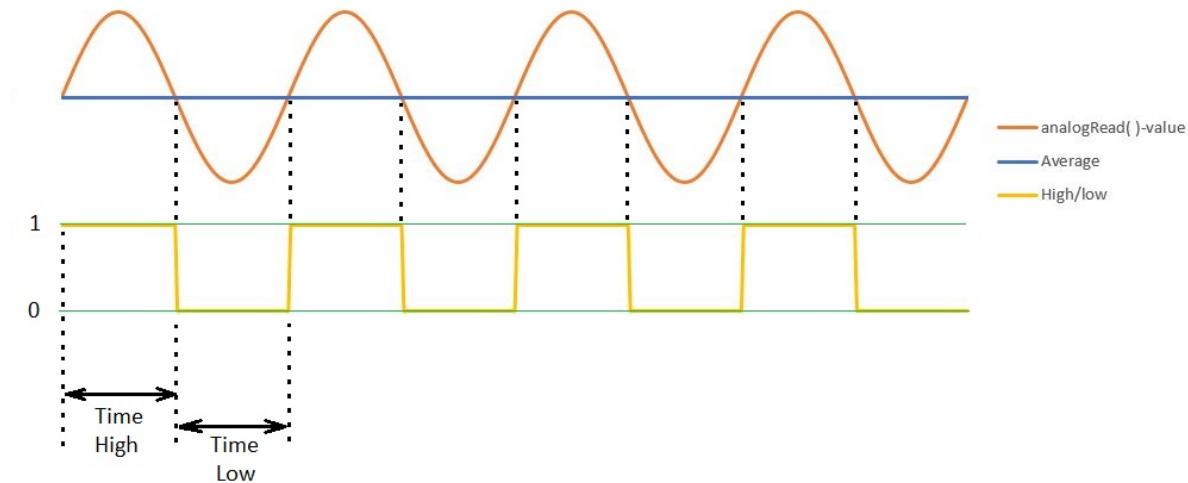


FIGURE 4.9: Script calculating the frequency

#### 4.4.2 Stroke of actuator

To be able to convert the estimated frequency and amplitude to a signal to regulate the actuators, the relationship between the output signal from the Arduino and the displacement has to be investigated.

This was done by sending a constant voltage from the DAC0 port of the Arduino through the amplifier and to the actuator. A dial indicator with a resolution of  $1\mu\text{m}$  was used to measure the displacements. A multimeter was used to measure the voltage applied to the actuator. The results from the measurements are presented in table 4.7 below.

From table 4.7 we can see that the maximum displacement of the actuator is  $450\mu\text{m}$ . This means that the simulated displacement of the virtual stack would be  $50\mu\text{m}$  when

TABLE 4.7: Applied voltage and corresponding displacement of actuators

<b>Actuator</b>	16.2V	33.5V	51.7V	71.0V
<b>1</b>	98 $\mu\text{m}$	221 $\mu\text{m}$	350 $\mu\text{m}$	450 $\mu\text{m}$
<b>2</b>	99 $\mu\text{m}$	224 $\mu\text{m}$	352 $\mu\text{m}$	453 $\mu\text{m}$
<b>3</b>	98 $\mu\text{m}$	224 $\mu\text{m}$	351 $\mu\text{m}$	451 $\mu\text{m}$

dividing the measured maximum displacement with the amplification factor found in equation 3.17. This was implemented in the simulations by having a restriction of the peak-to-peak amplitude of the simulated piezoelectric stack in the actuators to be  $50\mu\text{m}$ .

TABLE 4.8: analogWrite()-value, arduino voltage, amplified voltage

<b>analogWrite()-value</b>	<b>Arduino voltage output</b>	<b>amplified voltage output</b>
0	0.55V	0.0V
1023	1.09V	16.2V
2046	1.62V	33.5V
3069	2.16V	51.7V
4092	2.69V	71.0V

The relationship between the analogWrite()-values measured voltage from the Arduino and measured voltage from the amplifier is shown in table 4.8. The graph in figure 4.10 shows the relationship between the applied voltage and the displacement of the actuator. Using data from table 4.8 and table 4.7 a linear representation of the relationship between the analogWrite()-values and the displacement of the actuator was calculated shown in equation 4.5.

$$\text{Displacement of actuator} = 0.105 \text{ analogWrite()-value} \quad (4.5)$$

#### 4.4.3 Regulating output signal

A function to provide the desired output signal with amplitude and frequency estimated in the scripts mentioned is needed to regulate the signal for the actuator. Two different algorithms to generate a sine signal were investigated and compared to choose the best algorithm.

##### $\omega$ t-sine function

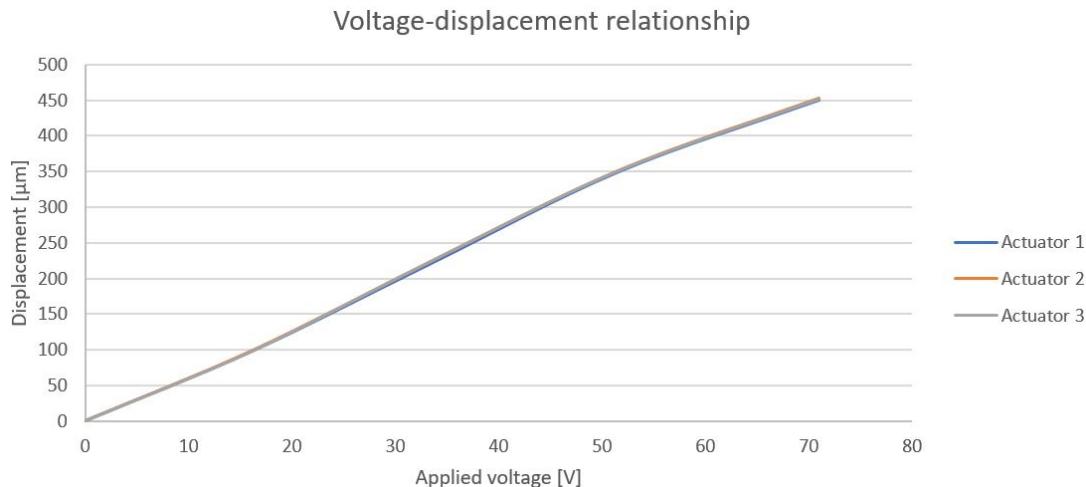


FIGURE 4.10: Voltage displacement relationship of actuators

The first script for generating a sine-wave is a mathematical formula for harmonic motion, as described in equation 2.13. In the simplified script shown in 4.11 the peak amplitude and mean value are set to 2047.5 while the frequency is hard coded to 100Hz. The code micros() returns the time lapsed since the start of the script. The variable "Current" is the output signal and is calculated for every round in the loop section. The time since the code started is measured and stored as a variable for each cycle. The Arduino uses this variable to calculate the value of the "current"-variable from the  $2047.5 \cdot \sin(\omega t) + 2047.5$  formula for every cycle in the loop function.

Since the loop section of the script has no delay()-function or other time demanding code, the frequency of the output signals are only dependent on the time spent running the code.

### Matrix-element sine function

The second script for generating a sine wave signal uses entries in a matrix to determine the output voltage. The entries in the matrix can be given as  $2047.5 \cdot \sin(\omega t) + 2047.5$ , where  $t$  is increased with  $\frac{1}{50}$  per increment. The function determining the matrix elements was chosen so that the whole resolution of the DAC pin was used. The loop function calls on the entries sequentially with the delayMicroseconds()-function controlling the frequency of the signal.

### Comparison of sine function scripts

Both scripts have their advantages and disadvantages. The DAC0 output pin was connected to an Instek GDS-1022 oscilloscope which showed the output signals of the Arduino

```

1 float OmegaTid=0.00;
2 float Current = 0.00;
3 float frekvens =100;
4 int Tid=0;
5 void setup() {
6   Serial.begin(115200);
7   pinMode(DAC0, OUTPUT);
8 }
9 void loop() {
10   Tid= micros();
11   float OmegaTid= ((float)Tid*2*3.1415*(float)frekvens)/1000000.0;
12   float Current= sin(OmegaTid)*2047.5+2047.5;
13   analogWriteResolution(12);
14   analogWrite(DAC0,Current);
15 }
16 }
```

FIGURE 4.11: Script for the  $\omega t$ -sine representation

Due. The Arduino Due spends  $205 \mu s$  to output all the entries in the matrix-element script when no delay was added, which leads to  $4 \mu s$  was used pr output. This leads to that the maximum frequency output with a 50 entry matrix is 4878Hz, which is above the desired range of up to 200Hz. Since the sine wave is represented by 50 entries in the matrix, the resolution of the signal is good. The largest drawback is that a `delayMicroseconds()`-function can only take in whole integers, so there is a restriction of the possible output frequencies. This implies that the frequency of the output signal can only be changed incrementally and not smoothly. Table 4.9 show some of the possible delay value inputs in the script and the corresponding output frequency. It is however possible to change the number of entries in the matrix and the delay time to obtain other frequencies. This would imply that multiple matrices have to be made in the code and the chosen matrix to regulate the sine wave would be determined by the frequency measured.

TABLE 4.9: Delay in matrix-element script and corresponding frequency.

Delay in script	129	130	131	132	133	134	135	136
Frequency	150.08	148.96	147.86	146.77	145.70	144.65	143.61	142.58

By using the time measuring script shown in 4.1 and inserting the  $\omega t$ -sine function in the loop, the time the code spent could be measured. The loop was run 12805 times in one second, meaning  $78 \mu s$  was used pr output. This means that the  $\omega t$ -sine function

```

1 int sine[] = {2048,2304,2557,2801,3034,3251,3449,3625,3776,3900,3995,
2 4059,4091,4091,4059,3995,3900,3776,3625,3449,3251,3034,2801,2557,304,
3 2048,1791,1538,1294,1061,844,646,470,319,195,100,36,4,4,36,100,195,
4 319,470,646,844,1061,1294,1538,1791}
5 ;
6 int DelayTime=10;
7 int i=0;
8 int Signal=0;
9
10 void setup()
11 {
12     pinMode(DAC0, OUTPUT);
13     Serial.begin(115200);
14 }
15 void loop()
16 {
17     for(int i = 0; i<50;i++)
18     {
19         analogWriteResolution(12);
20         Signal= sine[i];
21         analogWrite(DAC0,Signal);
22         delayMicroseconds(DelayTime);
23     }
24 }
```

FIGURE 4.12: Script for the matrix-element sine representation

uses about 20 times longer time for each output by the Arduino. At 200Hz the sine representation can maximum be represented by 64 points. The main advantage is that the function uses time as a variable, making the script able to adjust the frequency without being bound to incremental changes for the output frequency.

