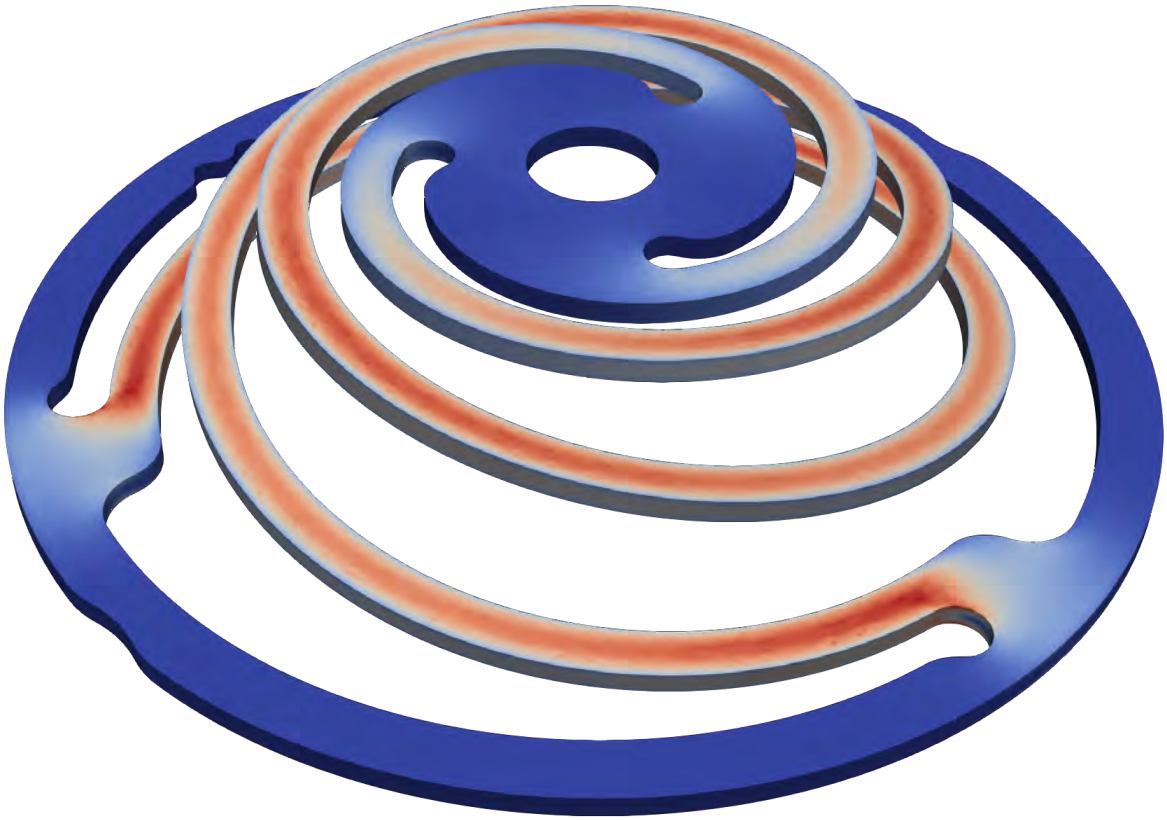
CHALMERS



Development of FEM-based Simulation Methods for Facilitating the Design Process of Electromagnetic Vibration Energy Harvesters

Master's thesis in Applied Mechanics

FELIX ERIKSSON

Department of Mechanics and Maritime Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2019

MASTER'S THESIS IN APPLIED MECHANICS

Development of FEM-based Simulation Methods for Facilitating the Design Process of Electromagnetic Vibration Energy Harvesters

FELIX ERIKSSON

Department of Mechanics and Maritime Sciences

Division of Dynamics

CHALMERS UNIVERSITY OF TECHNOLOGY

Göteborg, Sweden 2019

Development of FEM-based Simulation Methods for Facilitating the Design Process of Electromagnetic Vibration Energy Harvesters
FELIX ERIKSSON

Master's thesis 2019:32
Department of Mechanics and Maritime Sciences
Division of Dynamics
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: +46 (0)31-772 1000

Cover:
Spring from energy harvester model Q, subjected to static load in a simulation. The colors indicate von Mises-stress, where red represents higher stresses.

Göteborg, Sweden 2019

Development of FEM-based Simulation Methods for Facilitating the Design Process of Electromagnetic Vibration Energy Harvesters
Master's thesis in Applied Mechanics
FELIX ERIKSSON
Department of Mechanics and Maritime Sciences
Division of Dynamics
Chalmers University of Technology

Abstract

Electromagnetic vibration energy harvesting is an emerging technology for extracting small amounts of electrical energy from vibrations. It can be used to power sensors and data transmission equipment for monitoring industrial processes and machines in hard-to-access-places without easy access to electricity, without requiring regular replacement like batteries.

Methods, models and software to aid the design process of springs for electromagnetic vibration energy harvesters have been developed in this work. It has been carried out at ReVibe Energy, a company which develops such harvesters.

Previous FEM simulations of eigenfrequencies in the harvesters showed poor correspondence with experiments. To investigate this, experiments to measure stiffnesses of springs were set up. This was done by suspending weights in the springs. It was found that also the stiffness of the springs corresponded poorly to finite element simulations. There were also significant differences between spring individuals that were nominally identical. Dimensional measurements were conducted on the springs, and it was found that the spring arms were generally narrower than specified in drawings for the components. There were also arm width variations between individuals, which correspond to and explain the between-individuals stiffness variations. When simulations were performed with corrections for arm width, the stiffness prediction error was reduced by approximately 50 %.

The methodology employed in designing and manufacturing springs was studied, in an effort to improve spring design while making the design work less cumbersome. With the purpose of removing manual iterative design loops, which are very time-consuming, a method for automatically generating and evaluating different spring designs was conceived. It consists of automatic changes to a CAD-file in Autodesk Inventor, export of a STEP file containing the component geometry, meshing of the component with GMSH, finite element analysis on the mesh using Elmer, and finally evaluation of results. This was implemented in a software program written in Python.

Keywords: Electromagnetic vibration energy harvesting, finite element method, beam elements, parametric simulations, open source, water jet cutting, tolerances

Preface

First things first: Mother and father, you are awesome. I went through my childhood being able to disassemble perfectly working things in the garage, half-finishing a tree house (and a million other projects), discovering the world by bike and motorcycle, only to have to repair them in the garage so I could leave a mess there while learning how to master the tools. Thank you for enduring this. I would never have had the desire to study engineering without that.

I want to thank ReVibe Energy for providing me with premium office space which I have happily occupied while you have been crammed into small rooms. Your aggregate motivation and willingness to work towards a good technological product is superior that of most industrial giants. In particular, I feel thankful for my wonderful supervisor Andreas Josefsson, who has endured a more or less daily bombardment of questions and happily helped me understand energy harvesters, and the challenges and problems they pose. Your knowledge, ideas and thoughts have guided me well through this spring.

Samanta, you make me feel like a rock star when I produce good MATLAB plots, and you're almost as good a wiseacre as me. But most importantly I want to thank you for wanting to listen to and understand me even when I'm not able to myself.

And I cannot possibly forget the teachers that have put so much effort into making this learning journey possible. You have shown me how delightful it can be to think, and what joy it is to see the world through equations. From learning how to add apples and bananas, to using FEM for knowing if stuff will break, already before it does – it's thanks to you. Special thanks to Thomas Abrahamsson, who has been the examiner of this project, and provided many good suggestions and insight into the problem.

If only steel could drink coffee, we would have a lot less problems with fatigue.

Nomenclature

API	Application Programming Interface
Autodesk Inventor	Software for 3D CAD modelling of products
CAD	Computer Aided Design
GMSH	Mesh generation software
Elmer	FEM solver software
Model Q	Small energy harvester model
Nastran In-CAD	FEM meshing and solving software built into Autodesk Inventor
SLSQP	Sequential Least-Squares Programming, optimisation algorithm
STEP	Standardised format for exchanging 3D CAD model data
Bobbin	Cylinder around which a coil is wound
GUI	Graphical User Interface

Contents

Abstract i

Preface iii

Nomenclature v

Contents vii

1 Introduction 1
1.1 Mechanical parts . . . 1
1.2 Design optimisation . . . 2
1.3 Possible applications of energy harvesting . . . 3
1.4 Benefits of using simulations in spring design . . . 3
1.5 Aim . . . 3
1.6 Limitations . . . 4

2 Method 4
2.1 Previous work on FEM modelling of harvesters . . . 5
2.2 Spring stiffness . . . 6
2.2.1 Static load experiment . . . 7
2.2.2 Spring stiffness simulation . . . 8
2.2.3 Dimensional spring measurements . . . 8
2.3 Parametric simulations . . . 10
2.3.1 Parametric simulations using Timoshenko beam elements . . . 11
2.3.2 Python spring optimisation . . . 12

3 Results and Discussion 14
3.1 Spring stiffness experiment and simulation . . . 14
3.2 Spring dimension measurements . . . 15
3.3 Parametric simulations with beam elements . . . 17
3.4 Parametric simulations in Python . . . 18
3.4.1 Notable issues encountered during development . . . 19
3.4.2 Optimisation . . . 20
3.4.3 Parametric CAD model . . . 21

4 Concluding remarks and future work 22
4.1 Spring manufacturing process . . . 22
4.2 Resonance frequency error . . . 22
4.3 Continued use of parameter sweep software . . . 22
4.4 Improving computational efficiency . . . 22
4.5 Optimisation . . . 23
4.6 Other applications for parameter optimisation software . . . 23

References 24

1 Introduction

Electromagnetic vibration energy harvesting is the technology of extracting electrical energy from ambient mechanical vibrations with the use of magnetic fields and coils. It is a field that has come into the interest of the scientific community and commercial entities in the last decade, largely thanks to the growing interest in the industry for increasing capabilities of monitoring processes and machine condition, and gathering this data wirelessly. This concept is important in the emerging fields of Industrial Internet of Things (IIoT) and Industry 4.0 [1]. Vibration energy harvesting (VEH) can be used to power sensors and transmission equipment, to enable monitoring in hard-to-access places to where it is difficult or expensive to lay cables from the power grid. In many such applications, batteries are used, but VEH has the potential to provide power without requiring replacement at regular intervals unlike batteries.

Electromagnetic VEH works similarly to induction generators, using the principles of the Faraday's law, which relates the change of magnetic flux density B through a wire loop, to the electromotive force ϵ induced in the wire (voltage). The following expression can be found for the voltage ϵ induced in a coil

$$\epsilon = -N\frac{d}{dt}\left(\iint_A BdA\right) \tag{1.1}$$

where A is the surface enclosed by the coil [2], and N is the number of windings. This means that if the magnetic field passing through the coil is changing in time, a voltage will be induced.

In practice, an electromagnetic vibration energy harvester ("energy harvester" or simply "harvester" henceforth) is often realised as a system of magnets, coils and springs enclosed in a rigid housing. The springs suspend either the magnet or the coil (where the former is most common), in order to allow movement with respect to the other component when the housing is subjected to vibration, and subsequently induce a change in the magnetic flux density passing through the coil, which in turn creates an alternating voltage in the coil. This is illustrated in Figure 1.1. The coil is connected to an electrical circuit, which utilises the voltage and current induced. The electrical circuit is often designed to rectify, regulate and store the electricity before the power is used by an appliance.

