

**Department of Precision and Microsystems Engineering**

**Compliant mechanism based nonlinear spring design  
for inducing mode coupling**

Haolang Ding

Report no	: 2023.032
Coach	: Davood Farhadi
Professor	: Farbod Alijani
Specialisation	: Dynamics
Type of report	: Master thesis
Date	: 1 June 2023



# Compliant mechanism based nonlinear spring design for inducing mode coupling

---

Master thesis submitted to Delft University of Technology  
in partial fulfilment of the requirements for the degree of

**MASTER OF SCIENCE**

in **High Tech Engineering**

Faculty of Mechanical, Maritime and Materials Engineering

by

Haolang Ding

Student number: 5318580

To be defended in public on June 1st 2023

**Graduation committee**

Dr. F. Alijani, TU Delft, supervisor

Dr. D. Farhadi, TU Delft, supervisor

Z. Li, TU Delft

Dr. A. hunt, TU Delft



# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Project background . . . . .	3
1.2	Thesis objective . . . . .	4
1.3	Overview . . . . .	5
<b>2</b>	<b>Literature review</b>	<b>6</b>
2.1	Abstract . . . . .	6
2.2	Introduction . . . . .	6
2.2.1	Energy harvest . . . . .	6
2.2.2	Sensing . . . . .	9
2.3	Compliant mechanism . . . . .	11
2.3.1	Rigid body replacement . . . . .	13
2.3.2	Pseudo-Rigid-Body (PRB) model . . . . .	14
2.3.3	Structural optimization . . . . .	14
2.3.4	An example of structural optimization . . . . .	16
2.3.5	Flexible translational joints . . . . .	18
2.4	Discussion . . . . .	18
<b>3</b>	<b>Paper</b>	<b>22</b>
3.1	Introduction . . . . .	23
3.2	Nonlinear spring design methods . . . . .	24
3.2.1	Linear stage design . . . . .	24
3.2.2	Lumped model verification . . . . .	26
3.2.3	Design of spline-shaped mechanism nonlinear spring . . . . .	30
3.2.4	Nonlinear spring based on crack-slider mechanism arrangement . . . . .	37
3.3	Nonlinear spring design embodiment . . . . .	47
3.3.1	Result of spline-shaped mechanism nonlinear spring . . . . .	47
3.3.2	Result of crank slider mechanism nonlinear spring . . . . .	50
3.4	Mode coupling . . . . .	54
3.4.1	Equation of motion . . . . .	54
3.4.2	Numerical simulation . . . . .	56
3.5	Discussions and conclusions . . . . .	60
3.5.1	Challenges . . . . .	60
3.5.2	Conclusions . . . . .	64

<b>Appendices</b>	<b>67</b>
<b>A Prototyping</b>	<b>67</b>
A.1 3D printed prototype . . . . .	67
A.2 Laser cut prototype . . . . .	69
<b>B Experimental validations</b>	<b>74</b>
B.1 3D printed prototype . . . . .	74
B.2 Laser cut prototype . . . . .	76
B.3 Prospects . . . . .	77
<b>C Potential nonlinear spring designs</b>	<b>77</b>
<b>D Nonlinear spring optimization scripts</b>	<b>80</b>
<b>Bibliography</b>	<b>99</b>

# 1 Introduction

## 1.1 Project background

Mode coupling behaviour, where the energy transfer between different modes, can be seen in many different places, from the torsion pendulum that measures forces through the coupling of swing and torsional modes [1], to micro-scaled angular rate sensors utilizing the coupling of the gyroscope's sense and drive modes. For a multi-degree-of-freedom nonlinear system, if the ratio of its components' natural frequencies is very close to an integer, the frequency of the system's different vibration mode can be coupled at certain internal resonance ratio [2, 3]. When the system includes nonlinear terms, through appropriate parameter tuning, we can induce coupling between different modes. This phenomenon of stronger energy transferring through mode coupling is called internal resonance [4]. These features can be used in multiple fields, such as vibration energy harvesting and sensing.

Low-power microelectronic products are widely used in consumer electronics, medical equipment, aerospace and other fields [5]. Harvesting energy from the nearby of the devices has received more attention, and the harvested energy can be used as a supplement or replacement for traditional batteries [6]. In most cases, the input environmental vibration energy is distributed in a wide frequency range or direction. Through proper design, the internal resonance or mode coupling can be used to realize the energy conversion between different modes (with different direction or matching frequency), so as to realize higher harvest efficiency or the energy harvesting of multiple vibration modes by single transducer [7, 8, 9]. In order to monitor or measure mechanical, optical, acoustic, electromagnetic and other signal in the environment, sensors are widely used in microelectromechanical systems [10]. Mode coupling is also widely used in this field, and they can improve the sensitivity and bandwidth of sensors with relatively higher quality factor [11].

Compliant mechanism design has emerged as a valuable technique in mechanical engineering for enabling mode coupling. Compliant mechanisms are characterized by their ability to transfer motion, force, or energy through the deflection of elastic flexure joints, which are typically made from elastic materials. These mechanisms offer several advantages, including customizable motion, high precision, lightweight construction, low friction, reduced wear and tear, ease of manufacturing, and most notably, compactness [12].

These inherent characteristics make compliant mechanisms highly suitable for a range of applications, including microsystems, robotics, biomedical implants, and various other fields. The

ability to achieve compact designs while maintaining desired functionality makes compliant mechanisms particularly attractive in scenarios where space is limited or weight reduction is critical. The use of compliant mechanisms can lead to improved performance, increased reliability, and enhanced efficiency in a wide range of engineering applications.

While mode coupling has been extensively researched and applied in the field MEMS, the precise design and implementation of compliant mechanisms for achieving specific mode coupling and energy transfer remains an ongoing area of study.

## 1.2 Thesis objective

In the realm of microelectromechanical systems (MEMS), achieving mode coupling between resonators often involves employing optical or electrostatic actuation methods to couple one system to another[13, 14]. However, recognizing the advantages of compliant mechanisms in terms of system complexity, processing, and maintenance, utilizing mechanical elements to facilitate mode coupling is proposed. One such element is a nonlinear spring that exhibits an approximately cubic relationship between load and displacement.

Since this concept is still relatively unexplored, the focus lies in proving the feasibility of the idea rather than designing and fabricating a model for a specific engineering application. This thesis aims to conduct preliminary verification and exploration of the concept, which entails establishing mathematical models, performing dynamic calculations, designing physical parameters for each component, creating computer-aided design (CAD) models, conducting finite element simulations, exploring viable processing methods (such as 3D printing and laser cutting), and performing experiments to compare the mechanical properties of processed samples with theoretical design parameters.

The primary objective of this thesis is to establish a connection between mode coupling and compliant mechanisms. To achieve this, the goal is to design a simple compliant mechanism wherein an initial input of energy to the system allows for the transfer of energy back and forth between two modes. The system comprises two mass blocks, two linear springs connecting the blocks to the ground, and a nonlinear spring connecting the two blocks. The emphasis of the design is placed on the nonlinear spring component. It is crucial that the load-deformation relationship of the spring approximates a cubic curve while minimizing the presence of a linear term, thus facilitating improved mode coupling within the system.

### 1.3 Overview

Chapter 1 is the introduction of the entire final thesis.

Chapter 2 of this thesis is the literature review. It has two main sections. The first part is mainly about the application of mode coupling and internal resonance in the field of MEMS. The second part mainly describes the design method of the compliant mechanism, and nonlinear spring design.

Chapter 3 is a paper concisely describing the work in this thesis. It includes the math analysis, CAD design, and FEM simulation.

Additional information supporting the Chapter 3 paper can be found in Appendices. Appendix A provides information on the linear stage prototyping. Appendix B provides information on the testing of these stages. Appendix C provides some other possible designs for the nonlinear spring that remain further investigation. Appendix D is about the coding of nonlinear spring optimization program.

## 2 Literature review

### 2.1 Abstract

In this report, the application of mode coupling and internal resonance in the micromechanical field is studied and investigated. A simple mechanical structure that can achieve a designed mode coupling using nonlinear spring and compliant mechanisms is proposed, and the equations of motion are developed. The results show that the system can achieve ideal internal resonance ratio when the nonlinear spring has a very small linear term. The design concepts of nonlinear spring are explored, and the literature survey of linear stage which makes the parts move in one direction is also conducted.

### 2.2 Introduction

Mode coupling is a phenomenon in which two or more modes of a system become coupled and affect each other's behavior. This can occur in various physical systems, such as optics, acoustics, electromagnetics, and solid mechanics. The energy transfer between different modes, can be seen in many different places, from the torsion pendulum that measures forces through the coupling of swing and torsional modes [15], to micro-scaled angular rate sensors utilizing the coupling of the gyroscope's sense and drive modes. For a multi-degree-of-freedom nonlinear system, if the ratio of its components' natural frequencies is very close to an integer, the frequency of the system's different vibration mode can be coupled at certain internal resonance ratio [2, 3]. When the system includes nonlinear teams, through appropriate parameter setting, we can induce certain type of coupling. This phenomenon of stronger energy transferring through nonlinear coupling is called internal resonance [4]. These features can be used in multiple fields, such as vibration energy harvesting and sensing.

#### 2.2.1 Energy harvest

Low-power microelectronic products are currently widely used in consumer electronics, medical equipment, aerospace and other fields [5]. Energy storage systems (batteries) are still their most widely used energy source [16]. But batteries will limit these products in terms of size, quality, durability, lifespan, and environmental friendliness. With the development of science and technology, the power consumption of common MEMS devices can be reduced to the level of mW or even  $\mu$ W [17]. As a result, Harvesting energy from the nearby of the devices has received more attention, and the harvested energy can be used as a supplement or replacement for traditional batteries [6]. These energies include thermal energy from human body or devices

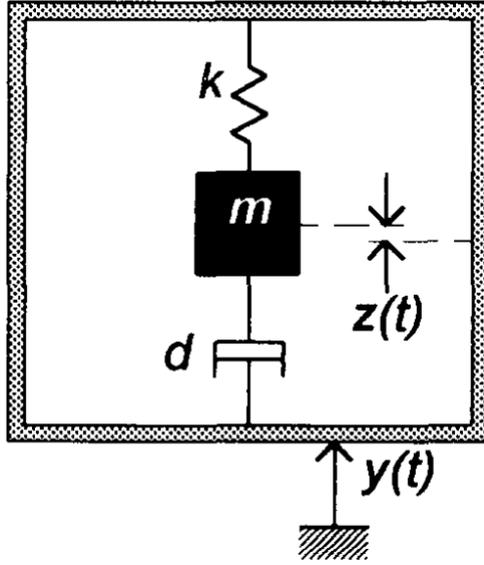


Figure 1: Schematic diagram of the energy harvester [20]

running, solar energy, mechanical energy, etc. Among them, the collection of mechanical energy is cleaner, more stable, and takes up less volume [18]. Piezoelectric, electromagnetic, and electrostatic are the three most common methods of mechanical vibration energy harvesting. The electrostatic method is to convert mechanical energy into electrical energy through the relative movement of the capacitive plates [19].

Williams and Yates [20] proposed a basic mass-spring system concept to convert the mechanic vibration to electricity. Figure 1 shows the schematic diagram of the energy harvester. It consists of a mass block  $m$ , a linear spring  $k$ , and an energy transducer  $d$ , which is depicted as a damper because the process of converting the mass vibration displacement into electrical energy is very similar to the behavior of a damper in the system. The damping coefficient  $d$  satisfies  $d = d_m + d_e$ , where  $d_m$  is the mechanical damping coefficient,  $d_e$  is the electrical damping coefficient. This energy transducer can be based on piezoelectric, electromagnetic and electrostatic. The outside vibration excites the system through the inertial frame. The equation of motion of system is:

$$m\ddot{z}(t) + d\dot{z}(t) + kz(t) = -m\ddot{y}(t) \quad (1)$$

Where  $z(t)$  is the spring deflection, which equals to the distance between mass block and the inertial frame.  $y(t)$  is the input displacement. When applying a vibration of  $y(t) = Y_0 \sin(\omega t)$ ,

the steady state solution of spring deflection becomes:

$$z(t) = \frac{\omega^2}{\sqrt{\left(\frac{k}{m} - \omega^2\right)^2 + \left(\frac{(d_m + d_e)\omega}{m}\right)^2}} Y_0 \sin \left[ \omega t - \tan^{-1} \left( \frac{d\omega}{k - \omega^2 m} \right) \right] \quad (2)$$

The energy extracted by the energy transducer equals to the power absorbed by the electrical damping behaviour [21], which satisfies:

$$p_{\text{out}} = \frac{\frac{d_e}{2\omega_n} Y_0^2 \left(\frac{\omega}{\omega_n}\right)^3 \omega^3}{\left[1 - \left(\frac{\omega}{\omega_n}\right)^2\right]^2 + \left[\frac{d}{m\omega_n} \left(\frac{\omega}{\omega_n}\right)\right]^2} \quad (3)$$

As the formula shown, the maximum deflection of spring will be achieved when the input vibration frequency  $\omega$  equals to the system's nature frequency  $\omega_n = \sqrt{k/m}$ . At this time, the energy transducer  $d$  can absorb the maximum amount of energy:

$$p_{\text{out,max}} = \frac{d_e m^2 Y_0^2 \omega_n^4}{2d^2} \quad (4)$$

According to the formulas above, it can be concluded that if a higher energy absorption value is to be obtained, the vibration frequency received by the energy harvester must be close to its resonant frequency.