The chosen sine generating function was the  $\omega t$ -sine generator due to the ability to generate a more exact representation of the frequency desired. The speed of this script was also considered to be more than satisfactory for the intended use.

#### 4.4.4 Initiating the signal

The signal regulating the actuators have to be initiated at the correct time so that the optimal damping is achieved. The section of the preliminary script shown in figure 4.13 shows one possible way to determine when to initiate the counter signal.

The script works by first waiting until the `analogRead()`-value is below its average value. After this the code waits for the `analogRead()`-value to be above its average vale acquired

```
140 //-----START SIGNAL INITIATE-----
141 while(TidPhase<1){
142     Voltage=analogRead(A7);
143     if(Voltage<Average){
144         Voltage=analogRead(A7);
145         initiate= 1;
146         if(Voltage>Average && initiate >0){
147             enableActuator=1;
148             TidPhase= micros();
149         }
150     }
151 }
152 }
153 tidSig = micros()-TidPhase;
154 Current= sin((tidSig*2*3.1415*frekvens)/1000000)*(amplitude/2)+Average;
155 analogWrite(DAC0,Current);
156 //-----END SIGNAL INITIATE-----
```

FIGURE 4.13: Script determining when to initiate the regulating signal

earlier. When this happens the time is measured and stored in the "TidPhase" variable for reference. The "TidPhase" variable represents the time when the analogRead()-value is at average value and rising as  $\sin(x)$  is when  $x=0$ . This value is then used for determining when the regulating signals are to be initiated.

## 4.5 Adjusting properties in the simulations

Firstly a sweep test of the whole test bench was conducted to compare the simulated results with the measured results using Vibration View and the "Data Physics V350/DSA1-1k" shaker. The sensors, software and the shaker provided by Teeness are assumed to be tuned correctly and will be used as reference for calibrating the Arduino components.

A physical sine sweep test with the shaker and Vibration View was conducted first with the structure while the counter mass, bolt, washer and spring were removed. This was done to compare the behaviour of the real life test bench with the simulated test bench before adding the counterweight. This was partly done because the modeling of the actuators could be a source of error and partly because the modeling of the rubber could be a source of error.

A physical test using the shaker to oscillate in a motion described by a sine wave with an increasing frequency from 20Hz to 250Hz. The maximum absolute peak acceleration was set to 0.1g. Figure 4.14 shows the graph in dark blue(Ch2) representing the acceleration of the main mass as a function of frequency. The maximum acceleration was measured to be 1.7g at 178Hz. This is an amplification of 17 compared to the acceleration regulated by the shaker. By considering equation 2.45 the amplification factor of 17 would indicate a damping ratio of the rubber to be slightly below 3%. The green line shows the applied acceleration by the shaker and light blue line show the measured acceleration of the base. The red lines show half/double of the demand output and the yellow striped lines show the 50% increase/decrees of the demand acceleration.

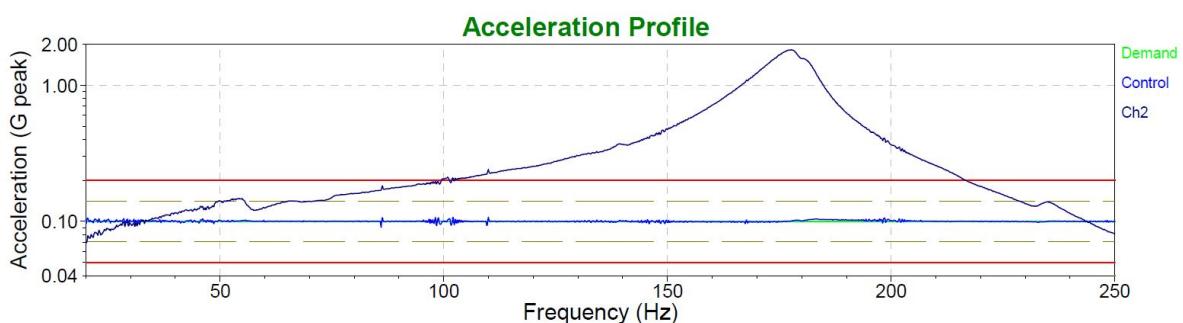


FIGURE 4.14: Results from Vibration View, frequency sweep at 0.1g acceleration without damping components

A simulation was preformed where the simulated shaker was regulated by a representation of the same input as in the test conducted in the laboratory. The free joint were regulated by describing the acceleration by the means of a sine formula. The formula describing the acceleration is given in equation 4.6

$$A\sin(\omega t) = 0.1\sin(2\pi x^2) \quad (4.6)$$

The input variables(x in formula 4.6) of the formula are restricted to either time or data from sensors. Generating sensor data that increases linearly would lead to more work compared to representing the frequency input with the time variable. Therefor the angular frequency is as in equation 2.12, is represented with the time variable x.

The acceleration data from one of the triads on the main body was investigated. At low frequencies it had the same acceleration ac the base and the acceleration got amplified when the frequency got close to the natural frequency of the structure. The Fourier transform of the frequency specter adjusted for the measured maximum acceleration is shown in the figure 4.15.

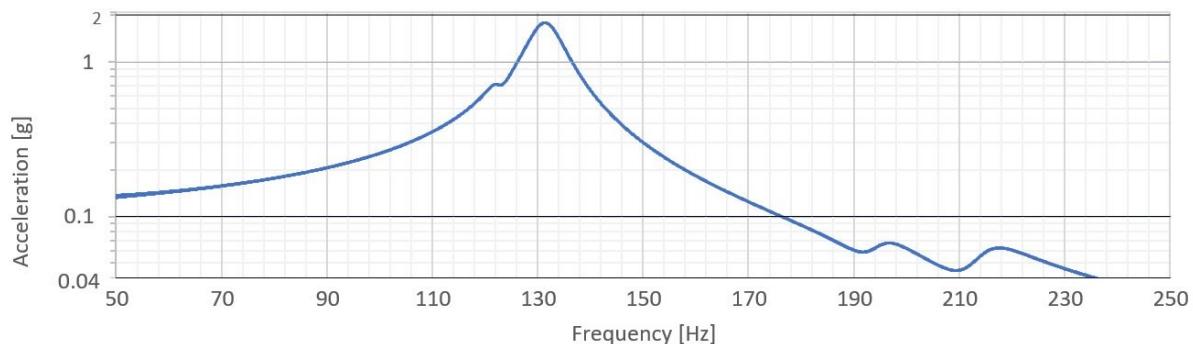


FIGURE 4.15: Results from FEDEM, frequency sweep at 0.1g acceleration without damping components before adjustments

The stiffness of the modeled rubber was increased by a factor of 2.4. This was done to change the eigenfrequency of the simulated test bench to be like the measured eigenfrequency of the physical test bench. The factor of 2.4 was obtained by changing the scaling of the stiffness parameter until the simulations showed that the simulated test bench had a natural frequency of 178Hz. The Rayleigh damping coefficients of the rubber part was adjusted so that the amplitude of the acceleration amplification was 1.7. The measured acceleration after the changes made is shown in figure 4.16.

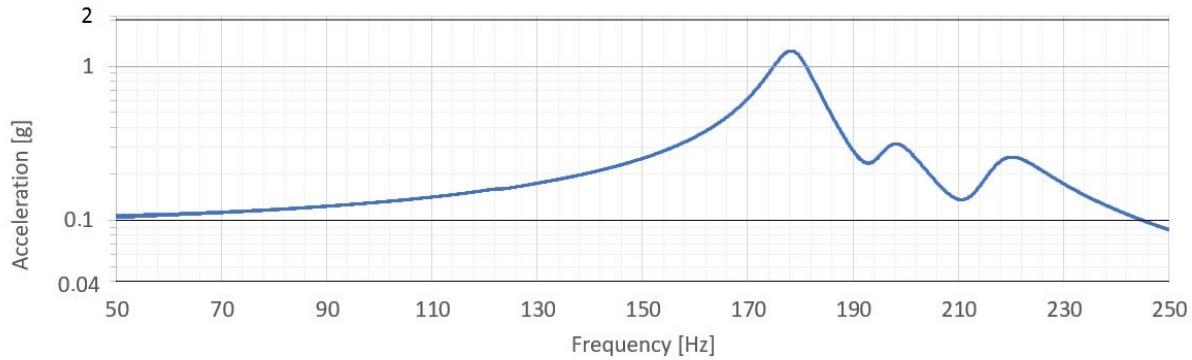


FIGURE 4.16: Results from FEDEM, frequency sweep at 0.1g acceleration without damping components after adjustments

After the adjustments mentioned above the counter mass, washer, bolt and spring was added to the physical test bench and the virtual test bench. This was done to compare the dynamics of the physical main mass and the dynamics of the virtual main mass with the complete test bench. The physical results from the shaker showed that the test bench acted like a vibration absorber as shown in figure 2.4. The accelerations of the main mass from the shaker test obtained in Vibration View is shown in figure 4.17.