1.1 Mechanical parts

The springs, together with the mass they suspend, make resonance possible, at frequencies that are governed by the stiffness of the spring and the mass that is suspended (eigenfrequencies). In the undamped uniaxial linearly elastic case, this is described by the well-known expression

$$\omega = \sqrt{\frac{k}{m}} \tag{1.2}$$

where ω is the eigenfrequency (rad/s), k is the stiffness (N/m) and m is the mass (kg) of the oscillating part [3].

This means that if the system is placed on some vibrating body, with vibration frequencies close to the eigenfrequencies of the system, and with proper arrangement and orientation of the spring-mass-coil system, the magnet will be put in a state of strong harmonic motion. This harmonic motion gives rise to an alternating voltage in the coil, which can be used to extract electrical energy.

To reach Equation 1.2, it must be assumed that the springs are linearly elastic. This means that force needed to deform a spring is proportional to the distance one wishes to displace it, measured from the spring's undeformed position. This can be written as

$$F = kd \tag{1.3}$$

where F is the applied force (N) and d is the displacement (m).

In a three-dimensional continuous elastic solid, there exists not only one, but an infinite number of resonance frequencies, each with a unique mode shape (deformation shape). In a discrete approximation of such a

continuous system, e.g. when using the finite element method, the number of frequencies is equal to the number of degrees of freedom in the model. Energy harvesters are generally designed so that the fundamental vibration mode (lowest frequency) for the spring-magnet system induces a large change in magnetic field through the coil. This is the eigenmode that one aims to attain during normal operation. Higher-frequency modes are not considered valuable for energy harvesting, because they usually result in much smaller changes in magnetic field through the coil, and therefore much lower voltage and power levels. However, higher frequency modes can still be interesting to analyse, since they might create unwanted resonance which can potentially damage the harvester. The goal of studying these modes in the harvester design process, is therefore often aimed at avoiding them during operation.

1.2 Design optimisation

Optimisation is the procedure of finding the set of parameters that result in the best possible performance under some constraints, measured with some metric. The metric, which one aims to maximise or minimise, is often called objective function.

In design optimisation, the objective function and constraints are chosen to represent performance of a component or structure. Good performance can be characterised by e.g. low stress or high stiffness. For a spring in a harvester, it is important that the spring stiffness k is correct, in order to achieve resonance at a frequency which is prevalent in the application according to Equation (1.2). Therefore, the spring constant can be used as a constraint during optimisation. It is also advantageous to design the spring so that the stresses during operation are low, in order to decrease the risk of fatigue damage, or to increase the allowable

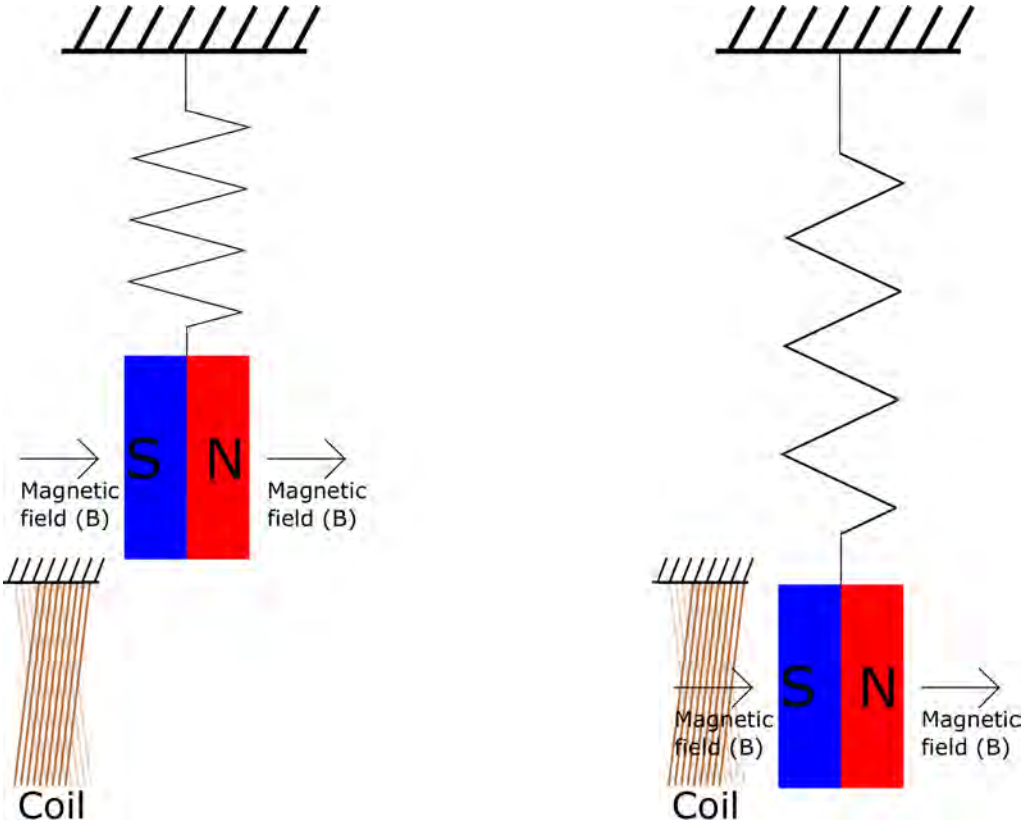


Figure 1.1: *Schematic of an electromagnetic energy harvester. The illustrations show the two extreme points of the magnet position. When the magnet moves between the two positions, the integral* $\iint_A BdA$ *in Equation 1.1 changes value with time, and subsequently induces a voltage in the coil. In practice, the relative orientation between the magnet and coil may vary, and multiple magnets, springs and serially connected coils can be used. The fixed support symbols in the illustration designate attachment to the harvester housing.*

deflection of the spring. This can be formulated as a minimisation problem. Minimisation problems are often mathematically described in a standard format called negative null form, as follows:

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimise}} && \sigma_{\text{vM,max}}(\mathbf{x}) \\ &\text{subject to} && k(\mathbf{x}) = \omega^2 m \\ &\text{where} && \sigma_{\text{vM,max}} \text{ is the maximum von Mises-stress in the spring,} \\ & && \mathbf{x} \text{ are design variables of the spring,} \\ & && k \text{ is the spring constant,} \\ & && \omega \text{ is the desired vibration frequency and} \\ & && m \text{ is the mass of the suspended magnet assembly} \end{aligned} \tag{1.4}$$

Gradient-based optimisation is a class of widely used optimisation methods, which rely on information about the gradient of the function to be minimised. The search direction in the space of design variables is chosen to be the same as the negative gradient of the objective function. [4]

1.3 Possible applications of energy harvesting

The electrical power that is typically achievable with an energy harvester placed on a vibrating industrial machine is small ($< 1\,\text{W}$). Therefore, it is not viable to use this technology to replace conventional power generation for any high power demanding application. However, due to recent development in energy efficient wireless transmission technology, the set of possible applications has been enlarged.

An energy harvester, together with a wireless transmitter, can power a sensor and enable sensor data acquisition without external wiring or batteries. This can potentially make it possible to place sensors on machine components in locations with very limited accessibility and/or on movable machine parts without the need of replacing batteries periodically.

1.4 Benefits of using simulations in spring design

The company ReVibe Energy develops energy harvesters of the kind previously described. In this development, it is desirable to have accurate finite element method (FEM) models of the mechanical spring-mass system.

Mechanical FEM simulations of the springs in a harvester can be used to determine the vibration frequency at which peak power is produced, given a certain harvester design, and which stresses the springs are subjected to. This can make it possible to design springs with correct stiffness, and low stresses during operation, when aiming for a certain peak power frequency. It can also be used to avoid unwanted resonance from higher-frequency modes. This is important for ensuring that the harvester does not break prematurely due to fatigue or impacts. Such simulation models were already in use at the company, but they had not attained sufficiently good accuracy.

1.5 Aim

The purpose of this project was to aid in the design process of energy harvesters by obtaining understanding about which phenomena and design parameters influence the harvester's performance, and with this understanding develop simulation and design methodologies for facilitating harvester design while ensuring good performance. The main focus of the project has been on the mechanical part of the company's harvesters, which is constituted by the springs, which are the main deforming elements, and the magnet holders and magnets that are suspended by the springs.

Mechanical simulation models of the harvester have to accurately represent a number of different parameters present in the energy harvester. These include the boundary conditions for the springs and the weights of the

moving parts. The finite element method was used on the elastostatic equations (Navier-Cauchy equations) to numerically evaluate the problem.

The overarching goal was to simplify the process of choosing the relevant harvester parameters, most notably the spring dimensions. A goal which was conceived early in the project was to embody the results in a software program, which can automatically compare different designs and help the user choose the best design.

The reason why good simulation models are desirable, is because such tools can aid in reducing costs, and give inspiration for how to enhance performance in the products they model. While the project was mainly focused on understanding and modelling the physics in the energy harvesters, more directly applicable modifications of the harvesters were not excluded from investigation and testing, when ideas about improvements to the products arose and time permitted, for example in geometrical design or material selection.

1.6 Limitations

In this project, the work was mainly be focused on the currently existing product models of energy harvesters that ReVibe Energy have developed. Before accurate and validated simulation models for these have been achieved, other energy harvester types or concepts were not considered valuable to model. The work was restricted to electromagnetic vibration energy harvesting throughout the entire project.

In the modelling of energy harvester springs, only linearly elastic behaviour has been considered. The springs of a harvester are subjected to very many cycles of loading, and they must therefore be designed so that the stresses in the spring are far away from the yield limit during normal operation. Knowledge of their behaviour during yielding is thus of limited value. It was therefore considered sufficient to use linearly elastic modelling.

The design of the electrically active parts of the harvester have largely been considered to be outside the scope of this thesis.

2 Method

The first stage in the project was to thoroughly understand the problem. This included studying theory relevant to the problem, which is mainly electromagnetics and dynamics applied to the energy harvester. The company has an extensive collection of literature, articles and other information about vibration energy harvesting which was used in this process.

When the theory governing harvester physics had been sufficiently understood, and the challenges that arise in energy harvester design had been identified, time was dedicated to understanding the current simulation models. This involved getting familiar with the software programs that are in use at the company, and understanding how the component interactions, interfaces and materials were represented in the simulation models.

Experiments were performed in order to assess and document the performance of the harvester, and how the performance changes in response to alterations in chosen components, assembly, load and other possibly relevant variables. These experiments were conducted in the company's lab, which hosts actuating and sensing equipment for vibrations, together with necessary data acquisition systems. The experimental results were compared to simulation data in applicable cases. In the experiments, it was considered advantageous to decompose the system by testing the spring-magnet system without the presence of electromagnetic damping from the coil, to more accurately be able to pinpoint where errors arise.

The frequency with which the simulation model eigenfrequencies had been compared in previous work at ReVibe, was measured as the vibration input frequency at which the energy harvester outputs the most power over a simple resistive load. In addition to this type of tests, it also proved beneficial to incorporate more decomposed experiments here, like measuring displacement of a spring when a known force is applied to it.