Xu and Tang [22] proposed a cantilever-pendulum system achieving multi-directional energy harvesting by one piezoelectric energy transducer through internal resonance behaviour. Figure 2 shows the arrangement of the device. The device includes a cantilever and a pendulum connected to its tip. The pendulum is hung in the gravity direction. There is a piezoelectric transducer installed on the root of the cantilever. The energy of the beam's bending motion will be converted into electricity through the piezoelectric transducer. Figure 3 shows the motion of two major parts. When the pendulum swings, it will generate acceleration in the z direction to drive the vibration of the beam, and if the natural frequency of the pendulum is half that of the beam, resonance of beam vibration will be induced. In this case, the pendulum's oscillation energy will be transferred to beam's bending and be collected.

In most cases, the input environmental vibration energy is distributed in a wide frequency range or direction. To conclude, through proper design, the internal resonance or mode coupling can be used to realize the energy conversion between different modes (with different direction or matching frequency), so as to realize simply higher harvest efficiency or the energy harvesting of multiple vibration modes by single transducer [7, 8, 9].

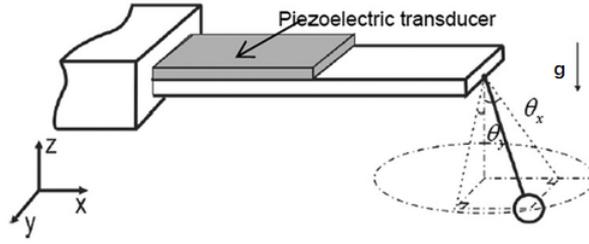


Figure 2: Schematic diagram of the cantilever-pendulum system [22]

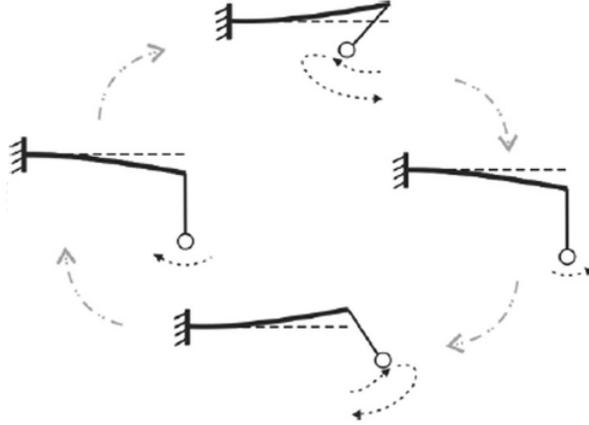


Figure 3: Motion of the system parts [22]

### 2.2.2 Sensing

In order to monitor or measure mechanical, optical, acoustic, electromagnetic and other information in the environment, sensors are widely used in microelectromechanical systems [10]. Mode coupling and internal resonance effects are also widely used in this field, and they can improve the sensitivity and bandwidth of sensors with relatively high quality factor [11].

**Heat sensor** Zhang et al. [23] proposed a highly sensitive thermal sensor using the internal mode coupling effect. When the thermal sensor is heated, the resonant frequency shifts. Internal mode coupling is created using fundamental bending mode and fundamental torsional mode. The coupling is induced by cubic Duffing nonlinearity. As a result, if modulate the input heat at a certain frequency, the frequency shift increased by two orders of magnitude.

**Mass sensor** Mechanical sensors are mainly used for sensing mechanical parameters such as displacement, velocity, acceleration, angular velocity, force, torque, deformation, etc [24].

For example, piezoelectric MEMS resonant sensors, have extremely high resolution and can be used for picogram to femtogram quality inspection [25] in the chemical and biological fields. A quartz crystal microbalance can be considered as an acoustic bulk wave resonator [26]. When

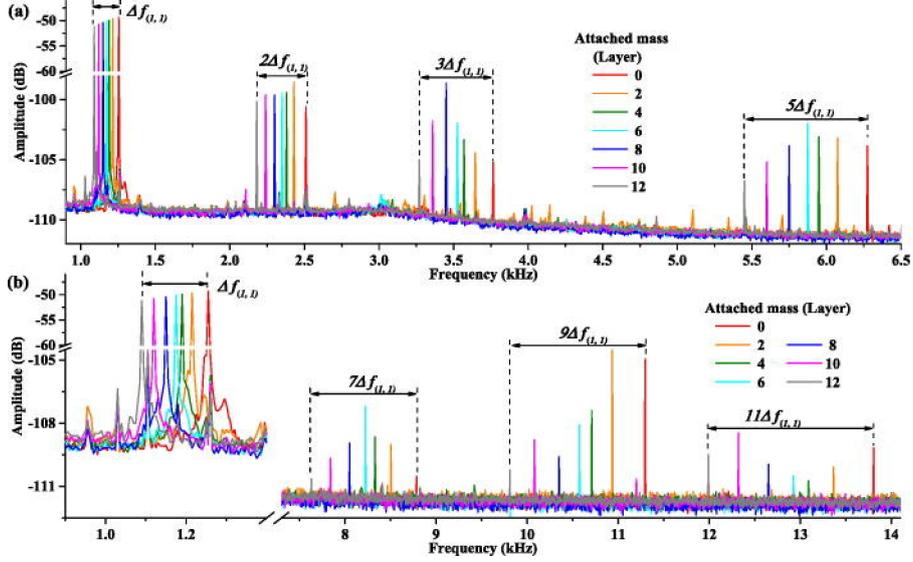


Figure 4: Higher-order modes have a more significant frequency shift [11]

the deposited mass is rigid and regularly distributed, the change of characteristic frequency satisfies a linear relationship with the mass change when the frequency change is below %, as shown by the Sauerbrey equation [27]:

$$\Delta f = -\frac{2f_0^2}{A\sqrt{\rho_q\mu_q}}\Delta m \quad (5)$$

But most research on such mass sensors has focused on linear mass detection. Kirkendall et al [28] proposed a device for mass detection in MEMS systems using internal resonance and mode coupling. When slightly changing the resonator surface mass, a nonlinear amplitude jump will be observed in the device's frequency response curve. Zhang et al [11] used the internal resonance between the fundamental mode and the higher-order mode to amplify the resonance frequency shift by an integer multiple, achieving an 11 times sensitivity amplification. As shown in the Figure 4, when the mass is changed, the higher-order modes have a more significant frequency shift than the fundamental mode.

**Angular rate sensing** Gyroscopes are widely used for angular rate sensing in MEMS [29]. Due to their small size, most MEMS gyroscopes do not use rotor rotation to get data, but instead use the Coriolis effect. In these Coriolis gyroscopes, vibration and rotation-induced Coriolis forces transfer angular velocity energy to the sensing mode. When the natural frequencies of the sensing mode and the driving mode are equal, the energy received by the sensor is the largest, leading to maximum sensitivity. Sarrafan et al. [30] proposed a structure that exhibits both 1/2 subharmonic and 2:1 internal resonance is introduced to tune the sense and drive modes. That can enlarge frequency bandwidth for nonlinear micro-gyroscopes and provide a higher

robustness against fabrication and operation errors. Another H-shaped nonlinear tuning fork micro resonator [31] utilizes a 2:1 internal resonance to eliminate the mode matching problem in a purely mechanical way, reducing the instability and drift caused by cross-coupling. This principle can make the output signal of the gyroscope is only sensitive to the input angular rate, thus improving the robustness of the gyroscope towards mechanical variations.

Venstra et al. [32] proved that flexural vibration modes of clamped-clamped micromechanical resonator can be strongly coupled. In this way, the frequency response of resonator can be easier detected by mechanical sensor. This can avoid the signal loss caused by the parasitic capacitance behavior of electrical detector.

Saad et al. [33] proposed a concept of coupled micro resonator array sensor that use mechanical springs to couple five micro resonators together, as shown in Figure 5. Because the response mode frequencies of the five resonators are different, the states of each resonator can be obtained by only measuring the modal resonance frequency of the output at the end coupled array. When these resonators are mass sensors, the resonator states can be the amount of mass attached on the sensor. In addition, these springs enhance the resonator's response magnitude. It is also possible to change the spring design parameters to better separate modal responses and increase the overall frequency bandwidth.

### 2.3 Compliant mechanism

Compliant mechanisms are the mechanisms that transfer motion, force, or energy through the deflection of flexure joints parts that is made of elastic materials. Compliant mechanisms have a lot of advantages, such as customizable motion, high precision, low weight, low friction, less abrasion, ease of manufacturing, and most importantly, compactness [12]. These characteristics make them very suitable for microsystems, robotics, biomedical implants, and other applications.

However, the motion behavior and mechanical properties of compliant mechanisms become nonlinear when large deflections are involved. Moreover, compliant mechanisms often involve more fragile and slender parts, and the strength of the material must be paid attention to. Therefore, the design of the compliant mechanisms becomes a challenging job. According to a literature review by Juan A. Gallego and Just Herder [34], the design methods of compliant mechanisms can be divided into three categories: the kinematics-based approach, the building blocks approach and the topology optimization approach, as shown in Figure 6. Other classifications of designing approaches include: finite element analysis, topology optimization, and



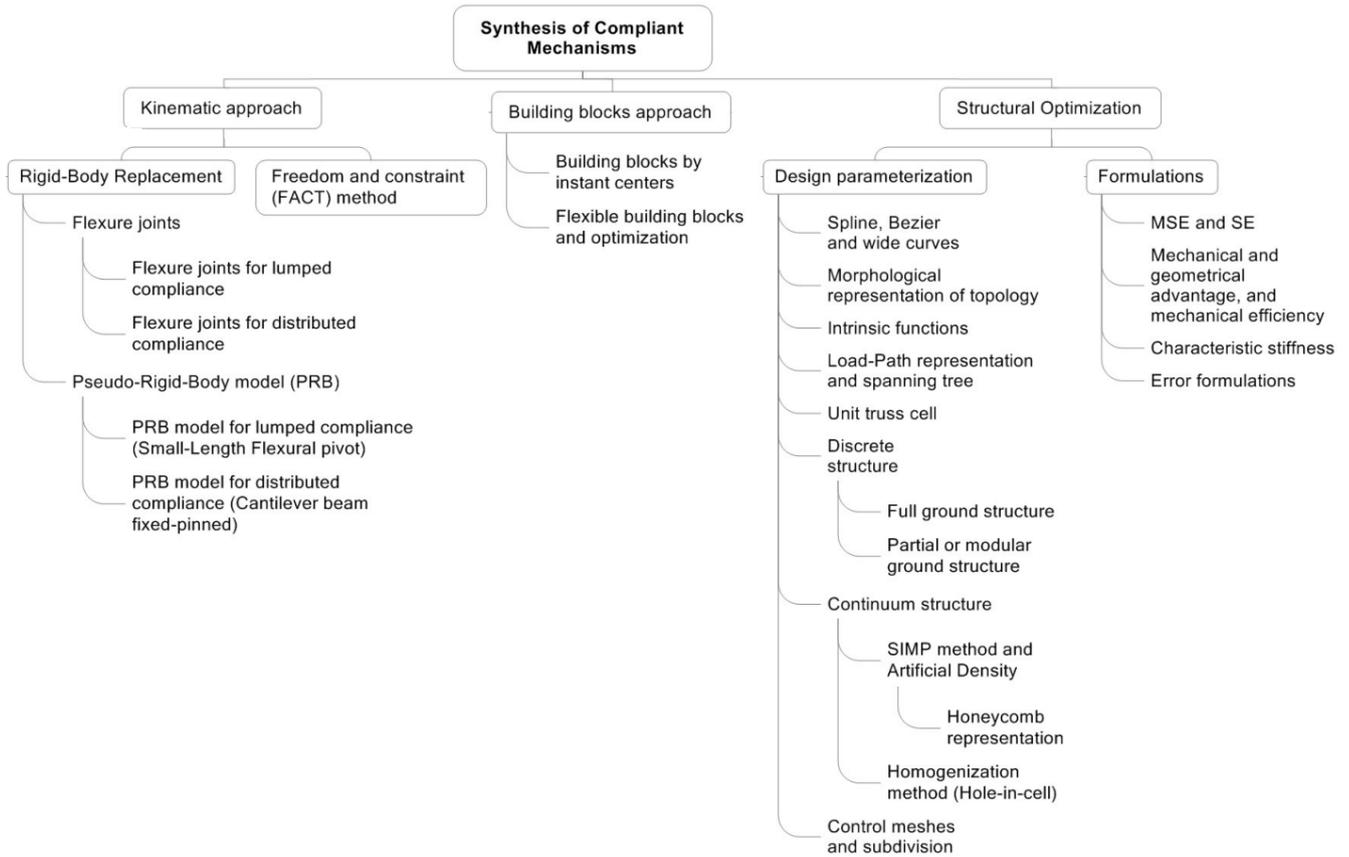


Figure 6: Synthesis of compliant mechanisms [34]

pseudo-rigid-body model (PRBM) [12].

### 2.3.1 Rigid body replacement

Rigid body replacement method is one of kinematic approaches. In short, the core of this method is to first design a rigid body model mechanism using rotary joints. The rigid body mechanism should be able to complete the designed motion. Then according to the rigid body mechanism, the equivalent compliant mechanisms are designed. The major replace method include using compliant revolute joints and setting up the pseudo-rigid-body model. After selecting a replacement method, complete the rest of the design based on the corresponding design ideas. The revolute joints consist of a single part whose ends can be fused with beams with higher stiffness. They act as the joints connecting multiple rigid bodies and are able to be bent at large angles. They come in a variety of designs. Davood et al. [35] summarize the design of some common compliant revolute joints as shown in Figure 7.