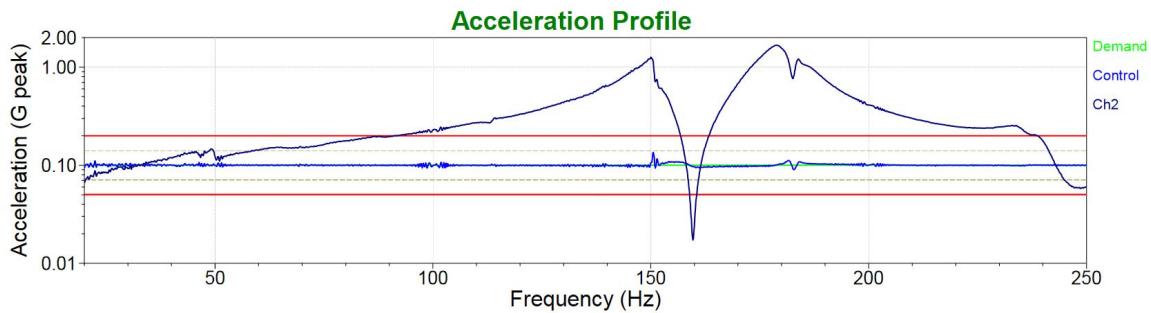


FIGURE 4.17: Results from Vibration View, frequency sweep at 0.1g acceleration with damping components

The acceleration of the main mass from the FEDEM simulations were different from the real life test. The change in behaviour due to the added components of the virtual test bench was mainly a shift towards a lower eigenfrequency (from 178 to 171.5) and a higher amplitude (from 1.7 to 2.2). The reduction of the frequency is how a impact damper would behave, but the increased amplitude is not. It is assumed that the reason for this behaviour is that the modelled actuator between the main mass and the counter mass is not modeled correctly.

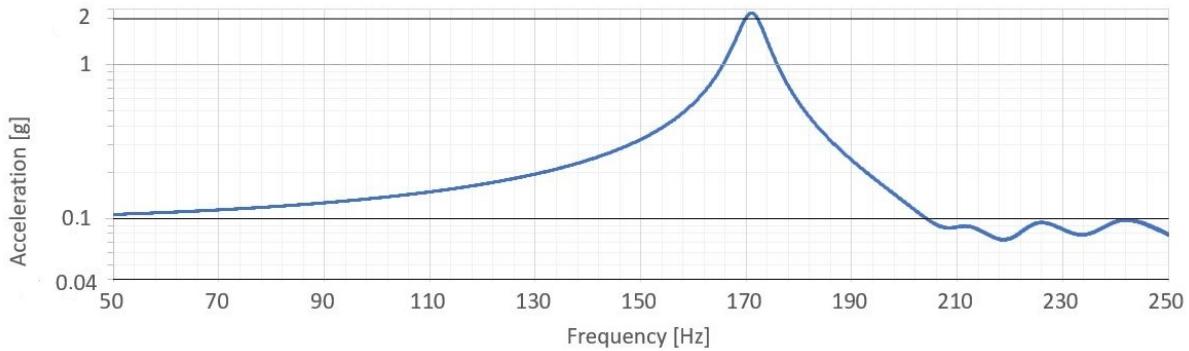


FIGURE 4.18: Results from FEDEM, frequency sweep at 0.1g acceleration with damping components

# Chapter 5

## Results

Several simulations were preformed to investigate the dynamics of the virtual test bench and how the actuators should be regulated to reduce the vibrations of the main mass. Further tests were conducted to measure the error of the Arduino readings and calculations. These results were used to evaluate if the Arduino is suited for regulating the active damping system.

### 5.1 Active damping simulations

The goal of these simulations were to investigate how to dampen the vibrations of the main mass. The effect of deviations in both amplitude and frequencies were also investigated.

In these simulations the simulated base excitation had a constant frequency of 140Hz was chosen as reference while the acceleration was set to  $5 \frac{m}{s^2}$  unless otherwise is specified. The frequency was chosen because it is a high frequency far from the natural frequency, making it a suited reference frequencies for general behaviour of the virtual test bench. The low value of the acceleration was due to the restrictions of the actuators. Also in real life testing of the test bench it is preferable to start with low forces for safety reasons. The free joins representing the shaker were regulated by displacements. The values describing the displacement of the shaker is given in table 5.1

The amplitude of the motion of the free joints was calculated by choosing the desired acceleration to be  $5 \frac{m}{s^2}$  and rearranging equation 2.15. The simulated displacement of the virtual main mass was obtained during the simulation when the displacement change of

TABLE 5.1: Input for sine function governing the base excitation function(SI units)

Description	Value
Frequency	140 Hz
Delay	0.00
Amplitude	6.46e-6m
Mean value	0.00
End	0.00

the simulated actuators were set to be none. The simulated displacement of the main body while the actuators were inactive was used as the reference displacement. The effect of the active damping system was measured by how much the displacement of the main mass was reduced while regulating the actuators. The results from the simulations are obtained from the "Model manage Result" list in FEDEM. The data from the main body were obtained from one of the triads in a rigid joint connecting the main mass to the upper part of the "rubber sandwich". The displacement and acceleration data obtained from the counter mass was from one of the triads in a rigid joint connecting the sphere of the top of the actuator to the counter mass. The data describing the virtual shaker were obtained from a triad in the free joint that was connected to the bottommost disc of the virtual test bench. A restriction for the displacement of the sphere of the actuator is set to  $450\mu\text{m}$  in accordance to the findings in table 4.7.

### 5.1.1 Amplitude of stack displacement

Firstly the simulated displacement of the main mass was obtained with the settings shown in table 5.1 while simulated actuator were inactive. The displacement of the main mass during this is from here on out called the reference displacement of the main mass. Secondly simulations where conducted where the displacement of the free joins representing the virtual shaker were held constant while the actuators were actuated. They were activated in the simulation by changes in length of the spring representing the piezoelectric stack in a sine motion. The displacements of the main mass from these two simulations were compared. The goal was to find the displacement of the simulated stack that resulted in the same virtual displacement of the main mass as the reference motion of the main mass. The amplitude of the motion describing the piezoelectric stack was changed until the displacements of the main mass became equal in the two cases. This was done to find the most suited stack displacement to dampen the vibrations of the main mass. Simulations were conducted with incremental changes of in the simulated

stack displacement was  $0.01\mu m$ . A representation of the virtual displacement of the main mass with varying the virtual stack displacement amplitudes is shown in the figure 5.1.

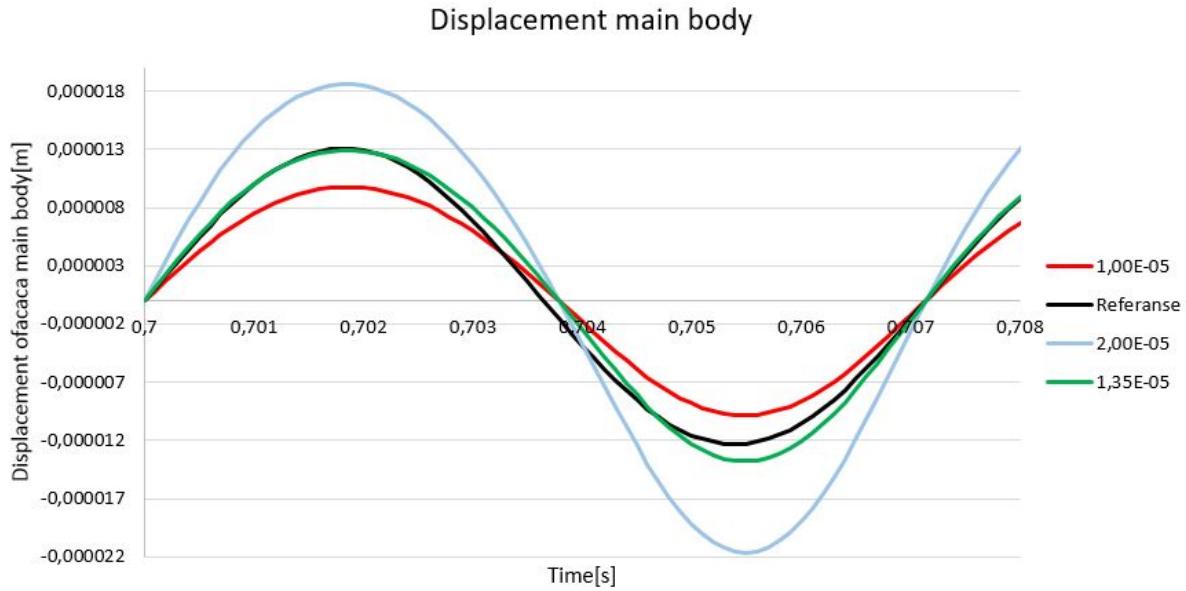


FIGURE 5.1: Amplitude for main body with varying amplitudes regulating the piezoelectric stack-function

The modeled piezoelectric stack amplitude that gave the smallest difference between the displacements of the main body compared to the case when only the modeled shaker was active was  $1.325e^{-5}m$  or  $13.25\mu m$ . This would mean a peak-to-peak amplitude of  $26.5\mu m$ . By using the amplification factor given by 3.17 the expected peak-to-peak displacement of the sphere would be  $238\mu m$ . The measured virtual displacement of the sphere was as a result of the stack displacement of  $26.5\mu m$  was  $288\mu m$ . Both the calculated and measured values are well within the limitations of  $450\mu m$ . The difference in the calculated displacement and the displacement obtained from the simulations were presumably due to the acceleration of the counter mass working in the same direction as the displacement of the sphere. This assumption was strengthened by investigating the relative displacement of the virtual actuator after the function regulating the motion of the virtual actuator was stopped. The relative displacement of the actuator had an amplitude of  $49\mu m$  after the regulation of the actuators stopped. The only forces acting on the actuators at that point were caused by the oscillations in the virtual test bench. The amplitude of  $1.35e^{-5}m$  for the simulated stack was used in all of the following simulations to regulate the actuators if otherwise is not noted.

### 5.1.2 Phase specter resulting in vibration reduction

If the relative displacement of the actuator work in the same direction as the displacement of the shaker simultaneously, the resulting displacement of the main mass would be amplified as a result for the actuation of the actuators. This would result in the displacement of the main mass getting a constructive interference, increasing the displacement of the main mass. To reduce the vibrations of the main mass a phase shift is applied to the function representing the piezoelectric stack. This is done so that the displacements by the main mass caused by the actuators work against the displacements caused by the shaker.