As more understanding was gained, the work transitioned into trying modifications of the simulation models, and conceiving new methods for efficiently and accurately designing and simulating harvester springs.

The tests and simulations were done with different versions of the components. The springs are the main deforming components in the operation of the energy harvester, so their stiffness, material and geometry are the most influential factors for the eigenfrequencies of the harvester. Therefore, different springs were tested and compared to simulations.

2.1 Previous work on FEM modelling of harvesters

At the start of the project, it was decided that the main goal would be to investigate and improve mechanical FEM modelling of the moving parts of the energy harvester, and use this to aid the design process. The moving parts are the springs and the magnet which they suspend. The primary goal in modelling these is to get accurate predictions of the main (lowest) eigenfrequency for the system, since this is the frequency at which the desired harmonic motion (normal mode) of the magnet is attained, which creates voltage when coupled with a coil. In the mechanical modelling, electromagnetic effects are not considered, and in pertinent experiments the coil is either removed or disconnected from other electronics in order to minimise the influence of electromagnetism.

In addition to enabling eigenfrequency prediction, an accurate FEM model of the spring-mass system makes it possible to calculate the stresses that the spring is subjected to during operation. This is useful for ensuring that the spring will not break during operation, due to fatigue or impact events.

Modal analyses of the spring-magnet system had already been performed by engineers at the company, with the goal of predicting the main (lowest) resonant frequencies that each different spring would give rise to. In these analyses, the geometry of components was taken from the same CAD-files as had been used for manufacturing the parts, and the material data for the spring was provided by the steel supplier. The simulations had been performed with the Stress Analysis tool in Autodesk Inventor, a CAD software program used for designing the company's products [5]. Figure 2.1 depicts results from such a simulation. However, the simulation results

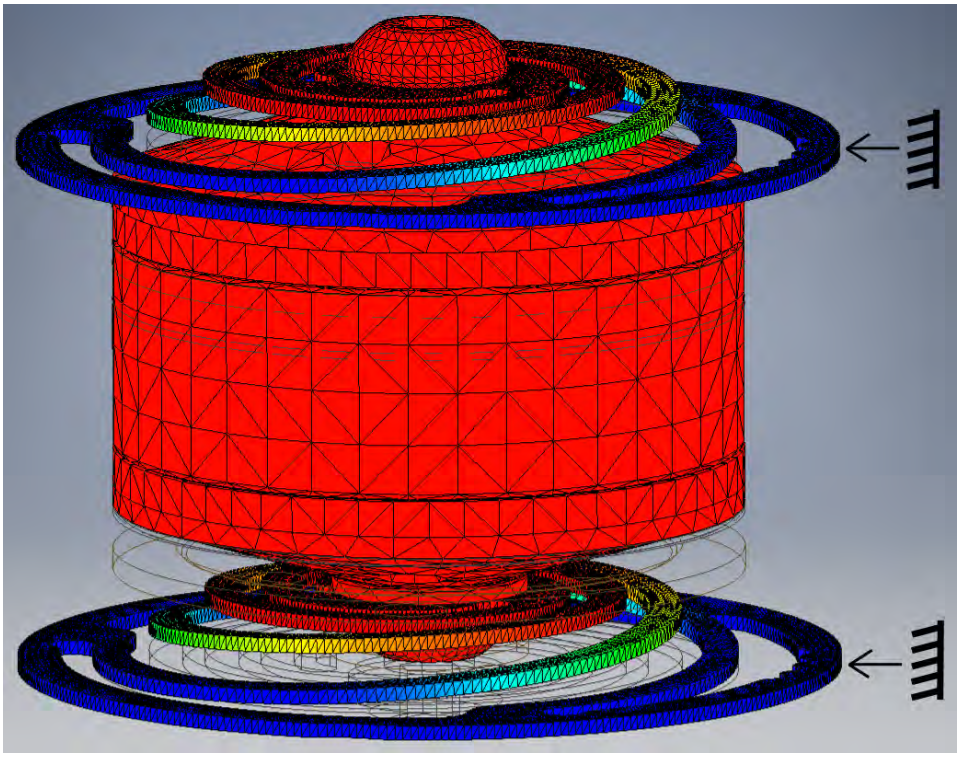


Figure 2.1: *Visualisation of the first eigenmode of the spring-magnet system in model Q, calculated with Autodesk Inventor. Two identical springs are mounted on each side of the magnet assembly. The outer perimeter surface of each spring is fixed in both rotation and translation around the entire springs. The springs are planar when undeformed. The colours indicate displacement, with increasing magnitude from blue to red.*

did not agree very well with experimental data, often predicting resonant frequencies with more than 10 % error.

2.2 Spring stiffness

The eigenfrequency predictions in a modal analysis can be understood using Equation 1.2. This tells us that it is the spring stiffness and the mass of the object the spring suspends that govern the resonant frequency. This assumes that the springs are linearly elastic (i.e. they obey Equation 1.3). To pinpoint where the problem lay, it was considered advantageous to try to measure the spring stiffness independently, instead of measuring the frequency which depends on both stiffness, weight and electromagnetic damping.

The effective stiffness of the spring not only depends on the material and geometry of the spring, but also on how the spring is attached to the housing. This is represented by boundary conditions in a simulation. In earlier simulations of the springs performed at the company, all the nodes on the outer perimeters of the springs were fixed, i.e. prevented to move in both translation and rotation (see Figure 2.1). The design of the harvester is such that the outer ring of each spring is clamped between two plastic washer-like parts, as shown in Figure 2.2. The clamping pressure comes from a threaded cap that seals the housing, and presses the entire assembly together. If the clamping pressure is great enough, it will prevent sliding between the spring and plastic washers, and it had previously been assumed that this was the case. The clamping pressure depends on the torque with which the threaded cap is fastened in the housing.

The following questions were now identified as relevant for determining why eigenfrequency predictions are inaccurate:

- **Is the spring stiffness accurately modelled?**
  - This is fundamental for being able to correctly predict the movements of the magnet, most importantly the eigenfrequencies of the spring-magnet system. To accurately model the spring stiffness, both the geometry and the material properties of the spring have to be accurately known.
- **Do assembly procedures affect performance significantly?**

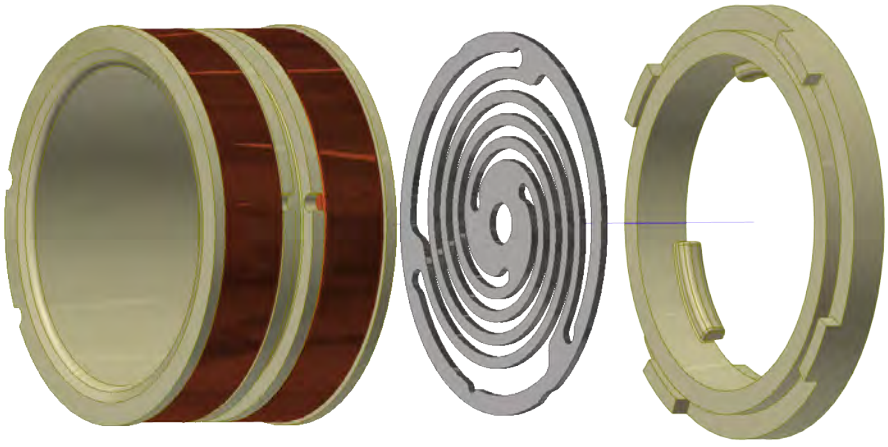


Figure 2.2: *Demonstration of spring attachment in model Q. The spring is clamped between the ridge on the washer to the right in the figure, and the bobbin to the left, which also has a smooth clamping surface similar to the washer. This assembly is placed in a cylindrical housing, which is closed by a threaded cap, which provides the clamping pressure.*

  - In order to set boundary conditions that well represent physical reality, it is important that the assembly procedure results in a component with consistent characteristics. In some harvester product models, the fastening of the spring is designed so that the outer perimeter of the spring is squeezed between two washers when the cover cap of the product is screwed in place. If the assembly torque of the cover cap is low enough, the spring might slip between the washers instead of being securely clamped, altering the effective stiffness of the spring assembly. Are there any risks for such insecurities with the current assembly procedure, and if so, how can they be mitigated?

- **Do interactions between the magnet and surrounding ferromagnetic materials affect harvester characteristics significantly?**
  - In addition to the desired effect of inducing current in the coil of the harvester, the permanent magnet in the harvester also interacts with other ferromagnetic materials. Most notably, the springs in the harvester are made of a magnetic steel alloy, and are mounted very close to the magnet. Does this affect the effective stiffness of the magnet-spring system?

- **Is there a large variation in stiffness between nominally identical springs?**
  - Manufacturing methods affect material properties and geometrical tolerances in the spring, which in turn affect its stiffness. These deviations from the nominal properties result in an inherent uncertainty in the harvester's characteristics, which cannot be predicted with a nominal simulation model. Therefore, insight in this uncertainty is fundamental to understanding the accuracy limit in the simulations, and if the uncertainty is too high, it might be worth considering other materials and/or manufacturing methods.

- **Are the weights of the components accurately represented?**
  - The weights of the moving components (most notably the magnet) are very important to accurately know in order to theoretically calculate the resonant frequencies of the system.

One of the energy harvester models currently under development, named model Q, was chosen as the first to investigate. Its springs are made using precision water jet cutting of sheet metal. The material is a type of cold-rolled steel specifically intended for use in springs. The material properties used in simulations were Young's modulus $E = 193\,\text{GPa}$ and Poisson's ratio $\nu = 0.32$. Most of the other harvester models have springs in similar material and with the same type of pattern for the spring arms, differing mostly in scale and magnet/coil arrangement from model Q. Therefore, it was considered possible to draw generally applicable conclusions from tests and simulations on this model.

2.2.1 Static load experiment

In order to investigate the four first points in the above list, an experiment for statically loading the springs while measuring their deflection was set up. In this experiment, a harvester housing and end cap were modified to allow for suspension of weights from the spring, and measurement of the spring's deformation, while the spring would still be mounted and clamped in the same fashion as in an unmodified harvester. The experimental setup can be seen in Figure 2.3. For displacement measurement, a laser triangulation sensor from Micro-Epsilon was used, model optoNCDT 1420, with a measurement range of 25 mm [6]. The harvester housing was fixed with a vice in a level position. The laser triangulation sensor was placed above the housing, where it had free line-of-sight to the upper side of the spring.

For model Q, there are several nominally different springs available, which enable the harvester to be tuned to different resonant frequencies. In the experiment, three different spring models for model Q were chosen as subjects; the most flexible spring, the stiffest spring, and one in between. The spring models only differ in spring arm width, see Figure 2.5.

For each spring model, a number of different individuals were randomly chosen. For each individual, tests were conducted at three different cap fastening torque levels; 2 Nm, 4 Nm and 6 Nm, giving rise to different spring clamping pressures. The torque was set using a click-type torque wrench from Red Cycling Products, with adjustable torque setting between 3 Nm-15 Nm.

Figure 2.3: *Test setup for measuring the stiffness of energy harvester springs. Weights (W) are attached to the spring from below, while the deflection is measured from above using laser triangulation (L).*

In each test, four displacement measurements were taken; without applied load (as reference), and with 100 g, 200 g and 300 g load applied, respectively. Before taking the reading, it was made sure that any oscillation of the suspended weights had vanished.