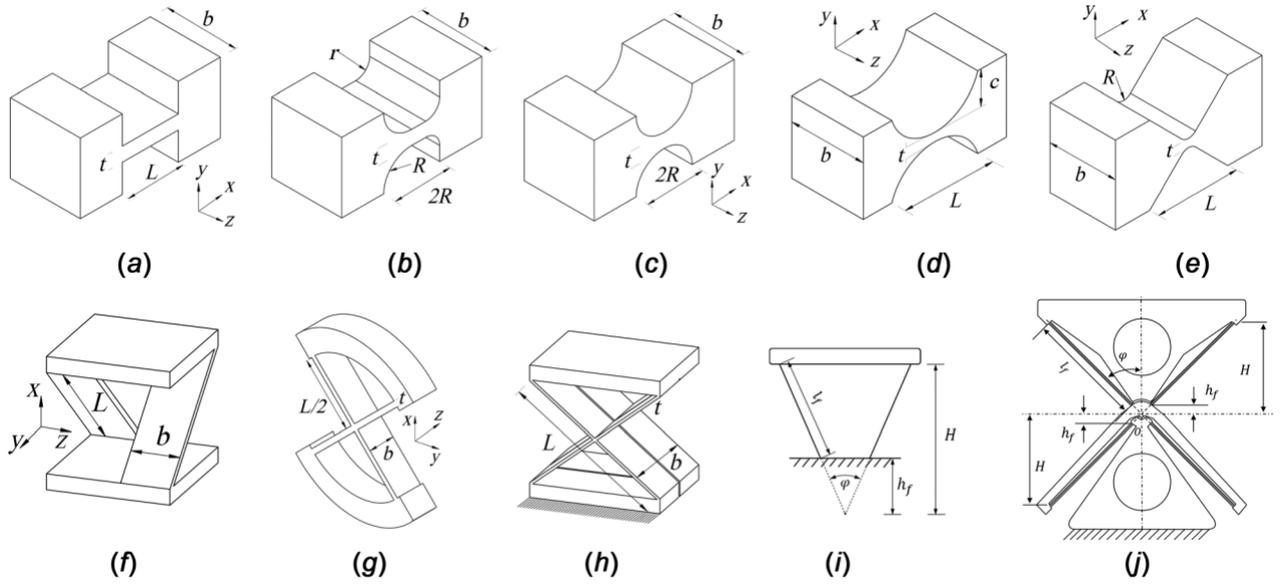


Figure 7: Compliant revolute joints [35]

### 2.3.2 Pseudo-Rigid-Body (PRB) model

PRB model is a method to analyze the macro scale deformation, force, torque and energy of a compliant component when it is subjected to a large deflection. This method can be used to convert a revolute joint into an equivalent compliant mechanism and analyze possible mechanical changes, thus designing suitable compliant joint. In this method, the small joint beam is equivalent to two rigid beams connected by ordinary revolute joints. A torsion spring is installed on the rotary joint. The models established by PRBM are also different depending on whether the deformation is concentrated in a certain part of the beam or widely distributed on the entire beam. Figure 8 illustrated the PRBM of a fixed-free beam with widely distributed deformation. The stiffness of the torsional spring satisfies  $K = 1.5164EI/l$ . Where  $l$  is the length of the small-length flexure,  $E$  is the Young's modulus of it,  $I$  is its and cross-section second moment of inertia. Figure 9 illustrated the PRBM of a fixed-free beam with deformation concentrated on the joint. In this model, the stiffness at the joint is much lower than the beam next to it. The stiffness of the torsional spring satisfies  $K = EI/l$ .

### 2.3.3 Structural optimization

Structural optimization approach, as a main method of compliant mechanism design, is used to find the optimal compliant mechanism design that satisfies the given constraints [34]. The fitness of compliant mechanism design is defined by objective function. An optimization prob-

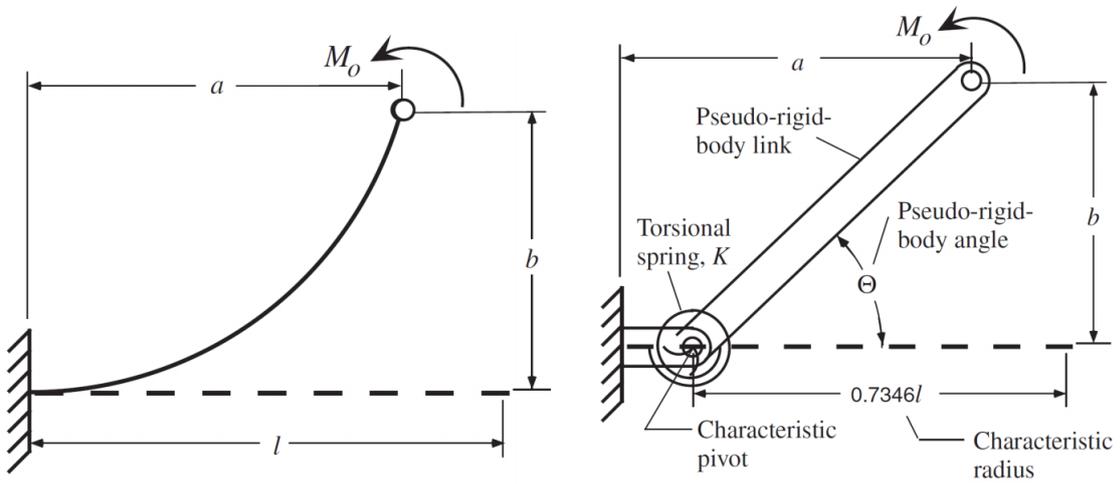


Figure 8: PRBM of a fixed-free beam with widely distributed deformation [12]

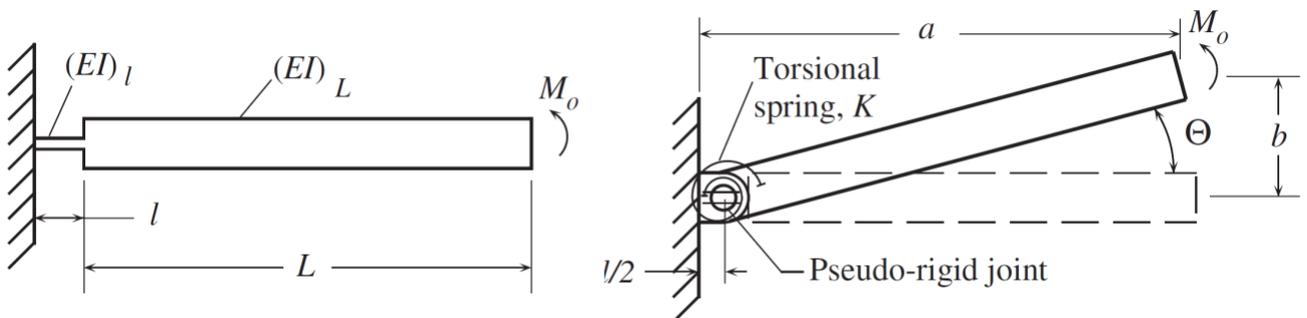


Figure 9: PRBM of a fixed-free beam with deformation concentrated on the joint [12]

lem can be expressed as Equation 6, where  $f(\mathbf{x})$  is the objective function.  $x$  is the vector of design variables.  $h_i$  and  $g_j$  are the constraining equations and inequalities.

$$\begin{aligned}
& \underset{\mathbf{x} \in \Omega}{\text{minimize}} f(\mathbf{x}) \\
& \text{subject to} \\
& h_i(\mathbf{x}) = 0 \quad i = 1, 2, \dots, p \\
& g_j(\mathbf{x}) \leq 0 \quad j = 1, 2, \dots, m \\
& \text{where} \\
& \mathbf{x} = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}
\end{aligned} \tag{6}$$

Structural optimization can be divided into topology optimization, shape optimization and size optimization [34]. For a compliant mechanism, its topology structure includes input, output, ground ports, and the connection (on or off) between different main components of the structure. Minor changes in shape and size will not affect its topology. The shape of the compliance mechanism includes the projected shapes of its beams in different directions, the cross-sectional shape, etc. The dimensional information of the compliance mechanism includes the specific lengths of different components of the model, cross-sectional geometric parameters, etc. All the topology, shape and size parameters are the design variables of the compliant mechanism. The object function can be defined by five dominant characteristics [36]: mutual strain energy and strain energy, mechanical and geometric advantage, energy efficiency, characteristic stiffness, and artificial I/O spring formulation. These characteristics can be defined with some certain functions and weighted according to the needs of the task.

There are several ways to define the topology, shape, and size of the compliance mechanism when performing structural optimization. The definition can be parameterized. For example, when designing a beam generated by a sweep, the coordinates of each point of the sweep reference polyline can be given, and then the cross-sectional shape and coordinates of the beam should also be given. In addition to polyline coordinates, spline curve control point coordinates, arc center coordinate and radius, etc. can also be used as structural parameters. The principle is similar to computer-aided drafting. Different kinds of parameters can be selected according to the task needs.

### 2.3.4 An example of structural optimization

Structural optimization methods can be used to design compliant mechanisms with prescribed input-output relationships. These compliant mechanisms can serve as the equivalent of con-

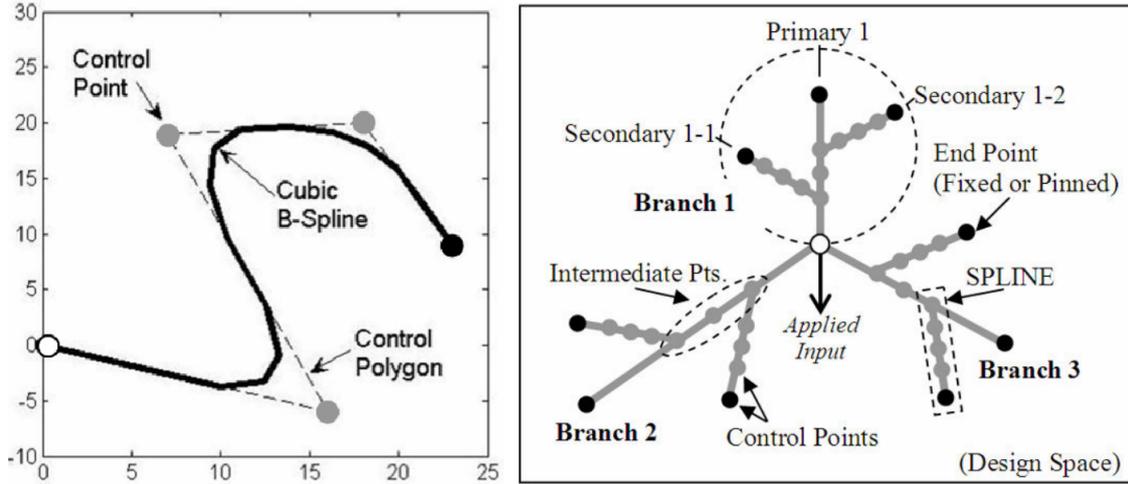


Figure 10: The splines and topology arrangement of the mechanism [37]

ventional mechanical components such as gears, linkages, improving the performance of various devices. Christine Jutte and Sridhar Kota [37] proposed a general approach to optimize the topology, shape, and size of compliant mechanisms. This method can generate nonlinear spring compliant mechanisms with a specific load-displacement function relationship. The nonlinear spring consists of a “fractal”-like network of splines in a single plane. Since the entire compliance mechanism is at the same plane, there are various ways to fabricate, such as stamping, forming, stripe bending, laser cutting and die casting.

The splines are all cubic B-curves. Every curve is generated by 5 control points. The whole mechanism consists of several splines, each spline is connected to a control point of another spline or an external port of the system. External ports can be input ports, output ports, or ground ports. Using sweeps to generate corresponding models along these splines. Figure 10 shows the topology of the compliant mechanism. The boundary conditions, existence, and control points of all splines build up the design variables of the entire compliant mechanism.

To realize the optimization, firstly, the load-displacement curve should be defined. The curve can be a polyline consisting of several key points. The highly customized shape of the curve can be J-curve, S-curve, and a constant force curve (flat-curve). Also, define some geometric and physical property parameters in advance, such as the cross-section thickness and the Young’s modulus of the material.

Then, MATLAB generate serval sets of design variables, input these spline data to python. Python generates beam elements B21H, divides mesh, creates analysis steps and jobs, and calls Abaqus for finite element calculation. Abaqus outputs the load-displacement curve data and von Mises stress.

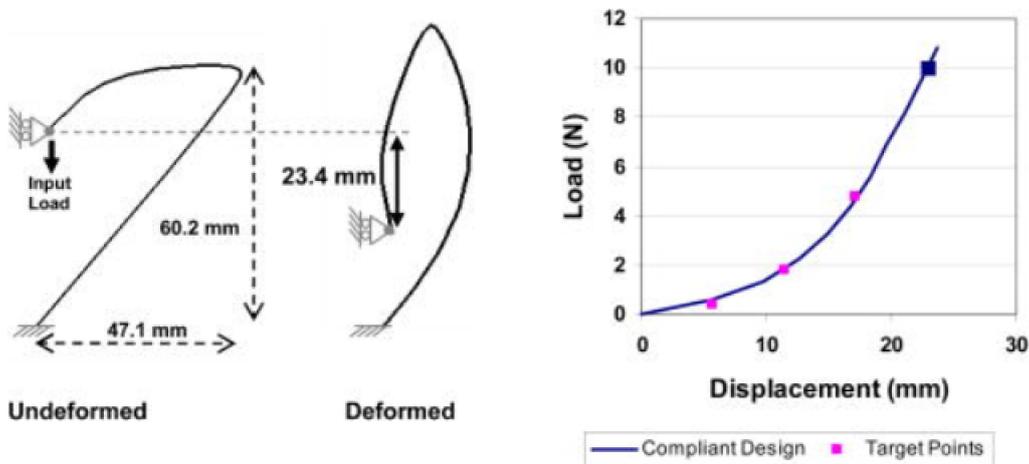


Figure 11: One optimized design and its "J"-shaped load-displacement curve [37]

Thirdly, MATLAB reads the displacement-load data and compares it with the originally prescribed displacement-load function, check if the design will break according to the von Mises stress. Using a genetic algorithm (GA), generate several new sets of variables with better fitness. Loop until the condition to terminate the GA looping is met.