To investigate how the phase delay of the function regulating the actuators affects the displacement of the main body several simulations were conducted. First simulations with a phase delay fraction of a period varying from 0.0 to 1.0 were conducted with 0.1 fraction of a period as increment. Later simulations with 0.01 increment of fraction of period were conducted near the phases where the maximum reduction of vibrations were achieved and where the effect of the vibration reduction got near none. A representation of the delay in the function regulating the actuators versus the amplification of the amplitude to the displacement of the main mass is shown in figure 5.2.

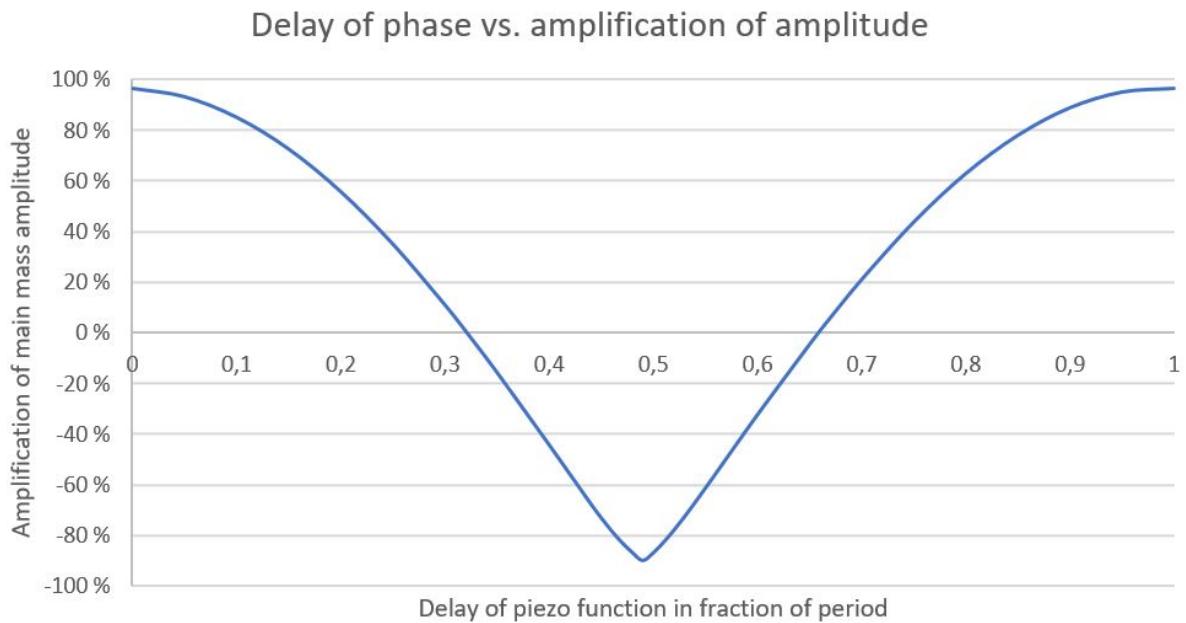


FIGURE 5.2: Fraction of phase vs. amplification of vibrations

The graph shows that at a delay of a 0.49 phase reduces the amplitude of the vibrations of the main mass about 90%. Further the graph shows that when the delay is between

0.32 and 0.66 of a phase, the amplitude of the displacement of the main mass is reduced compared to the case where function regulating the actuators is not applied. When the delay is between these values a reduction of the vibrations of the main mass is achieved. All other delays(in the same phase) will result in an amplification of the amplitude of the main mass, thus an vibration increase.

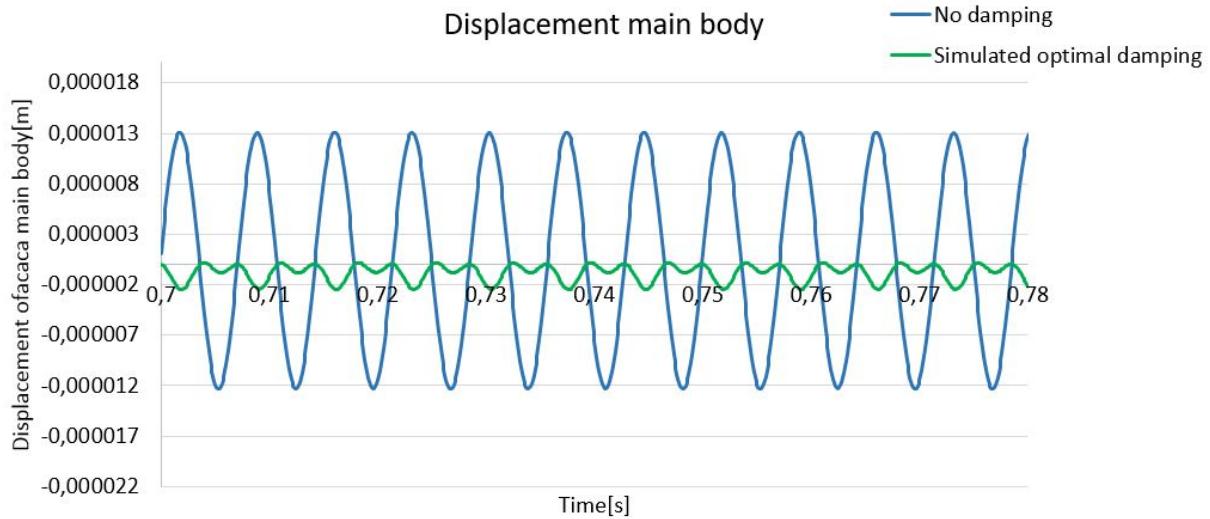


FIGURE 5.3: Displacement of main mass with and without damping by actuators

### 5.1.3 Change in frequency affects the delay specter

Several simulations were conducted to investigate how the change in frequency affects the delay span where reducing of the vibrations of the main mass is achieved by regulating the actuators.

Firstly the displacement of the virtual actuators and the simulated shaker were held constant while only changing the frequency. This also resulted in an increased acceleration of the simulated shaker motion and the actuator as can be seen in equation 2.15. The span of the phase where the regulation of the actuators reduced the vibrations of the main mass in the simulations are represented in table 5.2 and in the histogram shown in figure 5.4. The span for all the frequencies have an average value of 0.49 phase delay.

### 5.1.4 Effect of change in amplitude

Simulations were conducted to investigate how the amplitude of the function regulating the virtual piezoelectric stack would affect the amplitude reduction of the main mass.

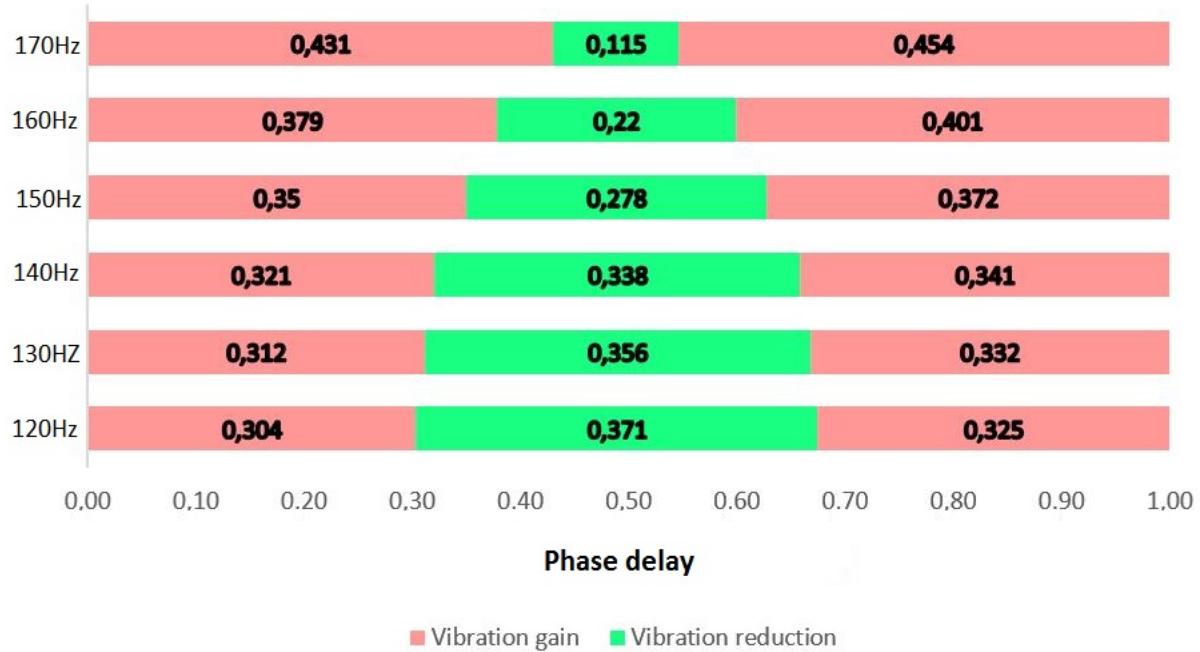


FIGURE 5.4: Specter of phase delay with vibration reduction for different frequencies

TABLE 5.2: Frequencies and corresponding minimum and maximum phase delay

Frequency	Minimum phase	Maximum phase
120Hz	0.304	0.675
130Hz	0.312	0.668
140Hz	0.321	0.659
150Hz	0.350	0.628
160Hz	0.379	0.599
170Hz	0.431	0.546

The effect of the vibration reduction with phase delays of 0.321, 0.49 and 0.659 were investigated while varying amplitudes. The amplitude of the function regulating the piezoelectric stack was increased and decreased with 10%, 30% and 50%. The vibration reductions(positive values) and the vibration amplifications(negative values) of the main mass are shown in table 5.3.

TABLE 5.3: Reduction(negative values) and gain(positive values) of displacement amplitude

<b>Amplitude</b>	<b>0.49 Phase delay damping</b>	<b>0.321 Phase delay</b>	<b>0.659 Phase delay</b>
$6.750e^{-6}$	54.7%	16.9%	17.0%
$8.508e^{-6}$	67.9%	15.3%	15.4%
$1.215e^{-5}$	89.5%	5.6%	5.5%
$1.325e^{-5}$	90.1%	-0.1%	-0.1%
$1.485e^{-5}$	81.4%	-6.6%	-6.6%
$1.755e^{-5}$	61.2%	-21.7%	-21.5%
$2.025e^{-5}$	39.7%	-38.9%	-38.5%

## 5.2 Active damping regulation

The main goal for this section is to investigate the scripts used for reading data from the accelerometers and how the data is treated.