The load-displacement data was imported into MATLAB, where `polyfit` was used to fit linear functions to the test data, and approximate the spring stiffness as the inverse of the lines' slope.

2.2.2 Spring stiffness simulation

In order to compare the static load experiment results to computational models, linear static simulations were set up in Autodesk Nastran In-CAD for the spring models that were used in experiments. A load was applied to the center of the geometry, to produce a symmetric deformation. This simulation setup is shown in Figure 2.6.

2.2.3 Dimensional spring measurements

In the experiment described in Section 2.2.1, large discrepancies between simulated and measured spring stiffnesses were found. To investigate this further, the dimensional accuracy of the spring arms was measured. The measurements were done with a micrometer screw gauge, with a resolution of 0.01 mm. One of the

micrometer's spindles was fitted with a spherical tip to enable measurement on concave surfaces, such as the

Figure 2.4: *Displacement measurement using laser triangulation. In order to provide an even surface for the laser to measure the displacement of, a small piece of electrical tape was attached to the screw head in the centre of the spring.*

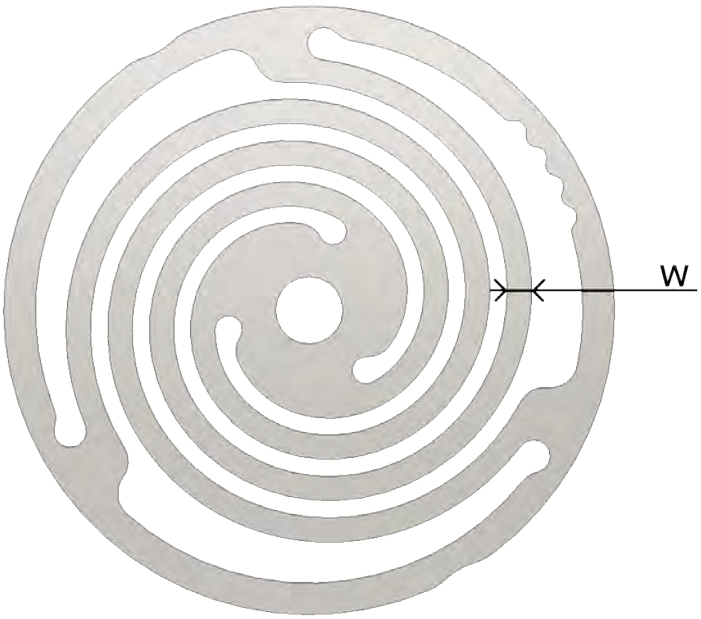


Figure 2.5: *A planar spring of the type used in model Q, viewed from a direction normal to the plane. The spring arm width, which varies for different spring models, is indicated with w.*

inner curved surface of the spring arms, see Figure 2.7. Each spring arm was cut off the spring and divided into three roughly equal parts. The spring arm width was measured once for each part, yielding a total of 9 measurements per spring.

2.3 Parametric simulations

Simulations are useful tools in product development, since they can provide information about a product's performance before any product or prototype thereof exists. This can be taken one step further, by automating simulations so that large numbers of possible designs can be simulated and evaluated, without requiring the user

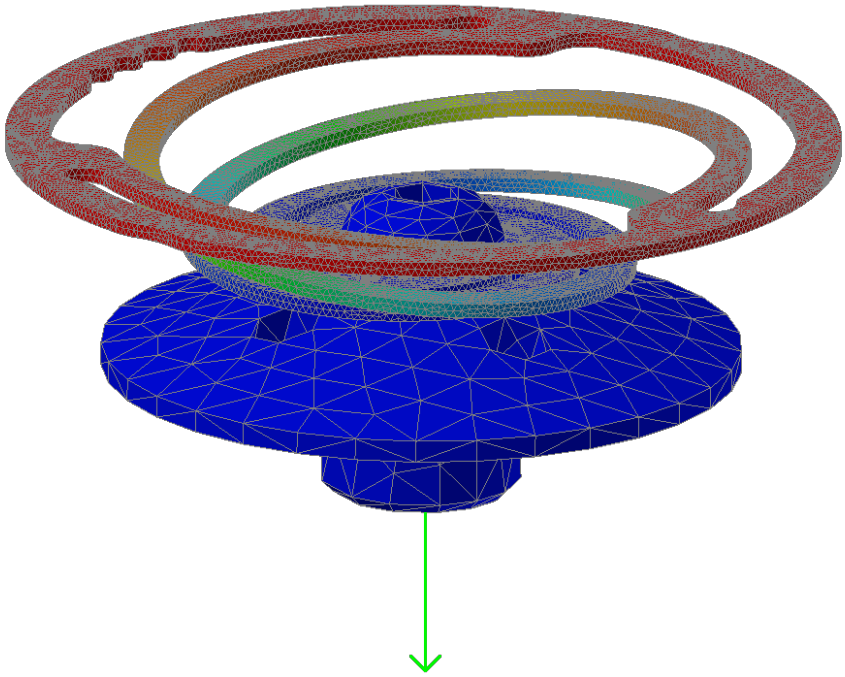


Figure 2.6: *Depiction of results from linear static FEM simulation in Autodesk Nastran In-CAD. In the picture, the spring is connected with a screw to a component which is normally used to attach the magnet. This component was also used to attach the weights to the spring during the static load experiment, cf. Section 2.2.1. The outer perimeter of the spring is fixed, and a force is applied to the center of the magnet attachment component, at the green arrow.*

Figure 2.7: *Measurement of spring arm width.*

to manually do the steps of changing the design, setting up a simulation, interpreting results, and comparing them for different designs.

When designing a spring for a harvester, the desired stiffness can often be known beforehand. The desired resonance frequency is known from the intended application, and if the weight of the moving part is known, the target spring stiffness can be calculated from Equation (1.2). However, there is no unique way to create a spring geometry which gives a certain stiffness – for example, one could make spring arms wider while making the spring thinner. Therefore, one has the opportunity to choose a geometry which gives low stresses during operation, which reduces the risk of fatigue damage to the spring. Doing this manually, i.e. without automatic design changes and simulation, is very time-consuming, and it is difficult to know which design parameter values to try.

The commercial software programs for FEM simulation available at the company were those built into Autodesk Inventor, namely "Stress Analysis" and "Nastran In-CAD". These do not have built-in functionality for design automation on springs in the fashion described above, and they are not easy to make communicate with other software through e.g. an application programming interface (API). Due to these limitations, other options were considered for implementing this.

The goal was to create a software which could

1. modify the design of the spring by changing numerical parameters,
2. test the design by simulation,
3. evaluate the results,
4. repeat this for ranges of parameter values, where each range and number of points in that range is specified by the user, henceforth called "parameter sweep"
5. or repeat the first three steps until a good (or even optimal) set of parameters is found.

2.3.1 Parametric simulations using Timoshenko beam elements

During operation, the parts of the spring that are responsible for most of the deformation are the spring arms, meaning the spiral shaped parts that connect the outer ring to the inner ring on the spring (cf. Figure 2.6). These have a constant rectangular cross-section throughout their entire length. It was therefore considered reasonable to model the spring deflection by taking only the spring arms into account. This was done using Timoshenko beam theory.

A finite element model using Timoshenko beam equations was implemented in MATLAB. Stiffness and mass matrices for 3D beam elements from Theory of Matrix Structural Analysis [7] were used. Such elements have two nodes each, at the end points of the beam element. Each node has six degrees of freedom (DOF) – three corresponding to translational deformation, and three for rotational deformation.

The goal was to emulate the mechanics of the actual spring, where the spring arms are attached to a washer-like ring at the outer perimeter, and a small plate in the centre. When the spring is deflected, the outer ring and central plate are rigid enough to deform only negligibly. Therefore, homogeneous Dirichlet boundary conditions were applied to all the DOFs of the outermost node on each arm to fix them completely. For the centremost nodes of each arm, all DOFs except for the out-of-plane translation were also fixed, to allow for the out-of-plane deformation that the spring experiences during operation, while mimicking the fixation that the centre plate provides.

After conducting dimensional measurements on spring arms, as described in Section 2.2.3, functionality was added for simulating displacement with varying spring arm widths. In the spring arm width measurements, each arm was measured in three points. In an effort to reproduce the arms' shape in the simulation, each arm's width was linearly interpolated between the three measurement points, i.e. in two segments.

2.3.2 Python spring optimisation

While the Timoshenko beam model works well for predicting spring deflections (and subsequently stiffness), and is also possible to use for stress calculations in the spring arms, it has limitations. It cannot capture all the stresses in the more complex geometry that constitutes actual springs. Problems with excessive stresses often arise in discontinuities in the geometry, such as where the spring arms attach to the outer ring of the spring. Therefore, a better way to accurately assess the risk of failure in the spring using simulations, is to do a more thorough simulation which also models the attachments of the spring arms to the rest of the geometry. To this end, a software program for automatically conducting such simulations was conceived.

In Autodesk Inventor, product design can be done parametrically. This means that parameters can be set and used in mathematical expressions to relate size and positioning of different features to each other. If used correctly, it allows the user to create designs that can be modified by simply changing a parameter value, so that multiple features are automatically altered to be compatible with each other. Inventor also provides an API, compatible with the programming language Python.

The goal for the conceived software was to evaluate different possible spring designs automatically, by altering the design and then simulating its performance using the finite element method. To this end, the parametrizability and API of Inventor were exploited, by developing a program which could access a CAD model and change parameters in it. Unfortunately, the built-in FEM analysis tools can not be accessed from the API. Therefore, an alternative solution for the simulation had to be found. The API allowed for export of STEP files in an ISO standardized format [8], which is widely used and therefore facilitates interfacing with other software.

Conduction of a FEM-based simulation requires the domain (i.e. the spring in this case) to be discretized into elements. This discretisation, or mesh, is described by the positions of the nodes connecting the elements. After this, the mesh and information about the problem such as boundary conditions, material properties has to be provided to a solver, which sets up system matrices representing a linear system of equations. The solution of this linear equation system is a discrete approximation (defined in the mesh nodes) to the solution in the continuous domain.

Meshing

Since Autodesk Inventor's built-in meshing abilities were inaccessible from the API, another solution for discretizing the domain was sought. The requirement was that it should be easily controllable with Python, and preferably open source, since this enables interested users to actually read and understand the code of the program themselves, and in addition it is free.

One program which fulfilled the criteria was GMSH [9]. It provides functionality for geometry creation, meshing and post-processing of result data. The meshing part offers adjustable element size, automatic element quality improvement algorithms and manually specified refinements in regions where the user desires a finer mesh. It also has a powerful API for Python, from which most of the functionality can be accessed. GMSH was therefore chosen for mesh generation in the spring optimisation project. A mesh generated with GMSH is shown in Figure 2.8.

FE equation solving

The requirements on a prospective solver software were similar to those on the meshing software - easy to manage from Python and open source code. A number of candidates were found, e.g. code_aster [10] and GetDP [11]. The choice finally fell on Elmer [12], a Finnish software developed at CSC - IT Center for Science. Elmer distinguished itself by having thorough documentation in English and an easy-to-understand syntax for the input files, which contain problem information such as boundary conditions. It also came with example implementations for linear elastic problems in three dimensions, which facilitated the process of setting up the same kind of simulation for a spring.