Finally, a nonlinear spring design with ideal load-displacement relationship will be generated. Figure 13 shows one of the designs generated for the J-curve. It can be seen that the load-strain relationship is in good agreement with the design value.

### 2.3.5 Flexible translational joints

Flexible translational joints is a kind of compliant mechanism device that restrict the movement of system component to one direction [38]. The device minimizes the axial drift while achieving a large range of motion. Therefore, the axial stiffness of such devices will typically be much less than the off-axis stiffness. The most important performance criteria of such translational joints include: axial stroke, ratio of off-axis to axial stiffness, and size. A study led by Davood Farhadi et al. [35] investigated the performance of various kinds of compliant translational joints, as shown in Figure 12. Only (c) (f) (g) (h) satisfy the condition of no axis drifts. High stroke can be achieved from (a) to (h). For (a) (f) (h), displacement can be input from outside of the outline. (a) (b) (c) has relatively small areas compared with their stroke.

## 2.4 Discussion

According to the state of art, the design for mode coupling in a 2-degree-of-freedom system that is based on compliant mechanisms has not received wide research attention. Normally, to

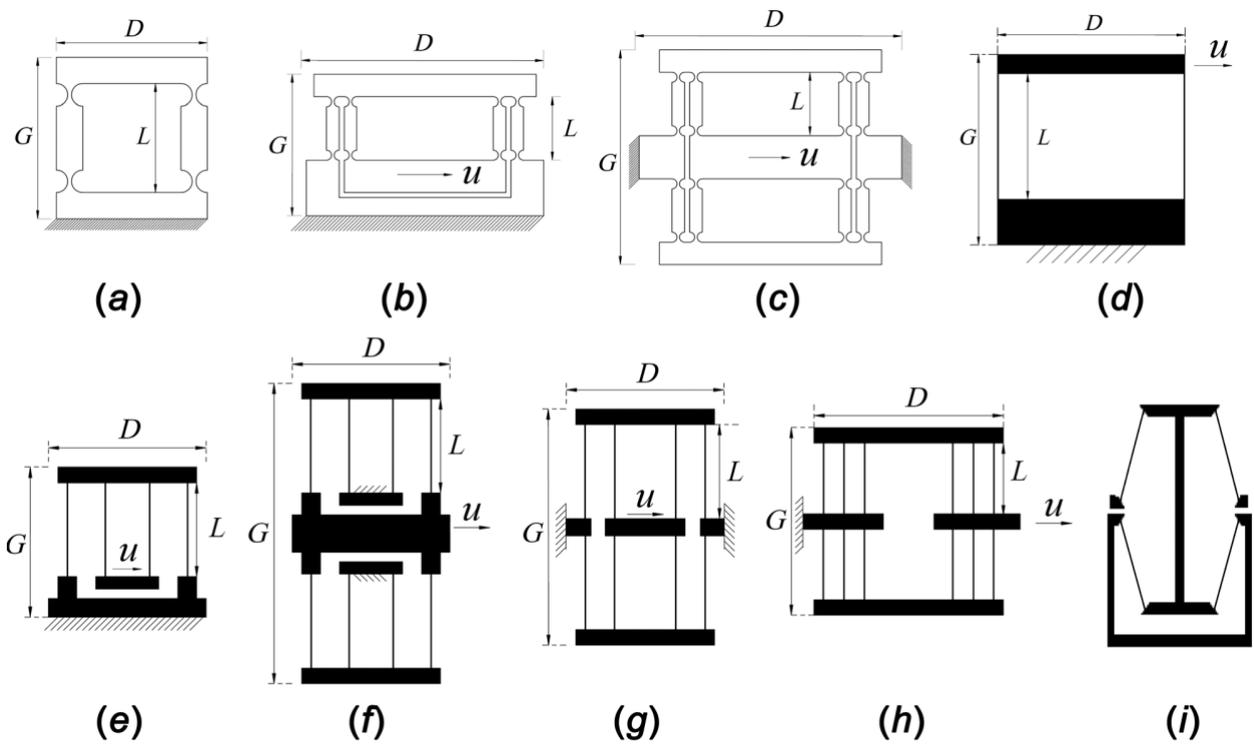


Figure 12: Compliant translational joints: (a) four-bar-notch block and (b) double notch block, (c) symmetrical double notch block, (d) four-bar block, (e) double block, (f) symmetrical double block, (g) folded beam, (h) planar CT joint, and (i) double XBob [35]

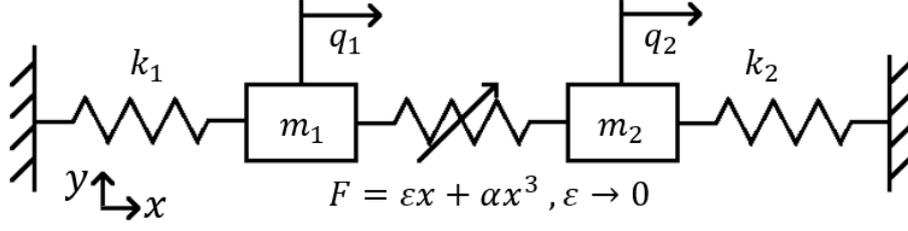


Figure 13: Schematic diagram of the 2-DoF system

couple one system to another, researchers use electrostatic actuation method [13, 39] or optical methods [4]. In this project, a compliant mechanism that use a nonlinear spring to achieve mode coupling is proposed, as shown in Figure 13.

In this 2-DoF system, both masses ( $m_1$  and  $m_2$ ) move only in  $x$ -direction. They are connected to ground with two linear spring  $k_1$  and  $k_2$ . The displacement of  $m_1$  and  $m_2$  in  $x$  direction are represented by  $q_1$  and  $q_2$ , respectively. A cubic linear spring with a very small linear term is used to couple the two masses. Equation 23 is the equation of motion of this system.

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} k_1 + \varepsilon & -\varepsilon \\ -\varepsilon & \varepsilon + k_2 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} + \begin{bmatrix} \alpha (q_1 - q_2)^3 \\ -\alpha (q_1 - q_2)^3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (7)$$

Use the eigenmode term  $[X]$  (Equation 24) of the equation of motion to transformation the system to modal space, as Equation 25 shows.  $f_{nl}$  is the nonlinear term.

$$\begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = [X] \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} \quad (8)$$

$$\begin{bmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{bmatrix} \begin{bmatrix} \ddot{\eta}_1 \\ \ddot{\eta}_2 \end{bmatrix} + \begin{bmatrix} \gamma_1 & 0 \\ 0 & \gamma_2 \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} + \begin{bmatrix} x_{(1)}^T & f_{nl}/\mu_1 \\ x_{(2)}^T & f_{nl}/\mu_2 \end{bmatrix} = 0 \quad (9)$$

The ratio of resonance frequencies satisfies Equation 26. It is a very complex formula that involves many variables and parameters. However, when the linear term of the cubic spring becomes very small. The frequencies will satisfy Equation 27, which is very simple and elegant.

$$\frac{\omega_2}{\omega_1} = \frac{\sqrt{\gamma_2/\mu_2}}{\sqrt{\gamma_1/\mu_1}} \quad (10)$$

$$\varepsilon \rightarrow 0, \omega_1 \rightarrow \sqrt{\frac{k_1}{m_1}}, \omega_2 \rightarrow \sqrt{\frac{k_2}{m_2}} \quad (11)$$

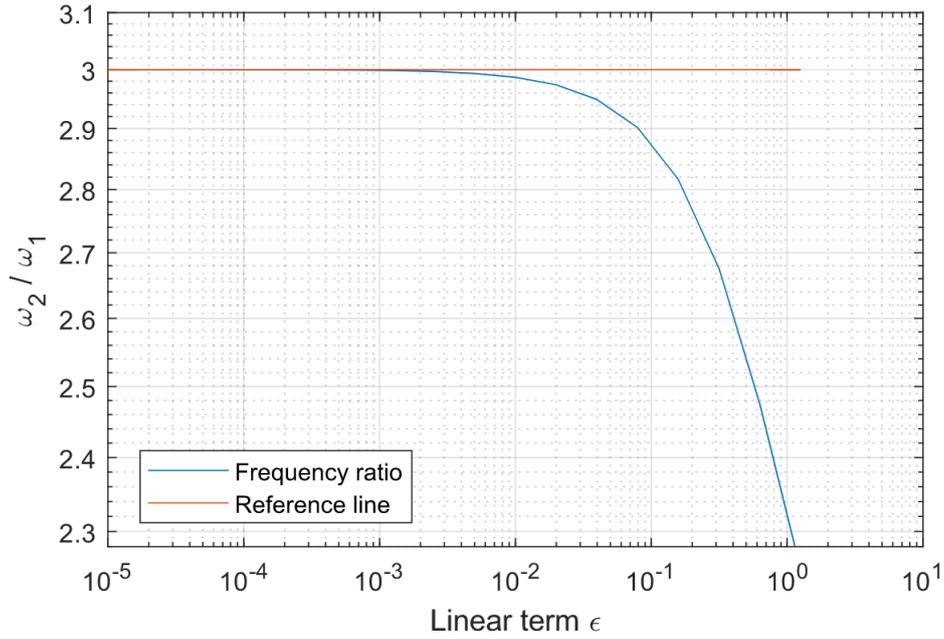


Figure 14: Relationship between frequency ratio and linear term

For example, if  $k_1 = 9k_2$ ,  $m_1 = m_2$ , the ratio of resonance frequencies will be like Figure 14, when linear term  $\epsilon$  approaches zero, the resonance frequencies ratio will be equal to 3, which lead to a 1:3 internal resonance for the system. Therefore, by designing or tuning the  $k_1$  and  $k_2$ , an internal resonance with prescribed ratio can be achieved.

To conclude, according to the mathematical analysis above, the design of the spring with extremely high nonlinearity is very important. Its design can be done according to the knowledge mentioned in the previous compliance mechanism design section and structural optimization section. Despite the numerous advantages of using nonlinear springs in MEMS, there are several challenges that need to be addressed in order to optimize their design and performance. The fabrication of nonlinear springs in MEMS can be challenging due to the small size of the device and the complexity of the design. Besides, the characterization and modeling of the mechanical properties of nonlinear springs can be difficult due to their nonlinear behavior and hysteresis. Also, the integration of nonlinear springs with other MEMS components can be challenging due to the different fabrication processes and materials used. Furthermore, in order to ensure that the two masses move in one direction, simple linear springs on both sides are not sufficient, because they will introduce an axial offset. Therefore, it is necessary to develop a suspension stage. The suspension stage design can be completed according to the knowledge of the previous flexible translational joints section.

### 3 Paper

# COMPLIANT MECHANISM BASED NONLINEAR SPRING DESIGN FOR INDUCING MODE COUPLING

Haolang Ding <sup>1</sup>

*Department of Precision and Microsystems Engineering*

*Delft University of Technology*

---

## Abstract

Mode coupling has extensive use in the MEMS field, including frequency division, vibration direction conversion, and energy transfer. In practice, these applications can be used to improve the performance of various devices such as sensors and energy harvesters. Nonlinear spring also has a wide application in MEMS field. Various nonlinear spring mechanisms, such as fixed-angle bows, bistable rotational mechanisms, H-shaped springs, and topology-optimized planar springs, have been proposed and utilized. Nevertheless, there is a lack of non-electric design elements specifically designed for mode coupling. This thesis proposes a simple system demonstrating the feasibility of using a nonlinear spring to achieve mode coupling. The system incorporates a spring-mass system with two mass blocks, two linear stages, and a unique nonlinear spring compliant mechanism. The integrated components can be fabricated using 3D printing resin or high-precision femtosecond laser-cutting of silicon wafers. Additionally, a new crank-slider structure and a spline-shaped nonlinear spring are developed and studied for this system. Each has advantages and disadvantages, and is suitable for systems of different sizes and materials respectively. Their load-displacement relationship roughly satisfies the cubic relationship, but still has a linear term. The proposed system holds potential for enhancing the performance of sensors, energy harvesters, and other MEMS devices.

*Keywords:* Compliant mechanisms, Nonlinear spring, Mode coupling, Nonlinear dynamics

---

<sup>1</sup>h.ding-2@student.tudelft.nl

### 3.1 Introduction

The principle of mode coupling has been widely used in the MEMS field. Its applications include, but are not limited to, frequency and division[1], changing of vibration direction (e.g. from displacement to torsion) [2, 3], transfer of energy[4]. It can be used to improve the performance of sensors [5] and energy harvesters [6, 7, 8] in their sensitivity and bandwidth by transferring the energy from a specific frequency range to another frequency. Vibration energy in different directions and frequencies can be transferred to another mode by taking advantage of the geometric nonlinearity. For MEMS systems, nonlinear springs also have multiple uses, such as eliminating the pull-in instability of the hybrid actuator[9], and improving the performance of robotic joint[10]. There are various spring-like mechanisms with nonlinear characteristics, such as fixed-angle bows [11], bistable rotational mechanism[12], H-shaped spring [13], splines[14], cam structure[15], and topology-optimized planar springs [16].

Although many nonlinear springs have been proposed and widely applied, and various optical or electrostatic actuation methods have been employed to achieve mode coupling, there is still a lack of corresponding design elements for nonlinear springs that is specifically designed to achieve mode coupling. To realize such application, a simple system is proposed to demonstrate the feasibility of using a nonlinear spring to achieve mode coupling. It is also required to design a nonlinear spring utilizing the concept of compliant mechanism for this system.