### 5.2.1 Measurement of frequencies

Errors in the frequency estimations may occur, when measuring the frequency of the oscillations of the main body form the accelerometer with the Arduino. The small deviation between the real frequency of the main body and the applied control signal to the actuators will over time result in a phase shift between the signals. Assuming that a script similar to the script described in section 4.4.1 can be modified to read the error and adjust the control signal regulating the actuators an estimation of the allowed error can be made. The script uses 20 oscillations to determine the frequency. In calculating the allowed frequency error some assumptions have to be made.

1. There are 20 oscillations between every time the frequency can be changed by the Arduino.
2. The script regulating the frequencies shifts the frequency to obtain a phase that is closer to the optimal damping phase.
3. The first phase delay is exact, only the frequency is wrong.
4. The amplitude of the displacement of the main body should not be increased.
5. The time the Arduino needs to do calculations is during the sampling time.

In the case where the frequency is 140Hz and the amplitude of the stack regulating function is  $13.25\mu m$  the allowed error is 0.00845, calculated from equation 5.1. If the error is larger than this then the phase shift after 20 oscillations will be outside the phase span where the vibrations of the main mass are reduced, shown in figure 5.2. This will result in that the amplitude of the displacement of main mass will be amplified in the last oscillations before the regulating script can adjust the error.

$$\text{Maximum frequency error} = \frac{\text{positive delay specter}}{2 * \text{oscillations in one set}} = \frac{0.338}{2 * 20} = 0.00845 \quad (5.1)$$

This gives a maximum tolerated error of 0.06% on the frequency estimation at 140Hz.

### 5.2.2 Reading the frequency with Arduino

The Arduino was hooked up to an oscilloscope to investigate how precise the script could estimate the frequency. The oscilloscope is assumed to be perfect and therefore all deviations from the applied frequency to the estimated frequency of the Arduino are explained by error in the code or the Arduino.

The error from the measurements by the Arduino vs. the frequency is plotted in figure 5.5. 40 reading samples for the frequency measurements were used to calculate the maximum error and the standard deviation error. The plotted maximum allowed error is calculated for 120Hz to 170Hz by using equation 5.1. For the frequencies below 120Hz, the increase in allowed error was estimated by using the increase of allowed error found in the lowest three measured frequencies. This will not give exact values, but it may give an indication of the measured errors and the maximum errors allowed.

### 5.2.3 Calculating the acceleration with Arduino

The ADXL335 accelerometer was mounted on a disc that was fastened to the shaker. The shaker was set to generate a motion with the frequencies 50Hz, 100Hz and 150Hz at different accelerations. The script first estimated the frequency and then estimated the acceleration of the accelerometer. The results from the test is shown in table 5.4.

The script is sensitive to noise, because it uses the extreme values to determine the span of the analogRead()-values. These values are then used to determine the acceleration of

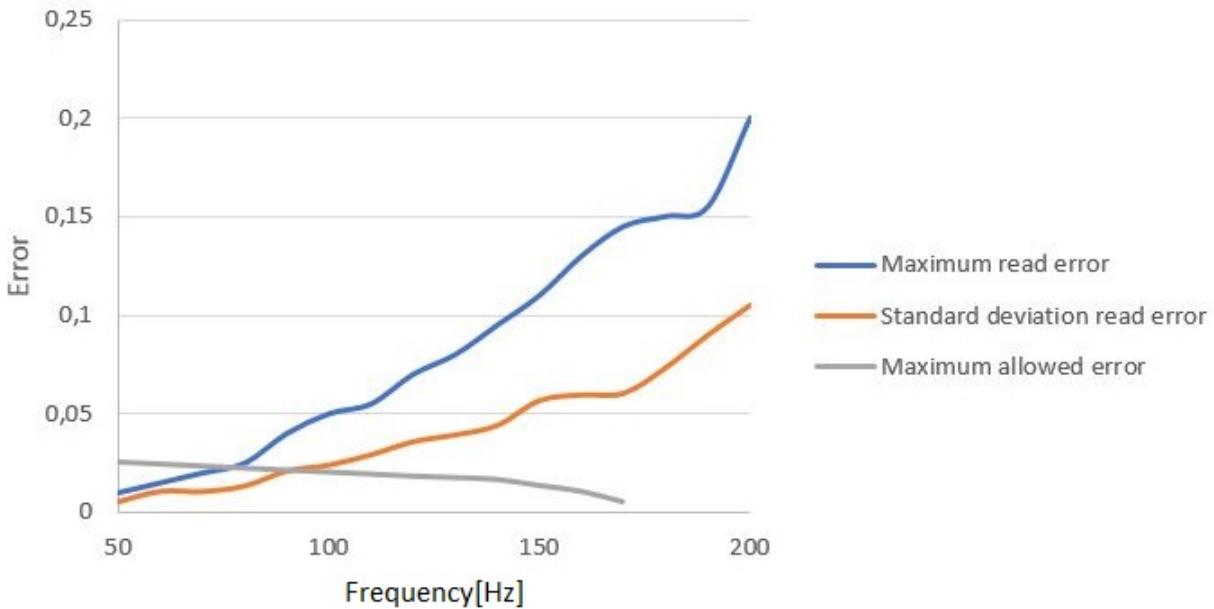


FIGURE 5.5: Maximum deviation from applied frequency fro oscilioscope and estimated frequency by Arduino

TABLE 5.4: Acceleration calculated with Arduino

Frequency	1g	2g	3g
50Hz	0.99	1.98	2.96
100Hz	1.04	1.99	2.98
150Hz	1.07	1.94	2.92

the accelerometer. The few first measurements if the accelerometer data are most affected by noise. These values should be ignored or treated in the final script.

#### 5.2.4 Applying regulating signal in correct phase

The script presented in [4.4.4](#) was run and results were found.

The time used by the script obtaining the frequencies for each loop was obtained by the script shown in figure [4.1](#). It completed the loop 43300 times in one second meaning that there was about  $23\mu s$  between every `analogRead()`-value measured by the Arduino.

The `analogRead()`-values are compared against the average value calculated. With a  $23\mu s$  time span between each measurement errors will occur. This is because the exact time the continuous signal from the accelerometer changes from above average to below average occurs between the measured points. The maximum error caused by this is the whole time

span between the measured points which in this case is  $23\mu s$ . For a complete sine period the error due to this can be  $\pm 23\mu s$ . Figure 5.6 show a graphical representation of how the error occurs. The figure shows points which represent the analogRead()-values read by the Arduino. The line between the points represent the continuous signal from the accelerometer. The horizontal line represents the average value which the analogRead()-values are compared up against.

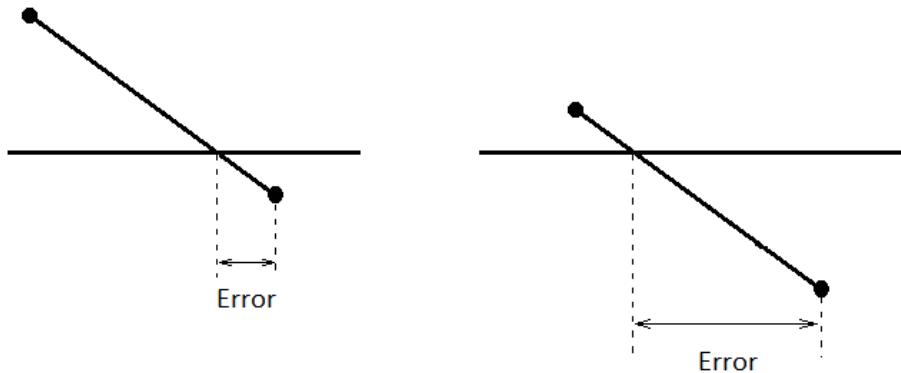


FIGURE 5.6: Error occurring as a result of time used by code instructions.

The error caused by this is assumed to be evenly distributed from the minimum value of  $0 \mu s$  to the maximum of  $23\mu s$ . The worst case error caused by this reading error is dependent on the frequency. This can be shown by investigating the formula for the estimation of the frequency when the error is included shown in equation 5.2.

$$\text{Frequency} = \frac{1000000\mu s}{(\text{Time high} + \text{time low}) \pm 23\mu s} \quad (5.2)$$

The scaled maximum error caused by the time increment between the analogRead()-values and the maximum read error measured is shown in figure 5.7. The error was scaled by dividing the maximum error by 9.5. The scaling factor was chosen to fit the curve representing the measured error and is a consequence from the averaging. This was done to show the correlation between the maximum measured error and the maximum calculated error.