Parametric CAD model

In order to successfully run the planned program, a properly parametric CAD model was necessary. It is a non-trivial task to set dimensions and positioning of features so that the model can tolerate parameter changes while retaining the original design intent. If it is done incorrectly, parameter changes often result in the CAD software not being able to execute the design change, or it gives unusable designs, as can be seen in Figure 2.9.

If the intended use of a CAD model does not include changing parameters, it is not necessary to make it parametric. Since the need for parametric models at the company had been limited before this, the existing CAD models were not stable with respect to parameter changes. Therefore, new CAD models had to be developed for use with the planned software.

Optimisation

The first goal of the software was to enable the user to run parameter sweeps, where the user specifies which design parameters to vary, which range each parameter is going to be in, and how many parameter values to sample in each range. The parameter values were planned to be evenly spaced in the ranges. Simulations are then performed on every parameter combination, to find and minimise stresses and stiffnesses. Parameter sweeping can be used as a method for optimisation, and has the advantage of giving detailed information about spring properties for many different parameter combinations, but also requires very many simulations, of which many are on springs with incorrect stiffnesses, and it's therefore computationally very heavy. In order to implement

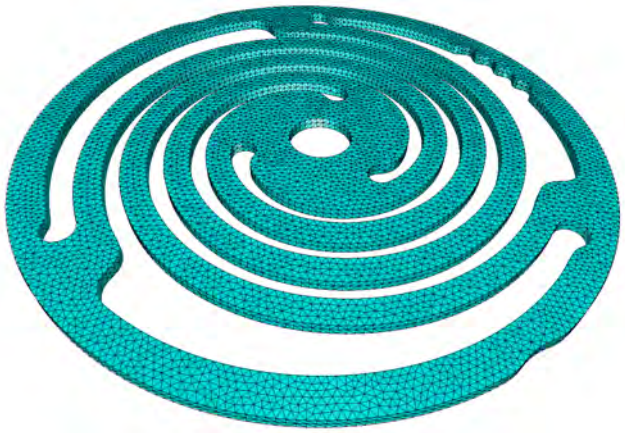
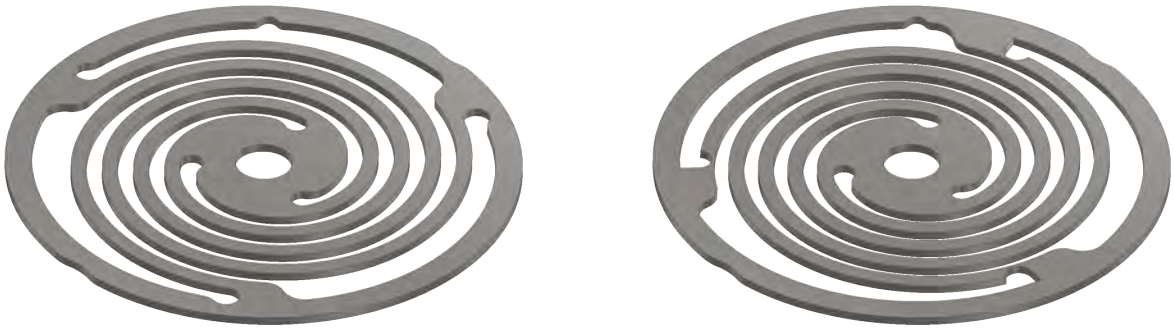


Figure 2.8: *Mesh of harvester spring. The spring has been discretized into elements with GMSH.*



(a) *Original design.* (b) *Unusable design caused by parameter change.*

Figure 2.9: *Improperly parametric CAD model of spring. The figures show the same spring model, with different parameter settings. Note how the spring changes from a usable design with smooth inner radii, to sharp internal corners and very thin spring arms where they attach to the outer ring.*

more sophisticated optimisation in the developed software, the Python package `scipy.optimize.minimize` [13] was used. This can utilise a variety of different optimisation algorithms, for both constrained and unconstrained optimisation. The problem of finding the parameter combination resulting in the lowest possible stress, while maintaining a set stiffness, as posed in Equation (1.4), is an example of a constrained problem. The algorithm SLSQP, Sequential Least-Squares Programming [14], was tested on the stress minimisation problem.

3 Results and Discussion

3.1 Spring stiffness experiment and simulation

Load-displacement data obtained in the spring stiffness experiment is plotted in Figures 3.1, 3.2 and 3.3. Simulation-generated data from Nastran In-CAD is included for comparison. There are considerable differences in stiffness between individuals of the same nominal spring model. The maximum between-individuals differences were in the range 2.8 % – 14 %. This suggests that there are inconsistencies in the manufacturing process. The springs are manufactured with water jet cutting of sheet metal by a contract manufacturer.

Simulations consistently predicted significantly higher stiffness than what was measured in experiments. This is most pronounced for the softest spring, 558, where the simulation results were 30 % stiffer than the experimental average, and less so for the stiffest spring, 571, where the simulation predicted 14 % higher stiffness than the experiments. The semi-stiff spring, 567, was in between.

If the elastic properties of the material, or the sheet metal thickness, would be misrepresented in simulations, it would impact both stiff and soft springs to almost the same relative extent, but since the simulation-experiment discrepancies varied between spring models, it was the author's hypothesis that the discrepancies arose from the spring arms having been cut consistently narrower than they were designed to be, and therefore narrower than assumed in the simulations. Such a geometry misrepresentation could give a larger stiffness deviation for springs with nominally narrower arms (soft springs), due to the relative error being larger than for nominally wide spring arms. The between-individuals differences are also unlikely to be caused by errors in elasticity modulus or sheet metal thickness, since each batch of springs are cut from the same sheet of metal, which

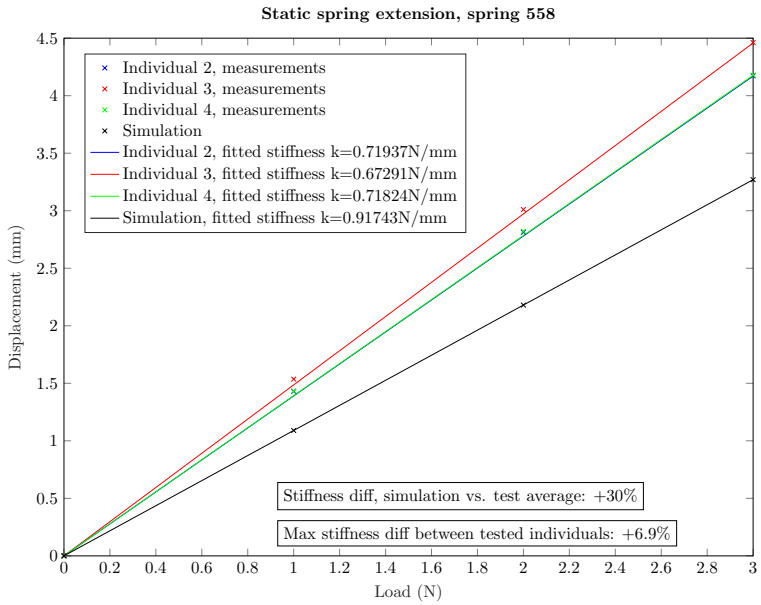


Figure 3.1: *Deflection of the softest spring for model Q, plotted against applied load. Linear functions have been fitted to the data using regression. The slope of these functions are approximations of the spring stiffnesses. The numbering starts with 2 to conform with the numbering in Figure 3.6.*

is likely to be of consistent thickness and material. Therefore, it was believed that also this could be due to insufficient accuracy of the water-cutting process, which is why the experiments described in Sections 2.2.3 and 3.2 were undertaken.

Results indicated that endcap torque, which governs clamping pressure on the spring perimeter, did not influence the measured stiffness very much. All stiffness measurements were grouped according to the torque level with which their endcaps had been fastened, and averaged in each group. The average increase in measured stiffness when going from 2 Nm to 4 Nm was 2.3 %, while there was an average decrease of 0.1 % when going from 4 Nm to 6 Nm. The observed difference between stiffnesses at torques 4 Nm and 6 Nm is so small that it is likely caused by measurement error. Between 2 Nm and 4 Nm torques, the difference is still small but significantly larger, and it is the author's speculation that this could indicate some movement between the spring and the clamping surfaces (cf. Figure 2.2) at 2 Nm. This could be investigated further with more experiments of the same type, possibly with finer resolution in torque e.g. 1 Nm, 2 Nm, 3 Nm and 4 Nm and by doing all experiments on the same spring.

The springs generally exhibit very linear force-displacement behaviour, as assumed. This can be advantageous since it prevents the resonance frequency of the harvester to change with amplitude (cf. Equations (1.2) and (1.3)).

3.2 Spring dimension measurements

The measured spring arm widths are displayed and compared to the nominal dimensions in Figure 3.4. In general, the spring arms are narrower than their intended design – only 6 out of 108 width measurements were at or above the dimensions specified in the spring drawings. The width deficiencies, averaged over all measurements, were $40\,\mu\text{m} \pm 36\,\mu\text{m}$, $40\,\mu\text{m} \pm 25\,\mu\text{m}$, and $35\,\mu\text{m} \pm 19\,\mu\text{m}$ for spring 558, 567 and 571, respectively, where $\pm$ designates one standard deviation.

The thickness of the springs (i.e. the sheet metal thickness) was also measured at around 10 randomly picked points. All measurements showed 0.40 mm, which is exactly according to design. This shows that the spring thickness is likely not the cause of spring stiffness discrepancies, as opposed to the spring arm widths. This result was expected, since the sheet metal forming process often is a large-scale, highly repeatable process with tolerances often in the vicinity of 0.01 mm [15]. In contrast, the water jet cutting of the tested springs

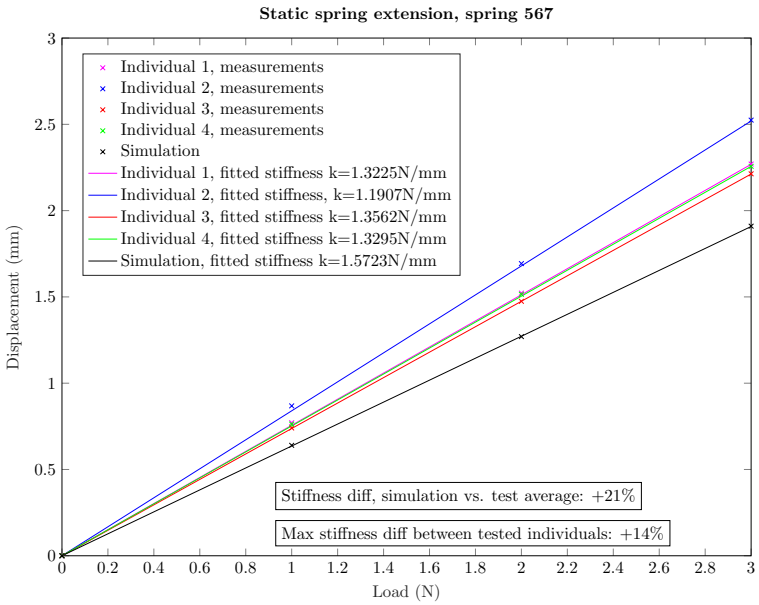


Figure 3.2: *Deflection of a semi-stiff spring for model Q, plotted against applied load. Linear functions have been fitted to the data using regression. The slope of these functions are approximations of the spring stiffnesses.*