In this thesis, a spring-mass system is designed, which includes two mass blocks, two linear stages, and a nonlinear spring. The two blocks are connected together using a nonlinear spring and each block is connected to the ground using a linear spring. The deformation direction of the linear spring and the nonlinear spring are coaxial. The nonlinear spring compliant mechanism has a unique shape that, when a load is applied to its end in a particular direction, there is an approximately cubic relationship between the load and the displacement at that point. The utilization of a cubic relationship for the nonlinearity is attributed to its ease of design and implementation. The linear stages are flexure structure that function as ordinary linear springs and constrains the degrees of freedom of the blocks except axial displacement. All components are integrated to a compliant mechanism in a single plane. The system can be made of 3D printing resin or made by high-precision femtosecond laser-cutting silicon wafers. For the nonlinear spring, this thesis proposes a new nonlinear spring compliant mechanism that is similar to a crank-slider structure. For the problems described in this thesis, an alternative spline-shaped nonlinear spring is also developed and studied.

## 3.2 Nonlinear spring design methods

This section describes the design techniques for the nonlinear spring with an approximately cubic relationship between the load and the displacement. The techniques includes FEM simulation and genetic algorithm optimization. In fact, nonlinear springs with higher order terms (such as 5th order, 7th order) can theoretically also be used in this system. However, as the displacement increases, the required load and the stress it brings will also increase rapidly, which will bring great challenges to the design. Therefore, the cubic behavior is finally selected as the nonlinear spring.

Two nonlinear spring design strategies are examined in this thesis. One is an spline-shaped mechanism inspired from literature [14]. The other is a newly developed method in this thesis, which is named crank-slider mechanism according to its shape. The two designs are suitable for different situations (available sizes and materials). Besides, although the most critical design element in the system is the nonlinear spring. Blocks connected to non-linear springs also need to be connected to the ground using a linear spring stage. Therefore, the design method of the linear spring stage is also introduced. It uses the concept of simple flexible translational joints. In this section, lumped model verification is also performed on FEM software COMSOL to ensure that the result of finite element model copes with the mathematical models.

### 3.2.1 Linear stage design

Linear stage is a compliant mechanism device that connects ground and the mass block together. When an object moves in a specific direction, this device exerts an opposing force proportional to block displacement, just like a spring. The device only allows the movement of the object in one direction only, and that direction is the same as the axis of the spring. In other words, this mass has only one degree of freedom. The full system in this thesis has two linear stages as the suspensions for two masses. The motion directions of the two linear stages are coaxial. However, the two linear stages are not necessarily the same. Since their equivalent linear springs may have different stiffness depending on the system design. Therefore, the geometry of these linear stages could be potentially designed differently. — Flexible translational joints [17] is a kind of compliant mechanism device that perfectly suits the linear stage requirements. The basic design of flexible translational joints is shown in Figure 15. It utilize the bending of blade flexures to provide stiffness. The most important parameters of the flexures are length  $L$ , width  $w$ , and thickness  $t$ . When the mass block is moving in  $x$  direction with a displacement of  $\Delta x$ , it will receive a counter force Equation 12, where  $E$  is the Young's modulus of the material. Therefore, the effective stiffness of the linear stage becomes Equation 13. However, the bending

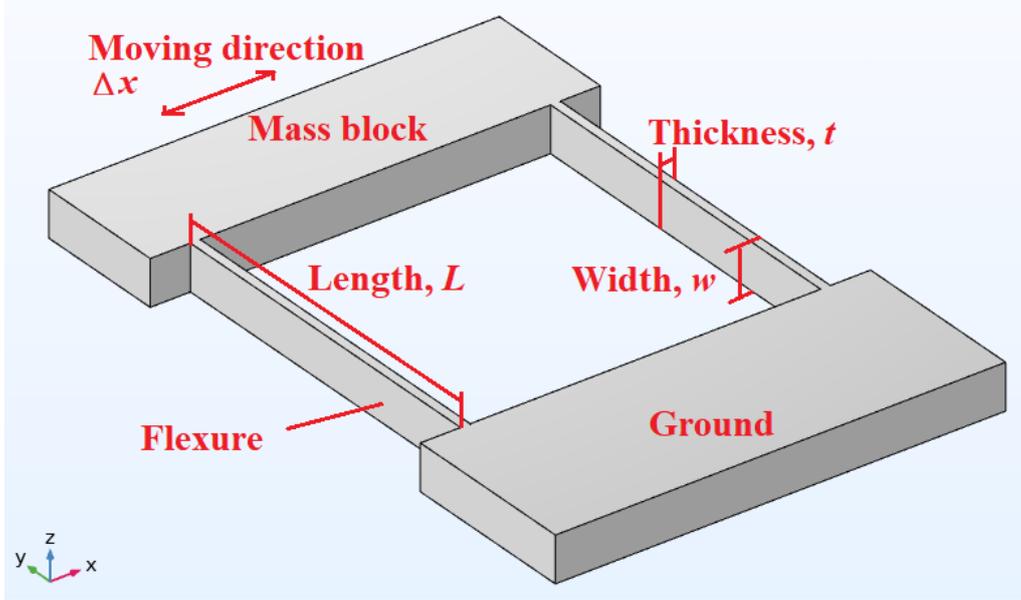


Figure 15: A basic design of flexible translational joints

of flexures will also introduce axis drift  $\Delta y$  in the  $y$  direction, which roughly satisfies Equation 14 [18]. When the  $x$  displacement stroke is about 20% of the length of flexures, axis drift  $\Delta y$  will be only 2.4 % of flexure length  $L$ . For the linear stage in this thesis, out-of-plane deflection is not under consideration because of the limited amplitude of the input excitation and strength of the material. Since the side surface of the mass block are exposed and free from interference from other flexures, external loads can be applied to the mass block directly. — Flexible translational joints [17] are compliant mechanism devices that are well-suited for linear stage applications. The basic design of these joints, as depicted in Figure 15, relies on the bending of blade flexures to provide the necessary stiffness. The key parameters governing the behavior of the flexures are their length ( $L$ ), width ( $w$ ), and thickness ( $t$ ).

When a mass block undergoes a displacement  $\Delta x$  along the  $x$  direction, it experiences a counter force:

$$F = \frac{2Ewt^3}{L^3} \Delta x \quad (12)$$

$E$  represents the Young's modulus of the material. Consequently, the effective stiffness of the linear stage can be expressed as shown below:

$$K_{\text{eff}} = \frac{F}{\Delta x} = 2 \times \frac{12EI}{L^3} = \frac{2Ewt^3}{L^3} \quad (13)$$

However, it is important to note that the bending of the flexures also introduces a certain

amount of axis drift  $\Delta y$  in the  $y$  direction [18] , which can be approximately described as:

$$\Delta y \approx 0.6 \frac{\Delta x^2}{l} \quad (14)$$

It should be emphasized that the axis drift is generally modest, reaching only around 2.4% of the flexure length  $L$  when the  $x$  displacement stroke is approximately 20% of the flexure length.

In the context of this thesis, the linear stage design primarily focuses on in-plane deflection, as the amplitude of the input excitation is limited and the material's strength is taken into consideration. Additionally, since the side surfaces of the mass block remain exposed and free from interference by other flexures, external loads can be directly applied to the mass block.

### 3.2.2 Lumped model verification

In this thesis, the most important design parameters of the whole system are spring stiffness, mass, and nonlinear spring performance. These system design parameters can be determined according to numerical simulation in Simulink or MATLAB. It is necessary to establish a lumped model for the system to verify that the response of the numerical model is consistent with that of the finite element model. A lumped model is often used in engineering and physics to simplify complex systems and analyze their behavior. It allows us to represent the system using a reduced number of elements, making it easier to understand and analyze the overall system response. It has ideal parameters and a simplified structure.

Since system design parameters (stiffness, mass, etc.) often change according to geometric size parameters (such as width, thickness and length of the flexures), and the topological relationships remain constant, it is necessary to develop a more convenient modeling method, namely parametric and feature-based modeling. The software COMSOL is employed for both CAD design and FEM simulations of the system and its constituent components. By utilizing parameter-defined geometric dimensions rather than fixed values, this modeling methodology facilitates efficient editing and modification through non-GUI code methods.

To assess the fidelity of COMSOL's modeling and simulation capabilities, a simplified lumped model system was devised, as shown in Figure 16. This reduced model serves to streamline the analysis. It comprises two rigid mass blocks and three slender elastic bars with linear behavior. These bars, constructed from distinct materials, possess specific Young's modulus values, thereby imparting designated stiffness to the elongation and compression of these slender rods, analogous to springs. Furthermore, the mass blocks exhibit varying densities, enabling

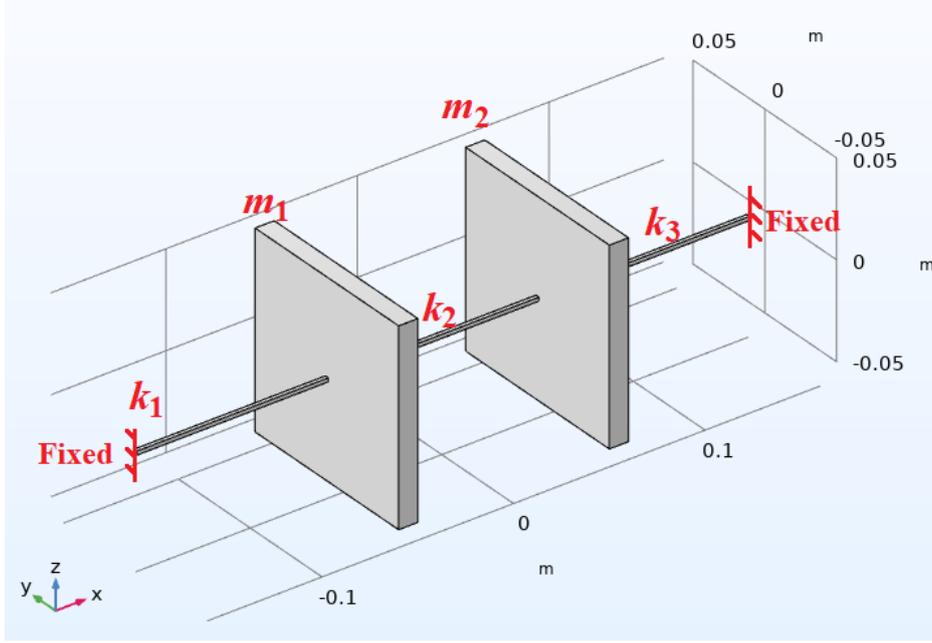


Figure 16: The lumped model for the COMSOL

customization of their respective masses. Notably, unlike the comprehensive systems explored in this thesis, the central spring within this lumped model assumes a conventional linear behavior to facilitate the modeling. The equation of motion of this system satisfies:

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix} + \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & +k_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} F_1(t) \\ F_2(t) \end{bmatrix} \quad (15)$$

An input step load of 5N for a duration of 0.1s was applied to mass  $m_1$  in the lumped model, and transient analysis was conducted using COMSOL. Additionally, a set of motion equations was established in MATLAB, incorporating the corresponding initial conditions (step load), to calculate the transient response of the two masses. The results, as shown in Figure 17, exhibit a high degree of agreement between the two software platforms, affirming the suitability of COMSOL for CAD modeling and FEM simulation based on mathematical models. This approach entails determining the system's crucial parameters (e.g., stiffness, mass) and employing mathematical tools such as MATLAB and Simulink to compute the theoretical system response. These parameters are subsequently utilized for modeling and finite element analysis in COMSOL.

The consistency observed between the results obtained from the mathematical tools and the finite element software in this lumped model experiment ensures the reliability of the overall process. Utilizing the efficiency of mathematical tools compared to finite element calculations,

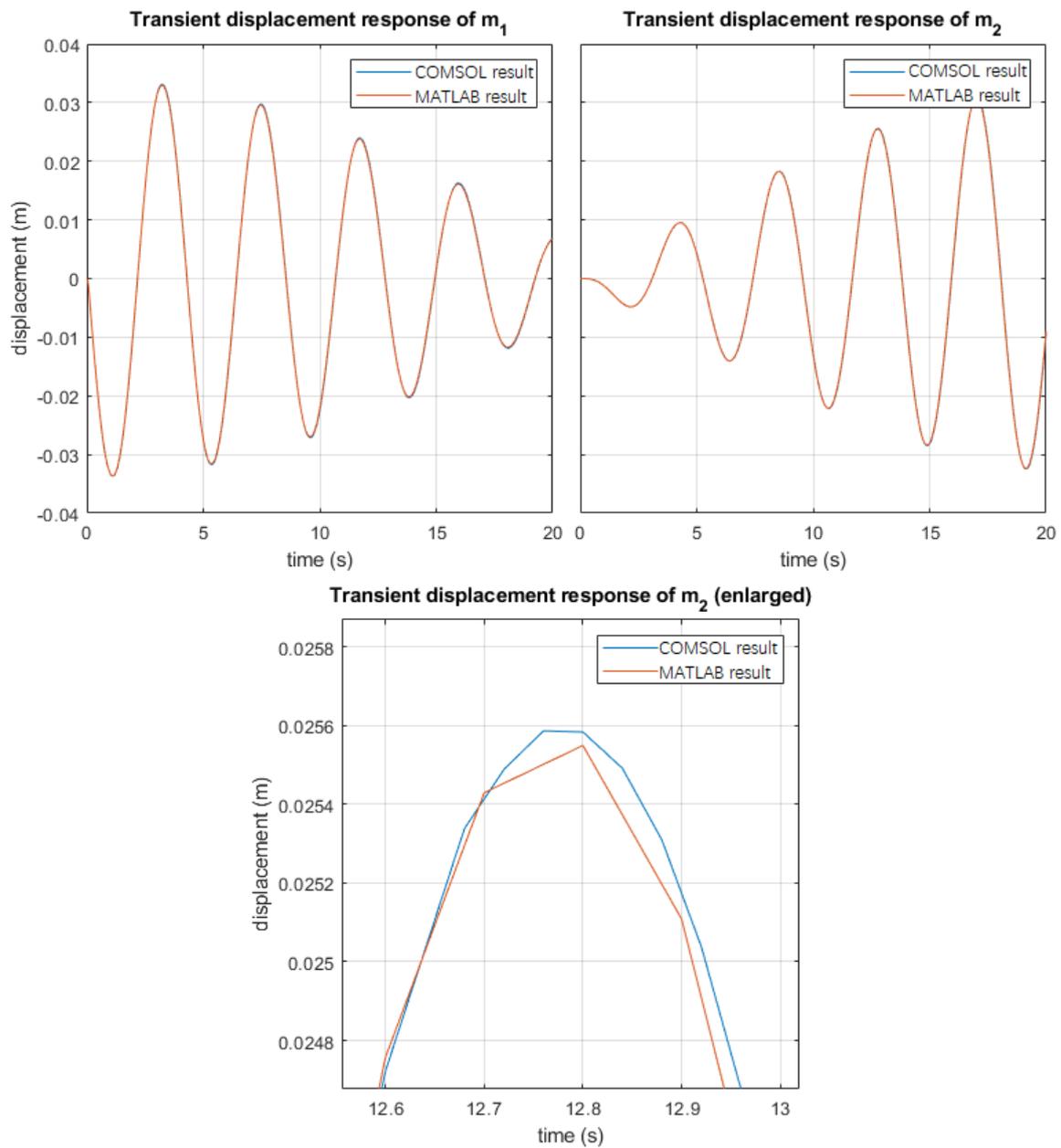


Figure 17: The lumped model response of different software

this method saves significant time during the debugging phase. However, it should be noted that the lumped model represents an idealized approximation, and subsequent iterations are indispensable in subsequent designs to mitigate errors from factors such as spring mass and geometric approximations.