The error discussed above will also affect the time for when the regulating signal should be initiated. The code uses  $6 \mu s$  from the instruction to initiate has been sent to the signal is sent. This was found by the "tidSig" variable shown in the script shown in figure 4.13. The initiation of the signal will then initiate from between  $6\mu s$  to  $29 \mu s$  after the acceleration voltage rises from below average to above average. An error of  $23\mu s$  is about

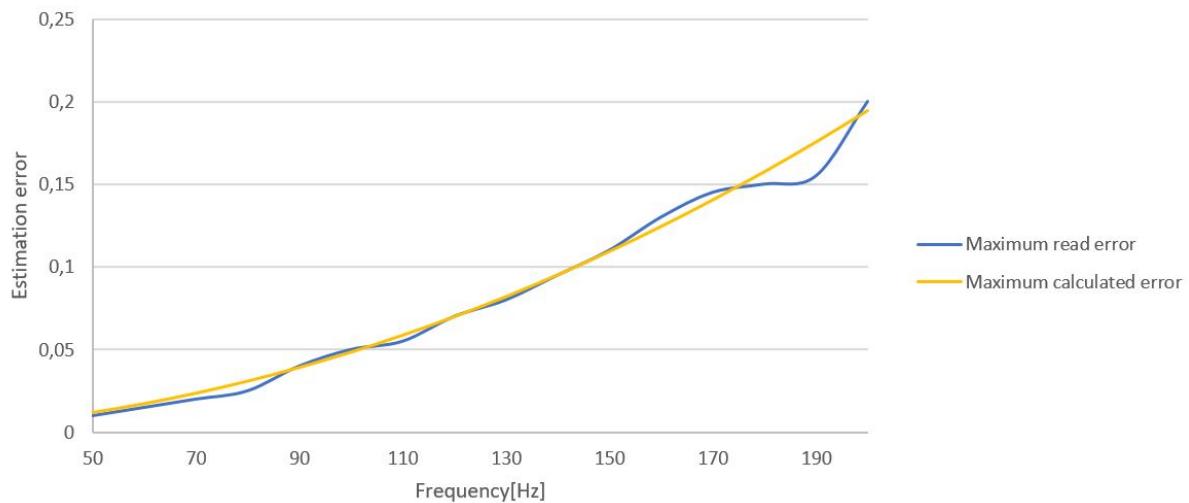


FIGURE 5.7: Measured maximum error compared to calculated maximum error.

0.5% of a phase at 170Hz. Meaning that the initiation of the signal is precise enough to not be considered as a focus area for improvement.

# Chapter 6

## Discussion

The goal of the discussion consists of the three parts presented in the list below.

1. First to discuss if the simulations are representable and how the results from these simulations should be used.
2. Secondly to discuss whether the physical test bench and the components chosen are well suited to demonstrate active damping systems or not.
3. Thirdly to discuss if an Arduino is suited for regulating the active damping system.

### 6.1 Simulations

Simulations are representations of the real world and some simplifications are made. Factors like humidity, temperature, noise from the surroundings are not included in the simulations. All the factors that affect the simulated test bench are therefore not addressed, only the assumed most important factors. The factors not implemented are considered not to affect the results in any significant way. The setup of the simulations is the main factor that determines how well it represents the physical test bench. The setup in this thesis is assumed to be fairly good, but improvements are possible with more data from testing.

The materials used in the simulations are chosen from the material list provided by the NX software. The material properties of the metals used in the real test bench, however, have not been investigated. This was not done because it is assumed that the potential

effect on the results would be neglectable when typical values were used in the simulations. The weight of the parts is the most important property for the metal parts. The parts were control weighed to compare to the simulations and make potential adjustments in the simulations. No changes were made due to weight difference between measure and simulated weight.

FEDEM only operates with linear elastic materials. The most important disadvantage of the linear representation is that the modeling of the rubber. The rubber has viscoelastic behavior in real life and the deformation of the rubber in the simulations are linear elastic. The stiffness of the rubber would be affected by the speed of the compression and the magnitude force acting on it. This could be modeled with by using data from testing of the rubber and adding a set dampers to the rubber making a KelvinâšVoigt model representation of the viscoelastic rubber.

The simulations where the actuators had a flexure represented by one part seemed to give a better representation of the sphere displacement compared to the simulation where the flexure was modelled with eight parts. This can bee seen by comparing figure 3.16 and figure 4.10. The actuators had an almost linear relationship as described in the data sheet 4.10. Further tests should be conducted to verify that the simulated actuators behave in the same way as the physical actuators. The results from the tests should be used to adjust the parameters in the simulations to minimize the deviation.

The eight parts used for representing the flexure of the actuators actuators were modeled with RBE2 connections and with a spring representing the piezoelectric stack and the flexure was connected. Choosing the stiffness and displacements of the spring used in the simulations were obtained by adjusting the spring stiffness and the stress free length of the spring so that the actuators gave the correct displacement and blocking force. However since the values used in the simulation model are far from the values presented in the data sheet, we can assume that the simulated model may behave differently from the actual actuator.

In the simulations the counter mass is connected to the top of the sphere of the flexure with a rigid joint. This is not like it is in the physical test bench. In the test bench the counter mass is placed on top of the spheres and is pushed down with the force coming from the tightened spring. The simulated test bench was made this way to eliminate simulation errors that occurred when other joints were used.

In the simulations the regulating signals are smooth and there is no deviation from the sent signal to the received signal or noise. In real life applications there will be noise altering

the signals slightly. This will probably increase the requirements for the regulating script since the noise may increase the error of the measurements.

Since the simulated test bench does not act like a vibration absorber, there is a difference in the fundamental dynamic behaviour of the test bench and the simulated test bench. The simulated test bench without the counter weight, spring, washer and bolt had the same behaviour as the results from Vibration View, after the stiffness of the rubber was adjusted. The effective stiffness of the rubber can be calculated by rearranging equation 2.22 and 2.12.

$$k = \omega^2 m = 1.6(2\pi 178)^2 = 2001335 \frac{N}{m} \quad (6.1)$$

The error occurred when the parts mentioned above were included in the simulations. Because of this it is assumed that the reason why the complete test bench did not act like a vibration absorber is due to the modeling of the actuators and how the counter mass is modeled to be connected to the actuators. The results from the simulations of the components show that the simulations give a good representation of the components' main physical properties. However there were some differences in the results from the simulations compared to the real life testing when all the components were included in the MBS. The results from the simulations should therefore be used for describing general properties and not to focus on the exact values found.

To find the optimal phase delay from the measured accelerometer signals to the initiation of the regulating script, physical tests should be conducted. The optimal phase can be found in the same way as in the simulations by investigating the effect of delay on the vibrations.

The virtual models used for simulations were adjusted to fit the results from the real life results from the shaker. The stiffness and damping of the rubber was adjusted and simulating of the piezoelectric stack. This resulted in the simulations being more representable of the physical test bench.

The simulations represented the physical behaviour of the test bench and its components relatively good. The deviations from the simulated dynamic response occurred when the counter mass, spring bolt and washer were included in the simulations. The error is assumed to arise due to the modeling of the actuators. However the general dynamic behaviour is assumed to give a representation of the real test bench that can be used to

determine the general properties. Therefore exact values should only be used to give an indication of what the values should be.

## 6.2 Test bench as demonstrator for active damping

The calculations used for designing the mechanical setup are simplified versions of the real applications. Therefor deviations between calculated properties and measured properties are to be expected. The natural frequency the rubber was dimensioned after was 100Hz, this frequency was calculated to increase to 136Hz when including the structural stiffness. The measured natural frequency of the test bench was however 178Hz. The increase from 136Hz to 178Hz may be a result from the viscoelastic properties of the rubber. With a lower eigenfrequency of the test bench a working demonstration of the regulating system would have less demanding requirements to the scripts and the regulating components. Therefor a real life demonstration could be achieved with less demands for the regulating system.

Achieving a lower natural frequency of the physical test bench is done by using a rubber with a lower stiffness. Changing the rubber is easy since the test bench is designed with an exchangeable rubber sandwich. Further the test bench can easily change the compressing spring and the counter mass if other physical properties of the test bench is desired. An extra disc can also be mounted between the rubber sandwich and the main mass to increase the vibrating mass the actuators are designed to reduce. This would also reduce the natural frequency of the test bench as shown by equation 2.65.

The chosen actuators were easily mounted with the screw holes on the protection cover. The design of the main mass made adjusting distance between the center of the main mass and the actuator possible. This might be practical if the counter mass of the test bench is changed with some other mass with different geometry. The discs made for the test bench were made with a flat cross section so that there were no unwanted wiggling between the plates. The parts were designed with tolerances and clearances so the test bench was easily assembled. The spring was easy to compress using the bolt to regulate the compression. The compressing of the spring with the bolt has a resolution of 8.4N pr revolution making the load from the spring fairly accurate. It is recommended to create a mark on the bolt to make it easier to see the rotation of the bolt and count the revolutions. The test bench as a whole was easy to move, assemble, mount on the shaker and parts were easily changed when desired.

The actuators were designed in a way that made the design of the main mass fairly easy and the actuators are easily regulated by voltage and do not need any external regulation. According to the data sheet and using the theory presented in chapter 2, the actuator chosen are well suited for being the actuator in a active damping system. To ensure that the actuator can operate at the desired frequencies with the full amplitude, testing at the laboratory should be conducted. These test should also be done with the counter mass mounted to see if the actuator behaves as described in the data sheet. Tests to investigate the phase delay of the regulating signals from the Arduino and the acceleration of the main mass will give needed data for the regulating script. The delay is needed to instruct the Arduino when the initiation of the regulating signal controlling the actuators should start. The delay should also be included in the simulations.

The slew rate of the amplifier can be a restricting factor. This can be investigated by running a frequency sweep test with a constant voltage amplitude. If the stroke of the actuator decreases with the frequency of the signal, the the slew rate probably is not sufficient. The amplifier used in this paper is an old Piezomechanic LE 200/150-1. Data sheet for this amplifier is not available due to its age. One of the more critical drawbacks with this amplifier is that it has only one output channel. The current from the output channel of the amplifier has to provide current demanded by all three actuators. A way to ensure that the amplifier is not a restricting factor is to acquire an amplifier with the desired properties. Thorlabs sells amplifiers designed to control piezoelectric actuators, some with three output channels and high current output. The "BPC303 - 3-Channel 150 V Benchtop Piezo Controller with USB" sold by Thorlabs can provide a 212Hz signal with 75V peak to peak amplitude[30] with sufficient current for the actuators presented in this paper from three channels. Therefore this amplifier is recommended by the author for the described application.