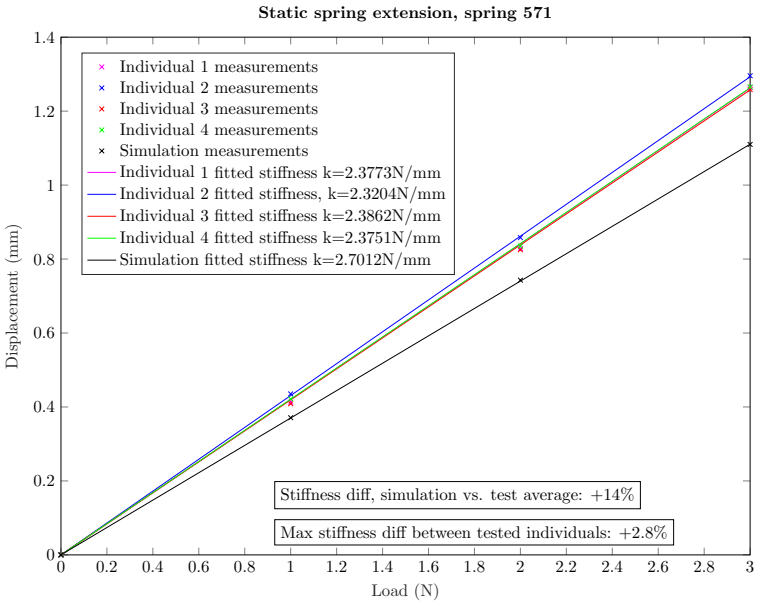
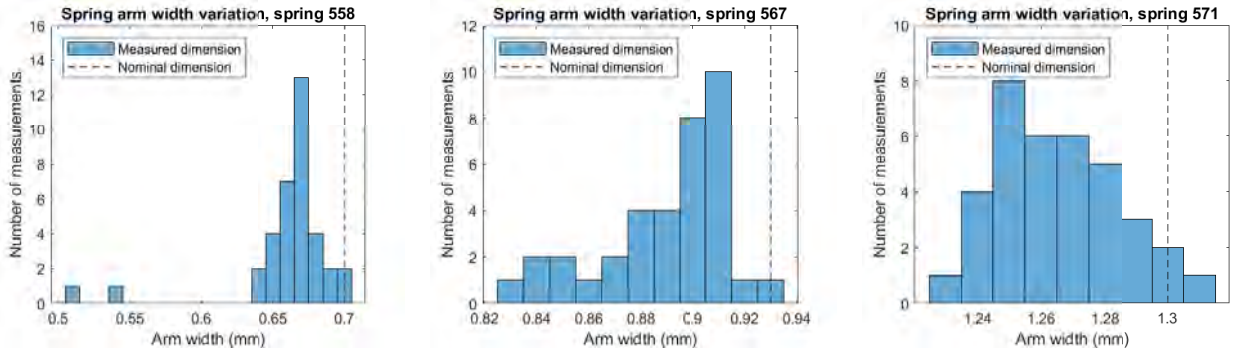


Figure 3.3: *Deflection of the stiffest spring for model Q, plotted against applied load. Linear functions have been fitted to the data using regression. The slope of these functions are approximations of the spring stiffnesses.*

was a small-series production without rigorous quality control. Water jet cutting is done by shooting a jet of water and abrasive grains at the material to be cut. The sheet metal subjected to cutting, is affected by inertial forces from the water. When cutting the long, thin and narrow arms of the springs, such forces can be sufficiently large to deflect the material away from its intended position, causing the water jet to cut wrong. Similar problems had been reported by the contract manufacturer that makes the spring, especially when doing long and narrow spring arms, sometimes causing the cutting operation to completely destroy the spring.

Studying the histogram in Figure 3.4a, there are two width outliers that are significantly narrower than the rest of the measurements. It is the author's belief that these are results of such manufacturing inconsistencies that are described above.



(a) (b) (c)

Figure 3.4: *Histograms displaying results from spring arm width measurements. The dotted lines show the dimensions from the part drawings. Each bin covers 0.01 mm, corresponding to the resolution of the micrometer used in the measurements.*

Table 3.1: Comparison of results from different simulation techniques for the same spring (567), including results from the static spring stiffness experiment in Section 3.1.

Nastran In-CAD	Beam elements in MATLAB	Elmer	Experiment
0.49 mm	0.50 mm	0.48 mm	0.64 mm

3.3 Parametric simulations with beam elements

The implementation of a finite element model using Timoshenko beam equations proved successful. A depiction of the results from such a simulation is shown in Figure 3.5. In this figure, the subdivision of the spring arms into elements is also shown. The model was parametric in the sense that spring arm width, thickness and spiral pitch could be changed by varying numerical parameters.

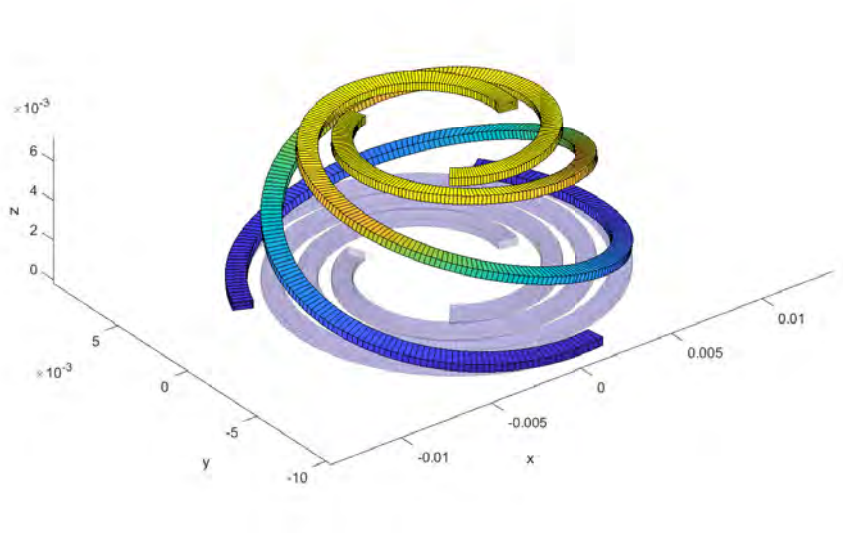


Figure 3.5: *Depiction of results from simulation with Timoshenko beam elements in MATLAB. Note the regular division into elements along the spring arms, as opposed to the unstructured tetrahedron mesh in Figure 2.8.*

Referring to Table 3.1, predicted spring deflections for spring 567 were close to those predicted by finite element calculations in Autodesk Inventor, where the elastostatic (Navier-Cauchy) equations are solved using 3D solid elements, instead of with beam elements. However, beam element FEM has an advantage in being less computationally expensive than 3D elastostatic FEM.

In Figure 3.6, spring stiffnesses from the static load experiment (Section 3.1) are compared to those calculated with the model. The effect of correcting the spring arm widths in the simulation, according to the dimensional measurements from Section 3.2, is also displayed. For the two softest spring models, 558 and 567, simulations overpredict the stiffness of the springs, but this is somewhat mitigated with the corrections for arm width. For the stiff spring, 571, the simulations predict slightly softer behaviour than the experiments show, and the fit is therefore slightly worsened with the arm width corrections. Despite this, spring 571 has the best simulation-experiment correspondence among the springs.

The arm width corrected simulations give slightly different results for each spring individual, since the measured widths vary between them. Still observing Figure 3.6, the arm width corrected simulations show between-individuals differences that are very similar to those from static load experiments. This indicates that a majority of the between-individuals variations, also seen in Figures 3.1, 3.2 and 3.3, is caused by arm width variations. The average stiffness overprediction of simulations, compared to experiments, was reduced by half by applying arm width correction.

The residual error might partly be the result of incorrect material data. The steel in the springs was reported to comply with the standard EN 1.4310. It is the author's observation that the nominal material properties of this steel vary slightly from manufacturer to manufacturer, the Young's modulus is, for example, given as 195 GPa from Lamineries Matthey SA [16] and 190 GPa from Alloy Wire International [17]. The supplier of the steel in the springs, and therefore also the nominal material properties, is unfortunately unknown.

3.4 Parametric simulations in Python

The developed parameter sweep software worked as intended. It was tested on a certain harvester spring, numbered 776. The program provided a spring geometry which maintained the spring's stiffness, while reducing the maximum von Mises-stress in the spring during operation by 19 %. This increased the maximum deflection amplitude the spring could sustain, without exceeding the fatigue stress limit of the steel, by 23 %. In Figure 3.7, a schematic for how the program works is shown.

To facilitate parameter sweep set-up and results interpretation, a simple graphical user interface (GUI) was developed for the program. The parameter sweep initiation interface in the GUI can be seen in Figure 3.8. The GUI allows the user to load a CAD-file, presents the parameters from the CAD-file to the user, and makes the user choose which parameters to sweep. Then, the program prompts for lower and upper limits of parameter values, and number of sweep points, before starting the parameter sweep run.

When running a parameter sweep, a result file is created. This contains information about which CAD-file has been used, what the non-swept parameters were set to at the time of sweeping. The main results values from each simulation in the sweep, i.e. stiffness and maximum von Mises-stress, are then appended to the file together with that simulation's parameter settings, and mesh size (which might vary between runs – cf. Section 3.4.1).

The post-processing part of the program allows for visualisation of the data from the result file. It can automatically produce surface plots like those in Figure 3.9, where a line indicates which parameter combinations yield the desired stiffness, and a star shows which of these combinations gives the lowest von Mises-stress, while the numerical parameter and stress values are provided to the user in a pop-up window.

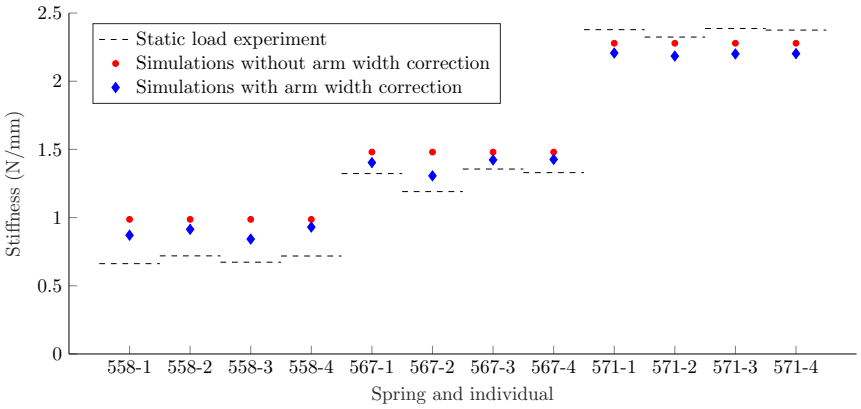


Figure 3.6: *Comparison between spring stiffnesses measured in the static loading experiment, and stiffnesses calculated with the Timoshenko beam model, both with and without arm width correction. The average deviation from experimental results is 18 % without arm width correction and 9,4 % after the correction, an error reduction of 48 %.*

3.4.1 Notable issues encountered during development

Shear locking

When the program development had reached the stage of testing the implemented simulation setup procedure and its solution in Elmer, the simulation output was compared to results from the same type of simulation in Autodesk Nastran In-CAD, and data from the experiments described in Section 3.1 since the first simulations were conducted on the same type of spring. It quickly came to the author's attention that the displacements were very underpredicted by Elmer. After some testing, it became evident that the mesh sizing also heavily influenced the result, with a finer mesh resulting in larger displacements. Convergence with respect to mesh size did not seem possible, as the solver ran out of its allocated memory in the computer before the mesh was fine enough to not influence the result.