### 3.2.3 Design of spline-shaped mechanism nonlinear spring

**Design principles** In this section, a design method based on spline-shape to achieve spring with a cubic load-displacement characteristics is developed. This method enables the generation of nonlinear springs with a non-linear relationship between load and displacement. The nonlinear spring is constructed using a fractal tree-like branching network of splines within a single plane, with at least one spline included, thus referred to as a spline-shaped mechanism. This approach is inspired from a previous generalized method [14] that optimizes the topology, shape, and size of nonlinear springs to achieve desired load-displacement curves. The compliant mechanism, being entirely planar, allows fabrication through techniques such as 3D printing and laser cutting.

The spline curves are defined by control points and can be generated using interpolation curves or cubic Bézier curves. The mechanism comprises one or more splines, with the primary nonlinear spring integrated with the mass blocks to fulfill the requirements of the compliant mechanism. The remaining splines possess three potential terminations: fixed to a control point of another spline, fixed to the ground, or unsupported (similar to a cantilever). Sweeps are employed to generate corresponding models along these splines.

In this thesis, MATLAB is utilized to generate multiple sets of design variables, including control point coordinates and cross-section thickness. COMSOL is then employed to generate a 3D solid model based on the MATLAB design variables. The process involves mesh division, establishment of boundary conditions, and subsequent finite element calculations based on the spline data. The resulting data includes the load-displacement curve and von Mises stress. MATLAB reads the displacement-load data, assesses its nonlinearity, and checks for potential design failure using the von Mises yield criterion. By employing a genetic algorithm (GA), several new sets of variables with improved fitness are generated. The GA chromosome typically contains around 20 design variables. The GA iteration continues until a specified termination condition is met. Eventually, a design for a nonlinear spring with a cubic load-displacement relationship is obtained. The workflow of the project is illustrated in Figure 18.

In certain cases, automatically generated splines may intersect or cross over themselves. An example of self crossing over is shown in Figure 19. However, for the purposes of this thesis, where fabrication methods are limited to laser cutting and 3D printing, there is no need to address this issue by reordering control points. However, if stripe bending fabrication method were to be introduced, it would be necessary to examine and account for potential splines crossover.

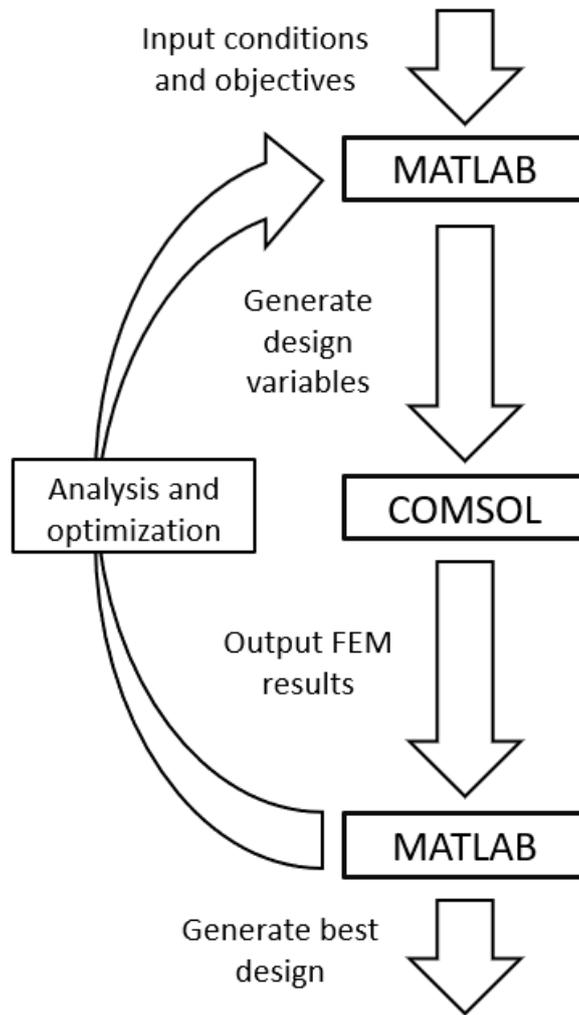


Figure 18: The optimization logic schematic diagram

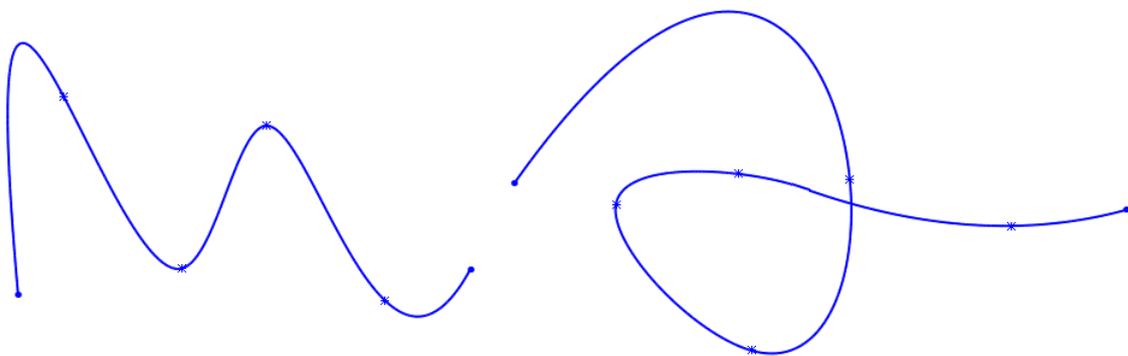


Figure 19: Spline with (left) and without (right) self-crossover

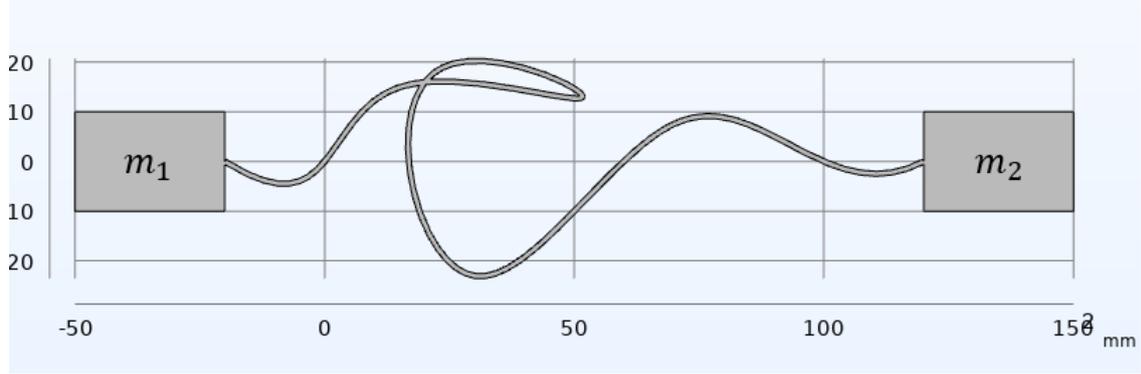


Figure 20: A basic 3D model of spline mechanism nonlinear spring with one primary spline

It is pertinent to highlight that during testing, employing two-dimensional beam elements for model construction yielded significant computational time savings. However, in the context of this thesis, the presence of potential splines crossover necessitates the use of three-dimensional solid models. To investigate this, both a beam element model and a three-dimensional solid model were developed for the same design, employing fine mesh densities and subject to identical boundary conditions. As a result, the load-deformation response curves exhibited notable disparities between the two models. This discrepancy may arise from the fact that the boundary conditions at the junction of the beam elements and the solid elements (mass block) differ considerably from those encountered in the more complex model and real-world scenarios. Consequently, it is imperative to conduct analyses based on three-dimensional solid models to accurately capture the system's behavior.

**Optimization implementation** To begin, a basic 3D model is constructed using COMSOL, as depicted in Figure 20. This model includes a nonlinear spring spline and two cuboids attached to its ends. The material is designated as acrylic, with properties sourced from the COMSOL material library. The two cuboids serve as platforms for applying supports and loads. The spring's main body is created by thickening the spline curve, which is generated based on the coordinates of predefined control points. The left cuboid is fixed, and a maximum load  $F$  is applied to the right surface of the other cuboid.

Using COMSOL's built-in tools, a mesh is generated and a stationary study incorporating geometric nonlinearity is performed, the parametric sweep function is employed to assess how the spring's deformation varies with load  $F$ , enabling the alteration of parameter values within a specified range and step increment. For instance, the load can be set to various values from  $-F$  to  $F$  in increments of  $0.05F$ .

Calculations are then conducted to determine the corresponding deformation and stress of the

object under each load. A probe records the right cuboid's displacement along the  $x$ -axis and the maximum von Mises stress throughout the model. This process generates a load-displacement data table for the spring. The project file (.mph) is then saved.

MATLAB can read this .mph project file and directly instruct COMSOL to execute simulations without a GUI. Moreover, nearly every aspect of the project file can be altered by modifying its MATLAB code. If needed, additional control points can be introduced for each spline, although this may increase the calculation time.

The optimization process for spring design is conducted using MATLAB and employs a genetic algorithm. For designs incorporating a single spline, the design variables in the optimization are the thickness of the spline body and the coordinates of the spline curve control points. In this model, the coordinates of the control points at both ends are fixed parameters; that is, the endpoints of the spline curve are predetermined. However, the remaining five control points in the middle are variables, constrained to a rectangular area between the two cuboids.

On the other hand, the geometric parameters of the rectangular force platform, the out-of-plane thickness, maximum load, grid density, load amplitude, and output data format are all fixed parameters. While these parameters can be readily modified in MATLAB, they are not the variables being optimized.

If the design calls for multiple splines, the variables will additionally include the coordinates of the new spline control points and the end type of the new spline, which could be fixed to the ground, attached to another spline, or unsupported. The necessity for multiple splines can be manually determined in advance, or can be automatically established by the system within the optimization program. In addition, the number of control points of the additional spline, and the range where these control points are located can also be changed.

The genetic algorithm utilized in this thesis does not employ the built-in toolbox provided with MATLAB, allowing for superior customization. The process begins by importing the .mph project file and setting the genetic algorithm parameters, including population size, number of generations, crossover rate, and mutation rate.

Next, if necessary, modifications are made to the fixed parameters of the model. Following this, the design variables for the initial generation are randomly generated within the designated range, MATLAB then automatically adjusts the variables in the project file accordingly and prompts COMSOL to perform simulations for each individual in the initial generation, The load-displacement probe table for each individual is then saved as a local .txt file.

Assuming that the data table of a given design contains a total of  $N$  pairs of load-displacement elements (where the load of the  $i$ -th element is  $F_i$  and the corresponding displacement is  $x_i$ ), MATLAB's built-in non-linear least squares fitting tool is utilized to fit the load-displacement data to a function:

$$f(x) = P_1x + P_2x^3 \quad (16)$$

As such, the fitted load for displacement  $x_i$  is represented as  $f_i$ . Here,  $P_1$  is the fitted linear term, and  $P_2$  represents the cubic term.

If the fitting function closely matches the original data and the linear term is small, it can be inferred that the load-displacement curve of this design group is satisfactory.

To ensure minimal deviation between the design results and the standard cubic fitting function, a shape function error (SFE) is introduced. The SFE quantifies the error between the fitting function and the original data, as demonstrated below:

$$\text{SFE} = \frac{\sum_{i=1}^N |F_i - f_i|}{N \cdot \max(|f_i|)} \cdot 100\% \quad (17)$$

The SFE is non-dimensionalized, making it applicable to designs of various sizes. A smaller SFE indicates a more desirable design, and it typically falls within the 10 % range.

As mentioned before, nonlinear spring should have smaller linearity. Therefore, the linearity penalty (LP) is used to measure the linearity of the fitted curve, as shown below:

$$\text{LP} = \frac{P_1 \cdot \max(|x_i|)}{P_1 \cdot \max(|x_i|) + P_2 \cdot \max(|x_i|)^3} \cdot 100\% \quad (18)$$

LP measures the proportion of linear force in total load at largest deformation. LP is between 0 and 100%.