The simulations show promising result for how well the test bench can reduce the vibrations. The simulations showed a vibration reduction of over 90%. This implies that the concept used for damping the vibrations can give very good results. This vibration reduction is achieved while the peak-to-peak amplitude of the stack displacement is  $26.5 \mu\text{m}$ . The maximum displacement of the stack regulation is  $50\mu\text{m}$  as mentioned in section 4.4.2. The maximum displacement of the stack is roughly double of the displacement used in the simulations. This implies that the maximum acceleration of the virtual shaker causing the vibrations of the main mass can be doubled before the actuators are the restricting factor at 140Hz. This is an acceleration of about  $10 \frac{\text{m}}{\text{s}^2}$  or 1g. The acceleration will vary with the frequency as can be seen from equation 2.15 and figure 2.3. The maximum

acceleration the shaker can provide when the actuators as the restricting factor is shown as a function of frequency in figure 6.1. This was calculated by using equation 2.15 and the restriction of 1g as maximum acceleration at 140Hz

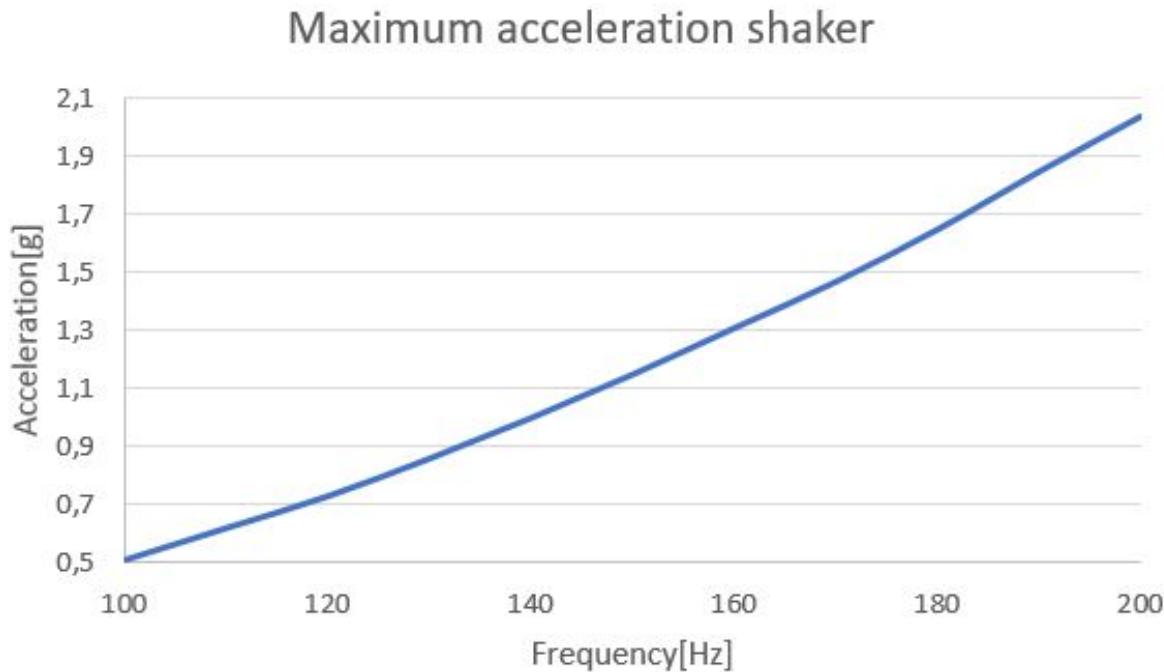


FIGURE 6.1: Maximum acceleration exerted by the shaker which the actuators can dampen optimally

By increasing the displacement of the actuators an increase in the vibrations of the main mass was shown in the simulation output. The maximum vibration damping at 140Hz was reduced as a result of a reduced displacement of the virtually regulated stack. At the same time the span of the fraction of the phase where damping was achieved increased. This means that there is a trade off with damping effect and span width where damping is achieved. This should be further investigated to find the best suited stack displacement when the precision of the script is known since this phase span determines the precision demand of the regulating script.

The results from the simulations show the span of where the actuation of the actuators reduce the vibrations to be about 0.3 of the oscillation period. This fraction can be higher if a lower stack displacement is used resulting in a reduced maximum damping effect.

In general the test bench should be able to demonstrate active vibration damping in the range of 100Hz to 200Hz when the accelerations are kept under the restriction given in figure 6.1. The time period fraction of 0.3 of a period is fairly small and decreasing with

an increasing frequency. This should however be possible to regulate with a precise and accurate regulation system.

### 6.3 Arduino as control mechanism

The voltage output for 1g acceleration of the ADXL335 accelerometer varies with the frequency. This is in this paper handled by using an equation to adjust the signal to normalize it with respect to the frequency. An ideal accelerometer should not give different voltage outputs when the frequency is within the bandwidth of the accelerometer.

By changing the ADXL335 with an accelerometer with an output independent of frequency, less calculations have to be run in the code, making the code faster. Further, equation 4.3 is a source to error that can reduce the precision of the regulating algorithm. The elimination of the frequency dependency of the voltage output to the Arduino may also be obtained by changing the circuit of the Arduino setup.

The noise in the circuit is fairly low as shown in tables 4.1 and 4.2. For the ADXL335 accelerometer this will result on a maximum reading error of 0.04g due to the noise in the circuit. This will most likely not result in any implications that will noticeably reduce the damping properties of the active damping system.

The noise will also affect the data used for estimating the frequency. The error that is presented in figure 5.7 and explained by equation 5.2 can be reduced and even eliminated by using a linear regression to estimate the exact time value when the signal from the accelerometer crosses the average. If a linear regression is used for this, the noise will be an issue and might result in error in the frequency estimation. The error caused by the noise has little effect on the precision of the initiation of the regulating signal. One possible way to reduce the implications of an error in the estimation is to reduce the number of samples used to calculate the frequency. Reducing the number of measurements to half allows the error to be twice as large with the same phase error after the signal has oscillated the number of periods equal to the sample size.

The Arduino Due has a clock speed at 84MHz, which means the CPU is 5.25 times faster than the Arduino UNO. The speed of the CPU is sufficient to do simple calculations with sub microsecond speed. By avoiding float numbers in complex formulas the high code speed result in a sufficient code speed for calculations. Data from the analog pins can be read or written within 2  $\mu s$ . The main issue for using the Arduino to regulate the

actuators is the error occurring due to the time between the measurements described in section 5.2.4. This error can be reduced by a shorter time period between reading the values from the accelerometer. This could be achieved by changing the code to run faster by using a more efficient code. This could also be achieved if the Arduino board had a faster CPU to conduct the calculations. Another approach to reduce this error is to use linear regression as mentioned earlier.

The drawback of that the serial communication is slow is not an issue when the script is completed. The serial communication is in this paper only used to get live feed of the values calculated during the script to obtain data. There is no need for serial communication during the run time of the script when it is calibrated and therefore the time spent on serial communication will not affect the time when the code is running.

The effect the noise has on the frequency estimation has been considered while making the scripts. The script divides the sine waves in periods when the analogRead() value crosses the average value. This is when the absolute value of the derivative of the sine wave is largest and therefore where the noise effects the change in the analogRead()-values the least.

The scripts used for regulating the damping system described in this thesis have high demands when it comes to accuracy and precision. The script estimating the frequency has a lower precision when the frequency increases although the accuracy is roughly constant. The increasing error is explained in section 5.2.4. The specter where the added vibrations caused by the actuators reduce the vibrations of the main mass is decreasing with increased frequency. The time used for one period is also inversely proportional to the frequency. These factors result in that the requirements for the regulating system increase rapidly with increasing frequencies. Further simulations where the acceleration is held constant while increasing the frequency and investigating the span should be conducted. This will show how the specter is affected when only the frequencies are changed.

The scripts uses average values to reduce the error that occur. The average values from 40 samples at 140Hz gave an error that could go on for 18 oscillations before it generates a negative effect for the vibrations, given that its original phase is correct. Figure 5.5 shows that the maximum frequency the script is suited to regulate the actuators in now is at about 80Hz.

The Arduino DUE has 12 analog input pins which can be coupled to sensors simultaneously. This could possibly be utilised when reading data from the accelerometers.

However the assumed best solution for reducing the error in the frequency reading script is the solution using a regression to obtain a more accurate time interval.

From figure 5.1 we can see that the actuation of the actuator has to be within a certain time period to be reducing the vibrations of the main mass. For 140Hz the range for where the signal would give a damping effect is when the delay between the excitation of the shaker and the counter motion is between 0.32 and 0.66 of the fraction of the period. This means that the signal delay in the simulation has to be between delayed with between 2.3ms and 4.7ms. This demands that the script has to be able to initiate the signal with high precision. The command `delaymicroseconds();` can be used in the script to adjust the delay time. This command has a one microsecond increment meaning that it is capable to have a precision of  $1\mu s$ (0.001ms) which is a resolution that is accurate enough for the delay requirements.

If the amplitude of the actuator is reduced one effect seems to be a larger spectre where the reduction of vibrations is present. The drawback is that the effectiveness of the damping is reduced. The optimal chosen amplitude will be dependent on how accurate and precise the regulating script is.

A very low error of the frequency reading is tolerated and this should be one of the focus areas for further work. The noise in the circuit should be reduced as much as possible and the scripts improved. The first step of improving the scripts should be the regression between the points to get a better estimate of the time used in one period. Further a way of handling the sensitivity of deviations when acquiring the maximum and minimum values for the frequency reading script should be addressed. For the active damping system to be able to give sustainable positive effect on the vibrations of the main body, a script to adjust the applied frequency is necessary. This script has to be made and has to meet all requirements mentioned. When these issues are addressed the Arduino seems to be able to be used for the regulation of the active damping system presented in this paper.

# Chapter 7

## Conclusion and Further Work

The test bench was modeled and produced to fit the chosen actuators. The test bench was easily mounted on the shaker provided and the regulating system using an Arduino Due was easily mounted. Simulations for analysing the dynamics of the test bench were conducted and scripts on the Arduino were investigated. The speed and precision of the results from the scripts were compared to the demands obtained from the simulations.

The general behaviour of the components in the simulation model behave as desired when investigated as independent components. However when they are combined in the MBS with all parts included the general dynamics of the virtual test bench was not as the measured dynamics of the real test bench.

The difference between the virtual and real the bench is assumed to be a result for the modeling of the piezoelectric actuators. The results from the simulations were used to describe the general behaviour. The maximum reduction of the vibrations achieved in the simulations was about 90% compared to when the actuators were inactive. The phase of the signal regulating the actuators affect the vibration reduction. The phase of the actuators must be regulated with millisecond precision.