Research was done and different alterations to the simulation setup were tested. It was found that a common reason for underprediction of deformations is element locking. This problem can appear when meshing with first-order elements. It especially affects simulations where bending deformation is dominating [18]. The arms of the springs studied in this project are subjected to significant bending, and it is therefore likely that locking

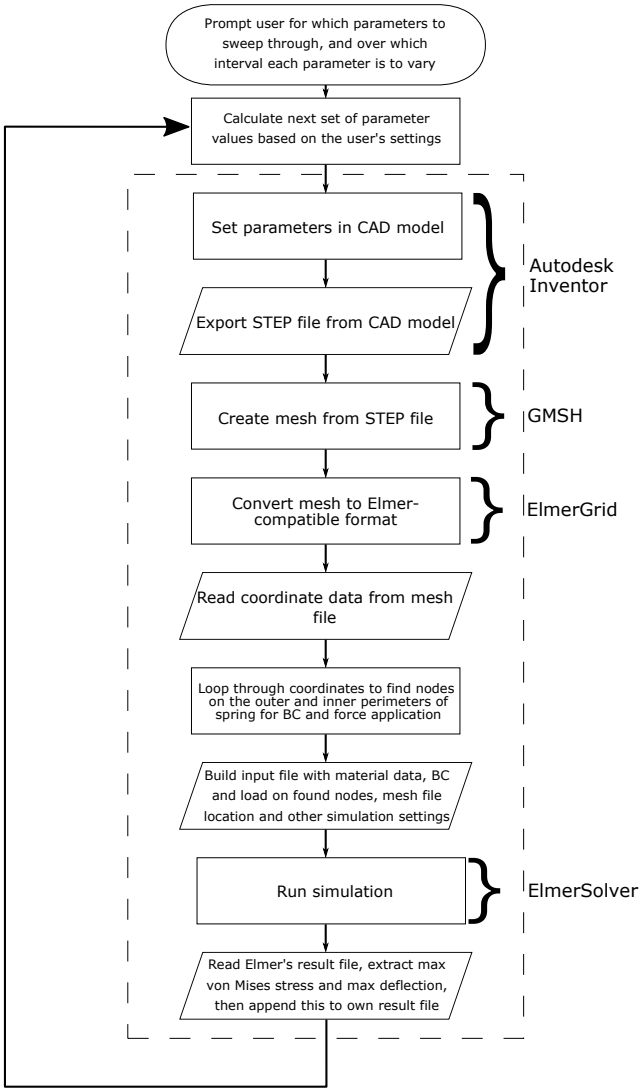


Figure 3.7: *Flowchart describing the algorithm for parameter sweep simulations. The dashed box encapsulates the actual component generation–meshing–simulation–evaluation procedure, which can be wrapped by both a sweeping procedure, as in the figure, or an optimisation algorithm, which decides what parameter values to try.*

can impact the result of the simulations. After switching to second-order elements in GMSH, the problem was resolved and even simulations on coarse meshes gave results close to those of Autodesk Nastran In-CAD.

Mesh limitations

Although the deflection results were accurate with coarse meshes after implementing second-order elements, a fine mesh is still advantageous for accurately predicting local stresses e.g. around the spring arm ends. There, the gradients of the variables of interest (displacement and stress) are large. In Elmer, the solution procedure that seemed to work best was a direct method that uses the UMFPACK software routines [19]. Iterative methods were tried, but they displayed difficulties in converging. There are advantages of using a direct method, e.g. high numerical accuracy, meaning the result obtained is an exact solution (up to the floating point accuracy of the machine) of the discrete problem formulation on the meshed domain. Direct methods are also robust. However, the memory requirement for using such a method grows quickly with problem size [20]. Problems with memory shortage were sometimes encountered when fine meshes were used. The automatic parameter alteration process results in slightly different components, and subsequently meshes, every time a different set of design parameters are tested. It was therefore not always evident whether the mesh setting was too fine at the start of a parameter sweep, because errors could arise far into the sweep run, when the spring assumed some certain combination of parameters. To mitigate this issue, functionality for automatically making the mesh more coarse by increasing the characteristic element length (i.e. the approximate dimension across an element in Figure 2.8) if the simulation failed was implemented.

3.4.2 Optimisation

In Figure 3.9, some sharp discontinuities can be seen in the stress surface. Each point on the surface corresponds to a simulation with its own parameter settings and a unique mesh. The finite element method only gives an approximation to the deformations and stresses in a body, the error of which depends (among other things) on the mesh. When performing finite element analysis, it is therefore important to ensure that the results don't change drastically for slightly different meshes. Since the mesh is different every run, and no check for ensuring mesh convergence is done in each step, the result is sometimes less precise, which gives rise to the "numerical noise" which can be seen as small peaks in the figure.

Attempts were made to implement more sophisticated optimisation than the brute-force method of parameter sweeping. The algorithm SLSQP was tested. Unfortunately, the attempts proved mostly unsuccessful in finding good parameter combinations. It is the author's belief that the non-smooth objective function, discussed in the

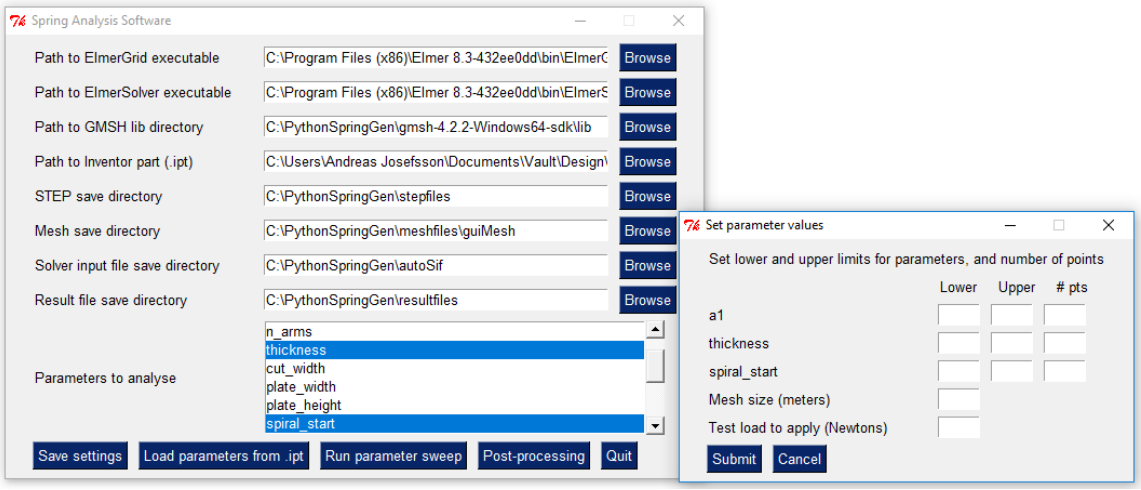


Figure 3.8: *The simulation automation software's main window, together with the parameter sweep setting window, which is shown after the user has selected parameters to sweep through and clicked "Run parameter sweep".*

previous paragraph, may have contributed to this. Such noise in the objective function makes it difficult for the algorithm to reliably evaluate the gradient of the function, making it very difficult to find a good search direction.

Another constraining factor was the computational demand of running such optimisation. Each evaluation of the objective function, or the constraint function, resulted in a simulation cycle (i.e. what is inside the dashed box in the schematic in Figure 3.7), requiring roughly 20 s – 2 min. Due to this, it proved slow and tedious to test different settings for the optimisation algorithm, or different algorithms. In order to not exceed the project's time limitations, it was therefore decided to use the time remaining, after the development of the program was done, to run parameter sweeps on springs, which also gave a better understanding of qualitative properties of the objective and constraint function, e.g. the unevenness in the stress surface.

3.4.3 Parametric CAD model

The CAD-model developed for use with the optimisation software is seen in Figure 3.10. It is stable with respect to parameter changes for the spring pitch, number of arms, thickness, spiral terminator radius, and start- and end radii of the spring arm pattern. This was achieved through extensive use of equations to position and size features, taking great care to make sure they were compatible and working as intended no matter what values the parameters were set to.

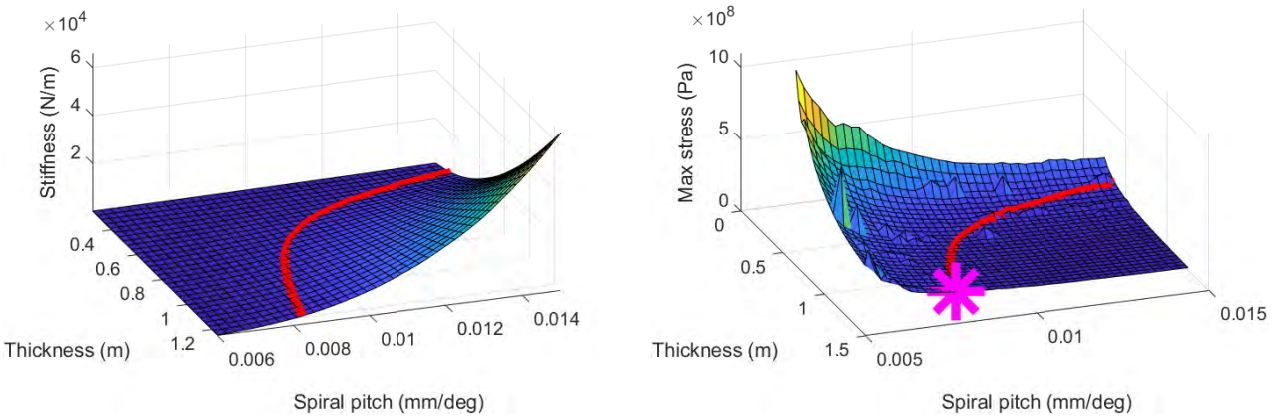
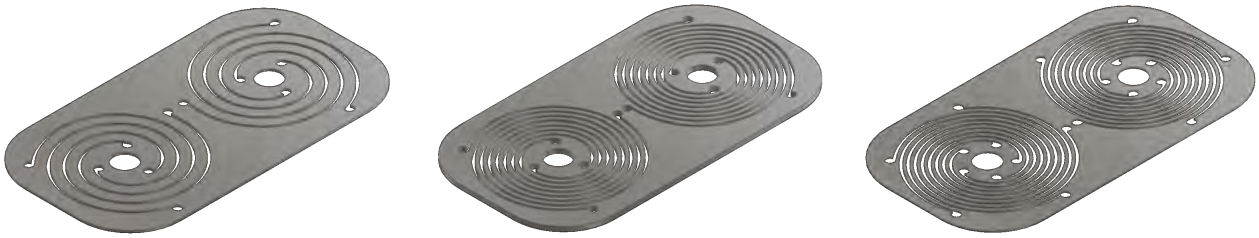