In addition, in order to avoid the model fracture due to stress. Stress penalty (SP) is also introduced, as expressed below:

$$\text{SP} = \begin{cases} 0, & \max(\text{von Mises stress}) < \text{Yield stress} \\ W_P, & \max(\text{von Mises stress}) \geq \text{Yield stress} \end{cases} \quad (19)$$

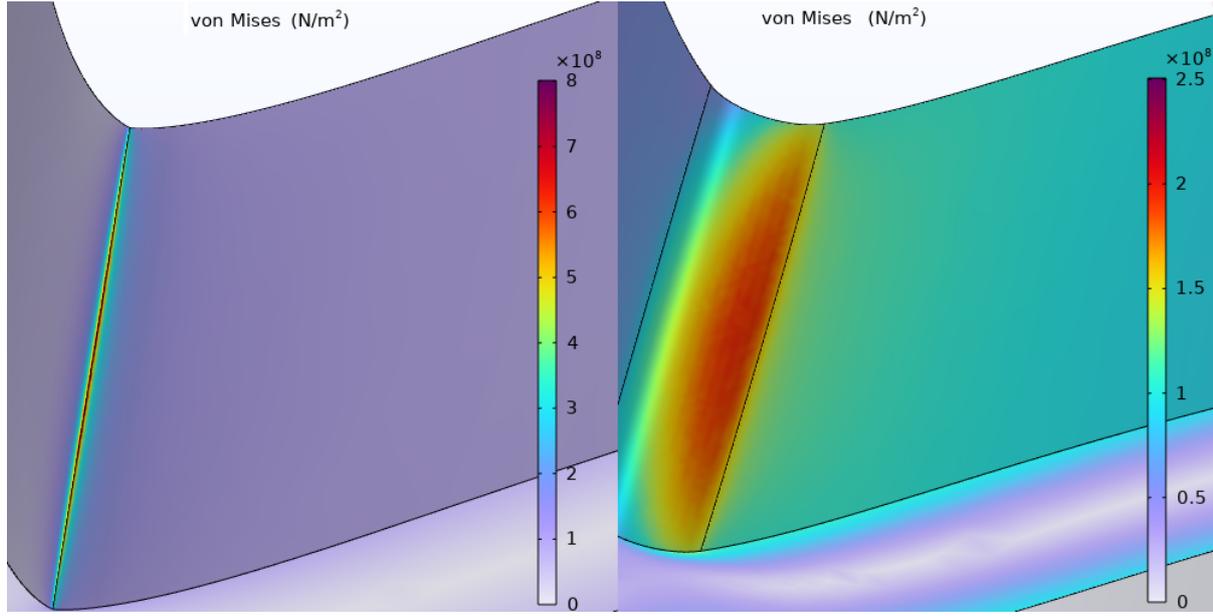


Figure 21: Abnormal surface and mesh. The left figure is the model before manual fine-tuning (maximum von Mises stress is 779.64 MPa), the right figure is after fine-tuning (maximum von Mises stress is 250.63 MPa)

$W_P$  is the weight parameter of penalty. It is set to 100 in this thesis. It is important to highlight a limitation of the 3D sweep function in COMSOL, which can introduce errors on the surface and result in the smooth surface becoming discontinuous at turning points. This irregularity can lead to local mesh distortions, causing extremely high von Mises stress values, as shown in Figure 21. Through multiple test iterations, it has been observed that these stress values can be up to four times larger than those obtained after manual fine-tuning.

Furthermore, not all designs require the maximum load to achieve large stroke capabilities. Consequently, in the stress penalty criterion, the yield stress can be set higher than the actual material parameters. This approach serves to prevent potentially favorable designs from being eliminated due to incorrect simulation results.

Based on the results obtained from multiple trials, SFE typically exhibits smaller values compared to the LP. However, it is important to consider both SFE and LP in the evaluation of design fitness. To account for their relative importance, a weight factor of 5 is applied to SFE. This weighting strategy ensures a balanced consideration of all three factors in the optimization process. As a result, the objective function becomes the minimization of the sum of weighted SFE, LP, and SP, as shown below:

$$\text{Obj function} = \min(5 \cdot \text{SFE} + \text{LP} + \text{SP}) \quad (20)$$

The fitness of each individual in the first generation design is obtained based on the objective function. Sometimes, the designs have very close fitness. In order to create differences, fitness can be squared or cubed, making it easier to screen out excellent individuals. Using roulette wheel method to select the next generation of designs and performing crossover and mutation operations. The loop is terminated when reaching the maximum generation number.

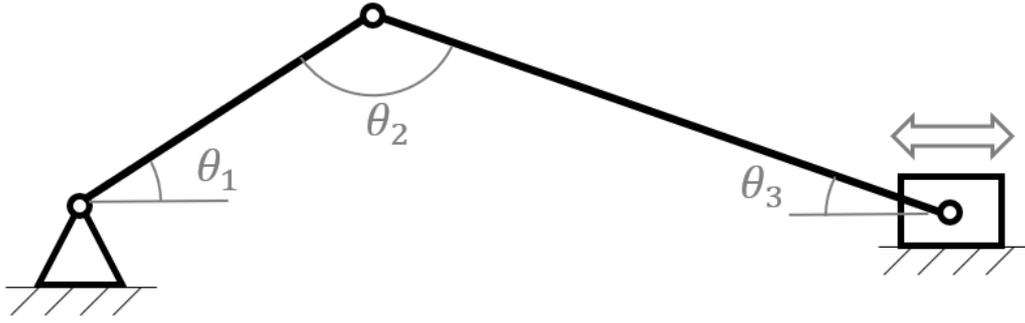


Figure 22: A crank slider linkage

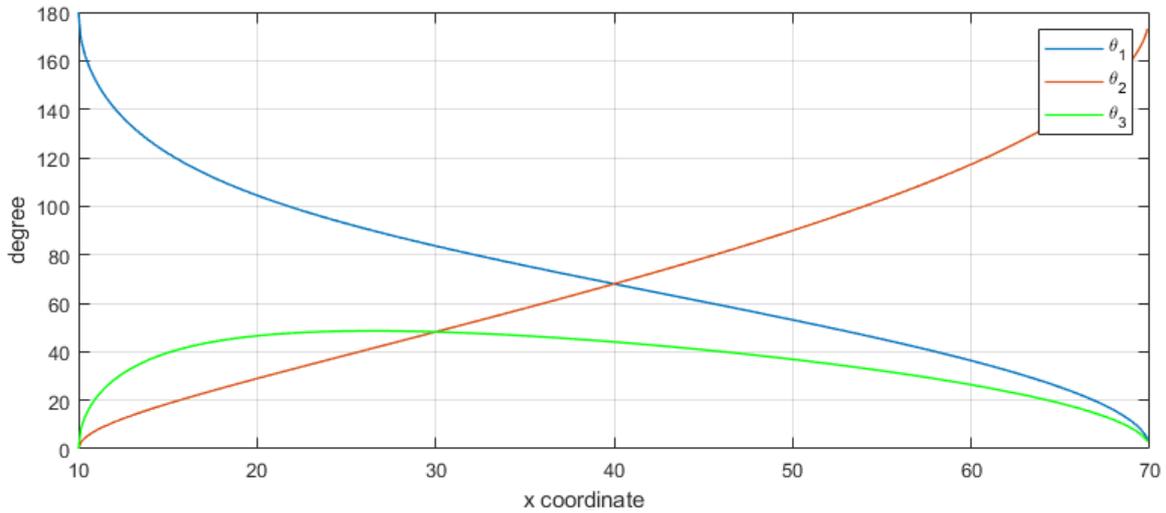


Figure 23: Deflection angle of flexures

### 3.2.4 Nonlinear spring based on crack-slider mechanism arrangement

**Design principles** In this section, a novel design method to achieve spring with a cubic load-displacement characteristics is developed. It is named crank slider mechanism. This design concept draws inspiration from the traditional crank slider linkage, wherein the rods are interconnected through revolute joints, as shown in the Figure 22.

In the crank slider mechanism, as the slider undergoes translation, the angles between the different rods vary. To illustrate this, an example is introduced, where the length of the crank is 30 mm and the length of the other rod is 40 mm. As the displacement of the slider changes, the angles between the rods will also change accordingly, as depicted in Figure 23. These angle variations contribute to the unique load-displacement characteristics of the crank slider mechanism.

In this thesis, compliant hinges are employed to replace conventional rotation joints in the crank slider mechanism, as depicted in Figure 24. The compliant hinges used in this design are blade

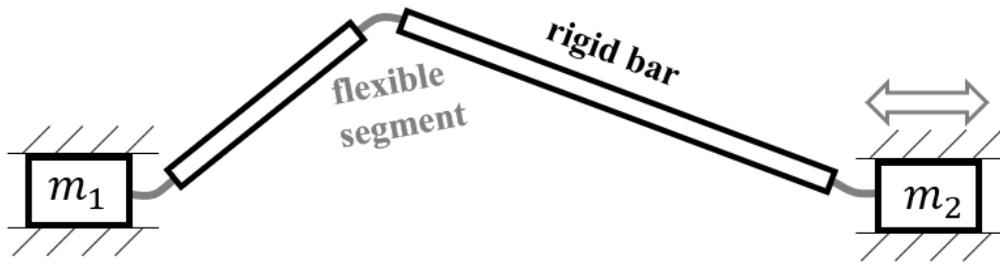


Figure 24: A crank slider compliant mechanism

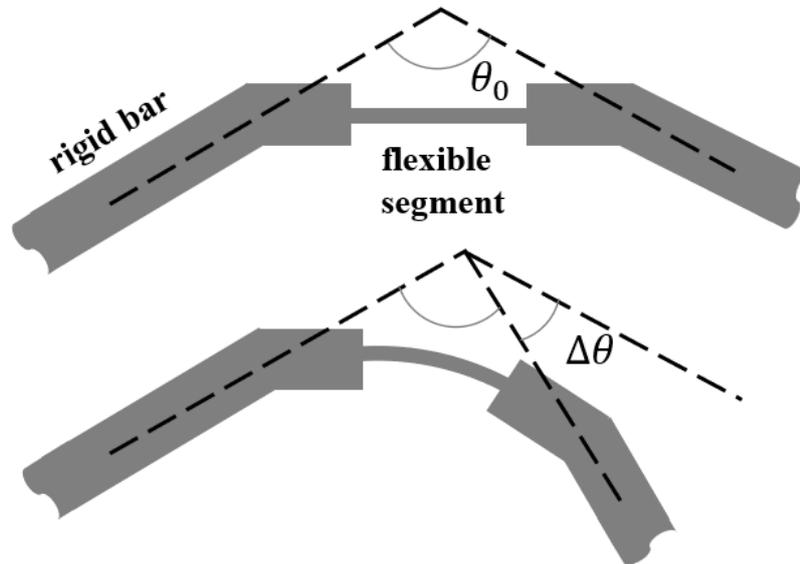


Figure 25: Rigid bars with bending ends

flexures, which serve as connectors between rigid bars with rectangular cross-sections. The in-plane thickness of the bars is significantly greater than that of the flexures. To ensure that each flexure remains unbent when the entire system is in its natural position, and to maintain a specific angle between each rod at the same time, a bend can be added to the end of the rigid bar, as shown in Figure 25 so that even if there is an angle  $\theta_0$  between the two rigid bars, the flexure in between will remain unbent.

During the translation of the slider in the crank slider mechanism, the degree of bending in the blade flexures varies, leading to different levels of torque. The displacement of the slider is directly related to the force it experiences, which is influenced by the geometric and material properties of each component, including the rigid bars and flexures. This particular characteristic allows for the design of a compliant mechanism that exhibits a distinctive load-displacement relationship, resembling that of a cubic nonlinear spring. The specific geometric configuration and material properties of the components play a crucial role in achieving this

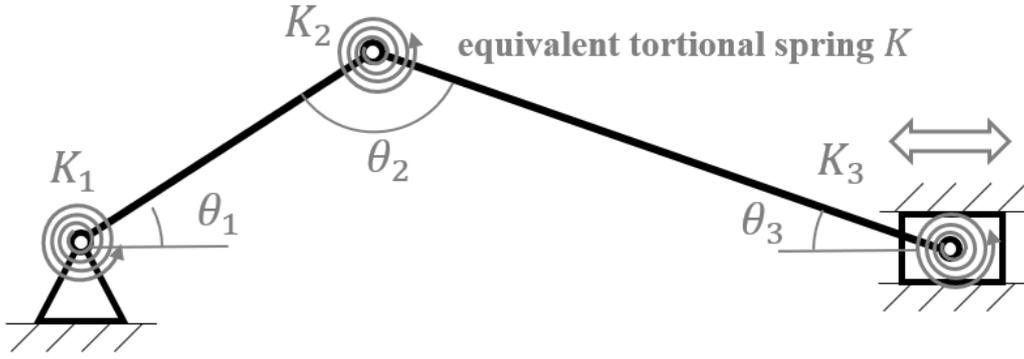


Figure 26: The PRB model of the crank slider compliant mechanism

desired behavior.

Creating a pseudo rigid-body (PRB) model for the crank slider mechanism, as shown in Figure 26. Since the flexures in this design are much shorter than the rigid bars, they can be approximated as torsion springs [19]. The torsion spring stiffness  $K$  satisfies:

$$K = \frac{EI}{l} = \frac{Ewt^3}{12l} \quad (21)$$

$E$  is the Young's modulus of the flexure material,  $w$  is the flexure width (out-of-plane thickness),  $t$  is the in-plane thickness of the flexure,  $l$  is the flexure length.