The precision of the script estimating the frequencies is not good enough as it is presented in this paper. The demands for the script increase while the precision decreases with an increase of frequency. The script presented is precise enough to regulate be used for regulating a system of about 80Hz and below when assuming perfect phase initiation. Further work on the scripts measuring the frequency and the initiation of the signal is imperative for the active damping system to perform. The relationship between the measured acceleration and the applied displacement which gives the most effective vibration reduction is

not obtained in this thesis. This should however be obtained either by simulations or by results from physical testing before the script regulating the active damping is suitable.

An amplifier better suited for regulating the actuators should be used for regulating the actuators. Furthermore noise from the accelerometer should be reduced or an other accelerometer should be used. The simulations of the actuator should be improved for the to make the simulations of the test bench more representable of the physical test bench.

# Appendix A

## Appendix

```
int Voltage = 0;
int maxVoltage=0;
int minVoltage=4094;
int spanVoltage=0;
int noice=0;
int Current=0;
int SampleSize=0;
int Average=0;
int initiate=0;
int state=0;
int mwp=0;
int grit=0;
int trig=0;
int pwm=0;
int Print=0;
int amplitude=0;
int enableActuator=0;
int startAcceleration=0;
int count=0;
unsigned long Tid=0;
unsigned long Tid1=0.00;
unsigned long tidSig=0;
unsigned long TidPhase=0;
float sample=0.00;
```

```
float avgSample=0.00;
float avgDev=0.00;
float avgNoice =0.00;
float frekvens=100.00;
float frq=0.00;
float Acceleration=0.00;

void setup(){
Serial.begin(115200);
pinMode(DAC0, OUTPUT);
analogWriteResolution(12);
analogReadResolution(12);
Serial.println("Script started");
}
void loop() {

//-----START OBTAIN MAX MIN AVERAGE-----
while(Tid<3000){
Tid= millis();
Voltage = analogRead(A7);
if(Voltage>maxVoltage){
maxVoltage=Voltage;
}
if(Voltage<minVoltage){
minVoltage =Voltage;
}
}
while(Print<2){
Serial.print("Highest read value: ");
Serial.println(maxVoltage); // Remove Serial.print when script
complete Serial.print("Lowes read value: ");
Serial.println(minVoltage);
Serial.println("Estimating frequency...");

Print=10;
}
spanVoltage=maxVoltage+minVoltage;
```

```
Average=(spanVoltage)/2;  
amplitude= maxVoltage-minVoltage;  
// -----END OBTAIN MAX MIN AVERAGE -----
```

```
// -----START FREQUENCY ESTIAMATION-----
```

```
while(SampleSize<20){  
    Voltage= analogRead(A7);  
    if(Voltage<Average){  
        if(state==1){  
            if(count<20){  
                sample= sample+ frq;  
                avgDev =avgDev+noice;  
                count= count+1;  
            }  
            if(count==20){  
                count=0;  
                avgSample= sample/20.00;  
                avgNoice = avgDev/20.00;  
                frekvens=avgSample;  
                avgSample=0;  
                sample=0;  
                avgDev = 0;  
                avgNoice=0;  
                SampleSize=SampleSize+1;  
            }  
        }  
        state=0;  
        trig = micros();  
        mwp=micros()-grit;  
    } if(Voltage>Average){  
        grit= micros();  
        state=1;  
        pwm = micros()-trig; //if it is low, end the time  
    }  
    Tid1 = pwm+mwp;  
    frq=1000000.0/Tid1;
```

```
noice= pwm-mwp;

}

//-----END FREQUENCY ESTIAMATION-----


Tid= millis();
//-----START ACCELERATION-----
while(startAcceleration< 1){
startAcceleration=1;
Acceleration=amplitude/((0.01*frekvens*frekvens)-(3.443*frekvens)+(420.78));
Serial.print("Frequency is estimated to be: ");
Serial.println(frekvens);
Serial.print("Acceleration is estimated to be; ");
Serial.println(Acceleration);}
//-----END ACCELERATION-----


//-----START SIGNAL INITIATE
while(TidPhase<1){
Voltage=analogRead(A7);
if(Voltage<Average){
Voltage=analogRead(A7);
initiate= 1;
if(Voltage>Average && initiate >0){
enableActuator=1;
TidPhase= micros();}
}
}
}

tidSig = micros()-TidPhase;
Current= sin((tidSig*2*3.1415*frekvens)/1000000)*(amplitude/2)+Average;
analogWrite(DAC0,Current);
//-----END SIGNAL INITIATE
}
```

# Bibliography

- [1] Birk Brockmann Christian Brecher, Stephan Baumler. Avoiding chatter by means of active damping systems for machine tools. *Journal of Machine Engineering*, 13 (3), 2013.
- [2] Thorlabs piezoelectric theory. [https://www.thorlabs.de/NewGroupPage9.cfm?ObjectGroup\\_ID=5030](https://www.thorlabs.de/NewGroupPage9.cfm?ObjectGroup_ID=5030), . Accessed: 2017-04-16.
- [3] Thorlabs pk2fvf1 - amplified piezoelectric actuator with flexure mount. <https://www.thorlabs.de/thorproduct.cfm?partnumber=PK2FVF1>. Accessed: 2017-02-05.
- [4] N Lhermet A. Harms, B Denkena. Tool adaptor for active vibration controll in turning operations. *9th International Concerance on New Actuators, Germany*, 2004.
- [5] M. Horodinca A. Premount A.Ganguli, A. Deraemaeker. Active damping of chatter in machine tools-demonstration with a "hardware in the loop" simulator. *Journal of Systems and Control Engineering*, 219, 2005.
- [6] H. Frahm. Device for damping vibrations of bodies., April 18 1911. URL <http://www.google.no/patents/US989958>. US Patent 989,958.
- [7] C. Collette S.Chezník. A simple hybridization of active and passive mass dampers. *ISMA conferance*, 2016.
- [8] A. von Flotow N.w. Hagood. Damping of structural vibrations with piezoelectric materials and passive electrical networks. *Journal of Sound and Vibration*, 1991.
- [9] Valder Steffen Jr Felipe Antonio C. Viana. Multimodal vibration damping through piezoelectric patches and optimal resonant shunt circuits. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 28, 2006.
- [10] Iwao Yamaji Atsushi Matsubara, Minetaka Maeda. Vibration suppression of boring bar by piezoelctric actuators and lr circuit. *Elesiever*, 2014.

- [11] L. Walter J. Millitzer J. Rohlfing, T. May. Self-tuning velocity feedback control for a time varying structure using a voltage driven electromagnetic inertial mass actuator. *ISMA*, 2016.
- [12] Alessandro D'Ausilio. Arduino: A low-cost multipurpose lab equipment. *Springer*, 2011.
- [13] Arduino UNO overview, technical specifications and documentation. <https://www.arduino.cc/en/Main/ArduinoBoardUno>, . Accessed: 2016-04-13.
- [14] Bill Drury Austin Hughes. *Electric Motors and Drives (Fourth Edition)*. International series of monographs on physics. Newnes, 2013. ISBN 978-0-08-098332-5.
- [15] Arduino comparing comparing boards. <https://www.arduino.cc/en/Products/Compare>, . Accessed: 2016-04-13.
- [16] Arduino memory learning memory. <https://www.arduino.cc/en/Tutorial/Memory>, . Accessed: 2016-04-13.
- [17] Arduino sketch learning sketch. <https://www.arduino.cc/en/tutorial/sketch>, . Accessed: 2016-04-13.
- [18] Arduino Library library. <https://www.arduino.cc/en/Reference/Libraries>, . Accessed: 2016-04-13.
- [19] R. Hooke. *Lectures de Potentia Restitutiva, Or of Spring Explaining the Power of Springing Bodies*. [Cutlerian lecture. John Martyn, 1678. URL <https://books.google.co.uk/books?id=LAtPAAAAcAAJ>.
- [20] R.C.Hibbeler. *Mechanics of Materials, Eighth edition in SI units*. Pearson, 2011. ISBN 978-981-06-8504-6.
- [21] Gene Mosca Paul A. Tipler. *Physics for Scientists and Engineers, sixth edition*. Freeman, 2008. ISBN 978-0-7167-8964-2.
- [22] Fridtjov Irgens. *Fastetsære, 7th edition*. Tapir Akademisk Forlag, 2006. ISBN 82-519-2137-6.
- [23] David G. Rethwisch William D. Callister. *Materials Science and Engineering, 8th edition*. John Wiley and Sons, 2011. ISBN 978-0-470-50586-1.
- [24] Marie Dillon Dahleh William T. Thomson. *Theory of vibration with applications*. Prentice Hall, 1998. ISBN 0-13-651068-X.

- [25] L. G. Kraige J.L. Meriam. *Dynamics, SI edition, seventh edition.* Engineering Mechanics. John Wiley and Sons, 2013. ISBN 978-1-118-08345-1.
- [26] Erwin Kreysig. *Advanced Engineering Mathematics, tenth edition.* John Wiley and Sons, INC, 2011. ISBN 978-47064613-7.
- [27] Physik Instrumente piezoelectric tutorial. [http://www.piezo.ws/piezoelectric\\_actuator\\_tutorial/Piezo\\_Design\\_part3.php](http://www.piezo.ws/piezoelectric_actuator_tutorial/Piezo_Design_part3.php). Accessed: 2017-02-05.
- [28] Piezomechnaic products brochures pdf. <http://www.piezomechanik.com/products/>. Accessed: 2017-04-16.
- [29] Gunnar Hårkegård. *Dimensjonering av maskindeler, 2nd edition.* Tapir Akademisk Forlag, 2012. ISBN 978-82-519-1970-8.
- [30] BPC303 - 3-Channel 150 V Benchtop Piezo Controller with USB overview, technical specifications and documentation. [https://www.thorlabs.de/newgroupage9.cfm?objectgroup\\_id=1912](https://www.thorlabs.de/newgroupage9.cfm?objectgroup_id=1912), . Accessed: 2017-04-13.