Figure 3.9: *Spring stiffness and maximum von Mises-stress in a spring when a load of 1 N is applied to it. The parameters on the base axes are demonstrated in Figure 3.10. The red line is a contour line in the stiffness plot, representing a desired stiffness for a certain harvester. The same line is projected on the stress plot, where the stress minimum along the line is shown with a pink marker.*



(a) *3 arms per spiral, 0.009 mm/deg spiral pitch and 0.3 mm thickness.*

(b) *3 arms per spiral, 0.006 mm/deg spiral pitch and 0.6 mm thickness.*

(c) *5 arms per spiral, 0.009 mm/deg spiral pitch and 0.2 mm thickness.*

Figure 3.10: *Parametric CAD model of spring. The figures show the same spring model, with different parameter settings. The parameters that are altered are the spiral pitch (i.e. how tightly the spring is wound), thickness and number of arms.*

4 Concluding remarks and future work

4.1 Spring manufacturing process

The inaccuracies in the manufacturing process discovered in Section 3.2 is likely to lead to inconsistencies in resonance frequency. It can also result in higher risk of failure due to uneven load distribution in the spring arms. This is especially disadvantageous in large-scale production, since it can lead to quality problems or need of extensive, and possibly time-consuming, testing. It is therefore desirable to mitigate this issue.

Each spiral-shaped cut in the spring (i.e. the separations between the spring arms) had been made in two passes, since the width of the cut is larger than the diameter of the water jet. If the width of the cut is instead designed to be equal to the diameter of the cutting jet, each spiral cut could be made in one pass, possibly reducing the risk for inadvertent in-plane movement of the spring during cutting, which could increase the accuracy and repeatability. This was discussed with the contract manufacturer, and found to be a promising idea.

It would also be possible to manufacture the springs with another process, e.g. etching or blanking. More testing would be required to determine if these could be feasible options.

4.2 Resonance frequency error

After it was discovered that there were significant errors in spring stiffness, a lot of time was dedicated to investigating and trying to mitigate that issue. It is the author's belief that most of the error in eigenfrequency prediction results from the spring stiffness error, since the observed eigenfrequency errors were roughly similar in magnitude, or smaller, compared to the measured stiffness errors, and the eigenfrequency error should be linearly related to the square root of the stiffness error, see Equation (1.2). However, more rigorous testing and comparisons are required to more accurately characterise errors arising from elsewhere, e.g. component weights and magnetic interactions with surrounding ferromagnetic materials.

4.3 Continued use of parameter sweep software

The developed software has the potential to easily evaluate large numbers of possible spring designs. To fully utilise this potential, and gain better understanding of how it is best used, it would be beneficial to work more with the software. Conducting parameter sweeps with different parameters and ranges gives information about how the parameters in a certain CAD-model affect each other, and what parameter combinations are feasible and likely to result in good performance.

It could also be interesting to test new spring concepts using the software by creating new, parametric CAD-models, and compare them between each other. While the software can change design parameters in a CAD-model freely, it is limited to changing parameters that the designer has created, still requiring creativity and logic from the user to make the most of it.

4.4 Improving computational efficiency

In its current state, the software generates a homogeneous mesh. It would likely be possible to increase result accuracy, without significantly increasing computation time, by using a finer mesh in regions of high gradients. If a coarser mesh would be applied to regions of low gradients, e.g. the perimeters of the springs, computational cost could even be reduced.

The springs have a rotational symmetry thanks to their arms. This symmetry could be used to increase the efficiency of computation, by only simulating one of the symmetric parts. This would require a CAD-model

containing only one of the arms and the pertinent circular sector of the spring, and appropriate boundary conditions.

Elmer supports parallel computing for solving the FEM equation system using message passing interface, MPI [21], which enables communication between different processes in software. With this, it is possible to use a parallel solver such as MUMPS [22], which is a parallel direct solver for sparse matrices. Parallel computing could significantly decrease execution time of the optimisation software, since solving the linear system of finite element equations is the most time-consuming part of the software. To enable parallel computing, it is required to compile and configure Elmer on the machine that it is going to run, while the non-parallel version could be run from pre-compiled executable files. In order to reach a usable software as soon as possible, the non-parallel version was used in this project.

4.5 Optimisation

To mitigate the issues with objective function non-smoothness mentioned in Section 3.4.2, it could prove useful to create a metamodel, e.g. by fitting a polynomial function to the objective and constraint functions. This can be done by sampling the true objective function at a few selected points in the design space, and minimising the error between e.g. a second-order polynomial and the objective function in those points in a least-squares sense. This provides a smooth approximation of the function, which additionally is very quick to run, which allows for rapid testing of different optimisation algorithms and settings for them [23].

It would also improve the computational efficiency of optimisation to incorporate the Timoshenko beam element simulation procedure (Sections 2.3.1 and 3.3) for predicting the stiffness of the springs, instead of running the full 3D elastostatic analysis, when evaluating the constraint function. This is much quicker to run, around 1 s compared to around 45 s, which was typically required for the full 3D analysis.

Another potential solution for removing the discontinuities in the objective function, could be to implement automatic mesh convergence check at each simulation step. For example, the same simulation could be run multiple times with slightly different mesh sizes, and the maximum stresses in each mesh compared, followed by a check so that the result does not vary significantly between the different meshes. However, this would further increase computational cost, and is therefore probably not a good solution unless computational efficiency of the software is improved first.

4.6 Other applications for parameter optimisation software

The developed software performs mechanical simulations, and is suitable for optimising components with respect to mechanical performance. With minor modifications to the code, it could be used for minimising or maximising variables other than stress, and with constraints other than stiffness. Possible objective functions and constraints could also, for example, be deflection, weight or moment of inertia.

However, the finite element solver software, Elmer, also has capabilities to solve other equations with FEM, such as Maxwell's equations, that govern electromagnetics. In the development of electromagnetic energy harvesters, it could be useful to simulate and optimise performance of the magnet-coil system. With some modifications to the developed software, it would be possible to do such optimisation, for e.g. choosing dimensions and positions of coils and magnets.

References

[1] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson. The industrial internet of things (IIoT): An analysis framework. *Computers in Industry* **101** (2018), 1–12. ISSN: 0166-3615. DOI: https://doi.org/10.1016/j.compind.2018.04.015. URL: http://www.sciencedirect.com/science/article/pii/S0166361517307285.

[2] D. Spreeman and Y. Manoli. *Electromagnetic Vibration Energy Harvesting Devices*. Springer Science+Business Media B.V., 2012. ISBN: 978-94-007-2943-8.

[3] R. R. C. Jr. and A. J. Kurdila. *Fundamentals of Structural Dynamics*. John Wiley & Sons, 2006, p. 56. ISBN: 978-0-471-43044-5.

[4] P. Y. Papalambros and D. J. Wilde. *Principles of Optimal Design. Modeling and Computation*. 2nd ed. Cambridge University Press, 2000.

[5] *Inventor Product Overview*. Autodesk. URL: https://www.autodesk.com/products/inventor/overview (visited on 2019-04-04).

[6] *optoNCDT 1420 Product Description*. Micro-Epsilon. URL: https://www.micro-epsilon.co.uk/displacement-position-sensors/laser-sensor/optoNCDT_1420_basic/ (visited on 2019-03-04).

[7] J. S. Przemieniecki. *Theory of Matrix Structural Analysis*. Dover Publications, 1968, p. 79.

[8] *ISO 10303-21:2016. Industrial automation systems and integration – Product data representation and exchange – Part 21: Implementation methods: Clear text encoding of the exchange structure*. International Organization for Standardization. URL: https://www.iso.org/standard/63141.html (visited on 2019-04-04).

[9] C. Geuzaine and J.-F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *Int. J. Numer. Meth. Engng* **79** (2009), 1309–1331.

[10] *code_aster*. Électricité de France. URL: https://code-aster.org/ (visited on 2019-04-10).

[11] P. Dular and C. Geuzaine. *GetDP. General Environment for the Treatment of Discrete Problems*. URL: http://getdp.info/ (visited on 2019-04-10).

[12] *Elmer*. URL: https://www.csc.fi/web/elmer/elmer (visited on 2019-04-10).

[13] *SciPy v1.2.1 Reference Guide. scipy.optimize.minimize*. SciPy. URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html.

[14] D. Kraft. A Software Package for Sequential Quadratic Programming (1988). URL: http://degenerateconic.com/wp-content/uploads/2018/03/DFVLR_FB_88_28.pdf (visited on 2019-05-17).

[15] *Sandvik: Size tolerances for strip steel*. URL: https://www.materials.sandvik/en/products/strip-steel/size-tolerances/ (visited on 2019-05-15).

[16] *Lamineries Matthey SA: Steel 1.4310 data sheet*. URL: https://www.matthey.ch/fileadmin/user_upload/downloads/fichetechnique/EN/1.4310_C.pdf (visited on 2019-05-22).

[17] *Alloy Wire International: Stainless Steel 1.4310*. URL: https://www.alloywire.com/products/stainless-steel-1-4310/ (visited on 2019-05-22).

[18] E. Wang, T. Nelson, and R. Rauch. Back to Elements - Tetrahedra vs. Hexahedra (2004).

[19] T. A. Davis. Algorithm 832: UMFPACK V4.3—an Unsymmetric-pattern Multifrontal Method. *ACM Trans. Math. Softw.* **30**.2 (June 2004), 196–199. ISSN: 0098-3500. DOI: 10.1145/992200.992206. URL: http://doi.acm.org/10.1145/992200.992206.

[20] E. Agullo, P. Amestoy, A. Buttari, A. Guermouche, G. Joslin, J.-Y. L'Excellent, X. S. Li, A. Napov, F.-H. Rouet, M. Sid-Lakhdar, S. Wang, C. Weisbecker, and I. Yamazaki. "Recent advances in sparse direct solvers". *22nd Conference on Structural Mechanics in Reactor Technology (SMIRT 2013)*. San Francisco, US: Elsevier, Sept. 2013, pp. 1–10. URL: http://oatao.univ-toulouse.fr/15162/.

[21] *MPI: A Message-Passing Interface Standard Version 3.1*. Message Passing Interface Forum. June 2015. URL: https://www.mpi-forum.org/docs/mpi-3.1/mpi31-report.pdf.

[22] *MUMPS: MUltifrontal Massively Parallel sparse direct Solver*. URL: http://mumps.enseeiht.fr (visited on 2019-05-07).

[23] G. G. Wang and S. Shan. Review of Metamodeling Techniques in Support of Engineering Design Optimization. *Journal of Mechanical design* (2006). URL: http://www.sfu.ca/~gwa5/index_files/Microsoft%5C%20Word%5C%20-%5C%20ASME-Metamodeling-6-12-06.pdf (visited on 2019-05-17).