The elastic potential energy stored in the torsion spring is proportional to the square of the deflection angle  $\Delta\theta$ , as shown below:

$$E_p = \frac{1}{2}K(\Delta\theta)^2 \quad (22)$$

To provide a more comprehensive explanation, the example in Figure 23 is revisited. In this example, the maximum value of  $\theta_3$  is set to  $48.6^\circ$ , denoted as  $\theta_{3,0}$ . By adjusting the bending end of the rigid bar, the torque spring  $K_3$  remains undeformed when  $\theta_3 = \theta_{3,0}$ . Consequently, the energy stored in spring  $K_3$  is proportional to  $(\theta_3 - \theta_{3,0})^2$ . As depicted in Figure 27, the relationship between this value and the displacement of the slider closely resembles a quadratic curve.

The energy-displacement curve of a standard cubic spring also exhibits a similar fourth power relationship. Therefore, the crank-slider mechanism demonstrates characteristics analogous to a cubic spring in certain aspects. Although the curve of  $K_3$  may not perfectly align with a quartic curve, it is possible to compensate for any discrepancies by performing similar operations on

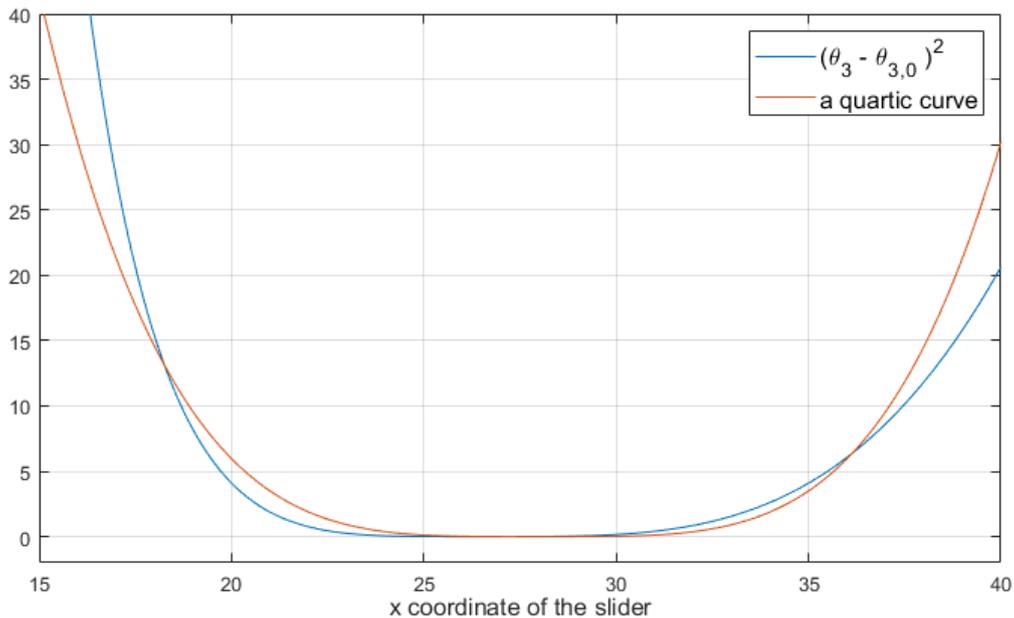


Figure 27: The square of deflection angle and a quartic curve for reference

$K_1$  and  $K_2$ . This can be achieved by adjusting the length and thickness of the corresponding flexures, as the stiffness of the torque springs can be fine-tuned during the design process.

Hence, employing a similar approach to the previously discussed spline mechanism, it is possible to generate a slider-crank mechanism that exhibits a cubic displacement-load relationship.

In contrast to the spline-shaped model, the performance of the crank-slider mechanism is particularly sensitive to the thickness of the flexures. Therefore, it is recommended to utilize femtosecond laser cutting due to its high precision. This fabrication method enables the production of flexures with the desired thickness, ensuring the accurate and reliable functioning of the slider-crank mechanism.

**Optimization implementation** The crank-slider mechanism is optimized using the GA method similar to the previous spline-shaped design. There are two ways to optimize this crank-slider mechanism. In this thesis, the second approach is applied.

The first method involves optimizing the design variables, including the stiffness and initial deflection angle of the three torsion springs, as well as the lengths of the linkages in the crank-slider mechanism. By conducting force analysis and mathematical derivations, the theoretical load-displacement curve of the system's PRB model under these parameters can be generated. The objective function of the optimization process is the degree of similarity between these curves and a standard cubic curve. The Genetic Algorithm (GA) method is employed to optimize the performance and determine the best combination of design variables. The CAD

design of the model is manually derived based on the performance parameters (stiffness and initial deflection angle) of the torsion springs, utilizing Equation 21 and Figure 25. Finite element simulation using COMSOL is performed to finalize the optimized design.

The advantage of the first method is its faster execution as it does not involve finite element analysis in the GA optimization stage. However, it does not consider stress calculations and feasibility evaluation during the optimization phase. This can lead to optimized results that may require flexures with extremely high or low stiffness, which may not be achievable in practice. Moreover, due to the discrepancies between the pseudo rigid-body (PRB) model and the actual model, the finite element analysis results of the optimized design may deviate significantly from the desired cubic relationship. Therefore, multiple rounds of iterations between geometric design adjustments and performance testing might be necessary, which can be a cumbersome process.

The second approach involves directly optimizing the geometric parameters of the crank-slider mechanism model. In this method, the stiffness and initial deflection of the equivalent torsion springs are treated as black box parameters, meaning they are not directly accessed or modified during the optimization process. Instead, the focus is solely on optimizing the geometric parameters of the mechanism.

The basic COMSOL model of the crank-slider mechanism, as shown in Figure 28, includes various geometric parameters that define the lengths, thicknesses, and angles of the components. These parameters are essential for the subsequent optimization process.

The flexible segments, denoted by  $L_1$ ,  $L_2$ , and  $L_3$ , represent the lengths of the flexures in the mechanism. The in-plane thicknesses of these flexures are represented by  $t_1$ ,  $t_2$ , and  $t_3$ , respectively.

The rigid bars in the mechanism have bending ends to ensure that the flexures remain undeformed when the ground and bars have prescribed angles between them. The lengths of these compensatory bending ends are denoted by  $L_0$ , and they have a uniform length throughout the mechanism. But if necessary, they can also be made optimization variables to provide additional design flexibility, but need to specify the minimum length. This minimum length ensures that the mechanism remains feasible and can be fabricated successfully.

The lengths of the rigid bars themselves are represented by  $L_A$  and  $L_B$ . It is important to ensure that these bars are considered rigid, so their in-plane thickness  $t_0$  must be sufficiently large.

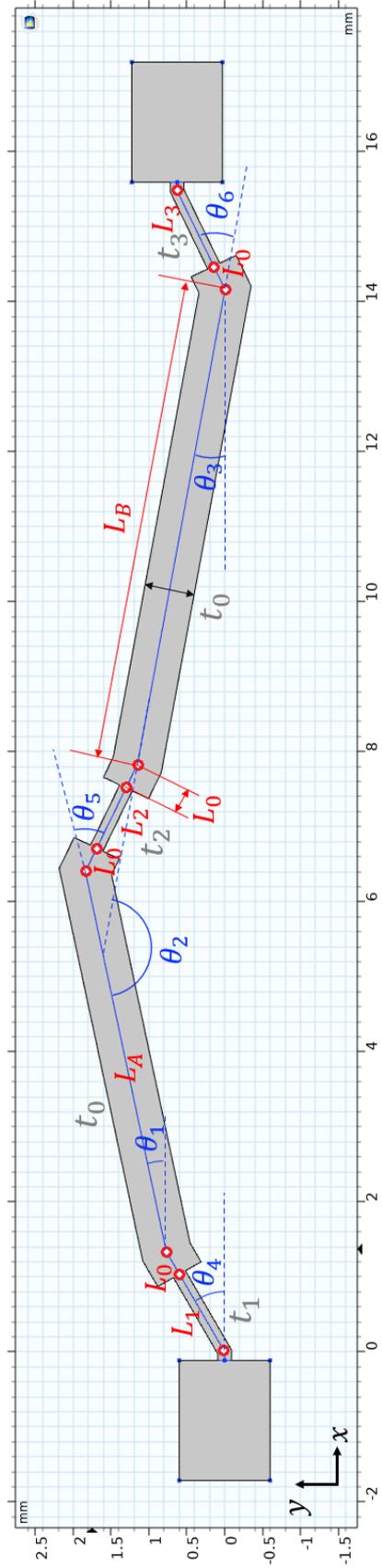


Figure 28: The design of the crank slider mechanism and its geometry variables

The angles between the ground and bar A, bar B and the ground, and bar A and bar B are denoted by  $\theta_1$ ,  $\theta_3$ , and  $\theta_2$ , respectively. Note that  $\theta_2$  is not an optimization variable, as it can be calculated based on the other two angles.

The angles  $\theta_4$ ,  $\theta_5$ , and  $\theta_6$  represent the deflection angles of the flexures relative to the ground, bar A, and bar B, respectively. These angles are limited to a range of  $-45^\circ$  to  $+45^\circ$  to ensure proper model generation and avoid any abnormal behavior.

Finally, the width  $w$  of the mechanism, representing the out-of-plane thickness, is determined based on the requirements of the laser cutting process and the thickness of the plate used. However, this value does not affect the quality of the nonlinear spring. While the thickness may impact the absolute values of the forces and displacements involved, it does not alter the fundamental cubic relationship between load and displacement. Thus, variations in plate thickness can be considered as a scaling factor without fundamentally altering the behavior of the spring.

The basic COMSOL model serves as the starting point for the subsequent optimization process, where the geometric parameters will be adjusted to achieve the desired cubic load-displacement relationship.

Due to the anisotropy of the silicon wafer material, a coordinate system of crystal orientation needs to be assigned to the material property in COMSOL. A (100) silicon wafer is utilized in this thesis, which means that the (100) cubic crystal plane is parallel to the wafer surface. The  $x$  and  $y$  directions in the design diagram Figure 28 correspond to  $x$  (crystal direction  $[110]$ ) and  $y$  (crystal direction  $[\bar{1}10]$ ) in Figure 29. The terms " $x$ " and " $y$ " are the orthogonal directions in the cubic crystal coordinates.

Similar to the COMSOL model for the spline-shaped mechanism, two cuboids are connected to the crank-slider mechanism. These cuboids serve as load receivers and platforms for applying supports and fixing surfaces. A load sweep from  $-F$  to  $F$  is applied to the model, and the corresponding deformation and von Mises stress of the object are calculated at each load increment. Probes are used to record the displacement of the right cuboid along the  $x$  direction and the maximum von Mises stress in the entire model.

The design optimization process employs a genetic algorithm, similar to that used for the spline-shaped mechanism. The design variables include the lengths ( $L_1, L_2, L_3, L_A, L_B$ ), thicknesses ( $t_1, t_2, t_3$ ), and angular parameters ( $\theta_1, \theta_3, \theta_4, \theta_5, \theta_6$ ). Other important parameters, which are not design variables, include  $L_0, t_0, w, F$ , mesh density, and material parameters. The

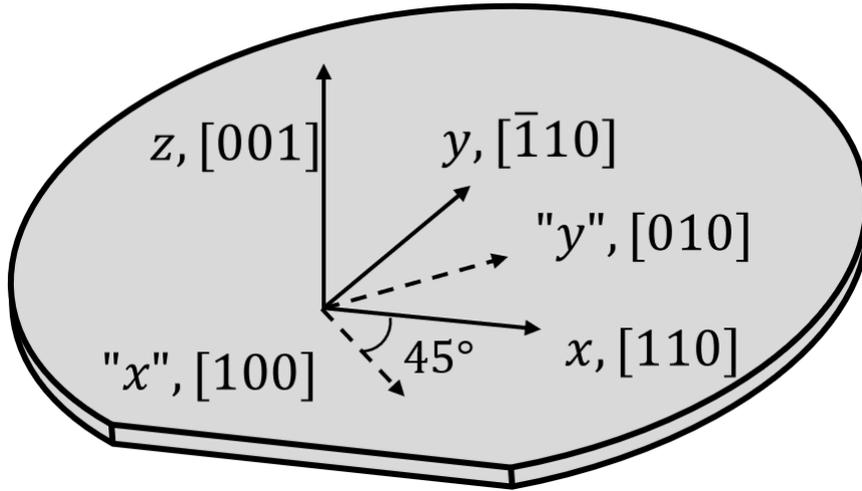


Figure 29: The crystal orientation of the silicon wafer

objective function is defined in the same manner as Equation 17-20.

It is also worth noting that since the laser beam of the cutter has diameter, the sharp edges of the original design will be fillets after fabrication. However, adding fillets to a model can drastically increase the number of meshes, thus increasing CPU hours. To study the effect of fillet on FEM results, fillets with 30  $\mu\text{m}$  diameter are added to a model. The mesh of the model is shown in Figure 30. The model without fillet has 2,574 mesh elements and the FEM simulation takes 12.50 s. The model with fillet has 61,727 elements and the FEM simulation takes 319.59 s. The simulation results are shown in Figure 31. It can be seen that adding fillet has very little effect on the load-displacement curve of the model. But adding fillet will greatly increase the simulation time. Although the existence of fillet will affect the maximum von Mises stress, but the law is still very obvious. Under different loads, the maximum von Mises stress of the model without fillet is usually about 60 % of that with fillet. Therefore, the maximum von Mises stress of this without-fillet model can be used to estimate the real situation. Also, the stress can be rechecked when the final optimization result is generated. Therefore, adding fillets during GA optimization is unnecessary.

The second method, which involves optimizing the geometric parameters of the model, introduces a larger number of design variables compared to the first method that focuses on torsion spring stiffness, deflection, and bar length. Consequently, the chromosomes in the genetic algorithm become longer, requiring a larger population size to explore a wider design space. Additionally, the evaluation of each individual's fitness necessitates performing finite element simulations, resulting in increased computational time for the GA optimization process.

However, this method offers several advantages. Firstly, it eliminates the need for calculating

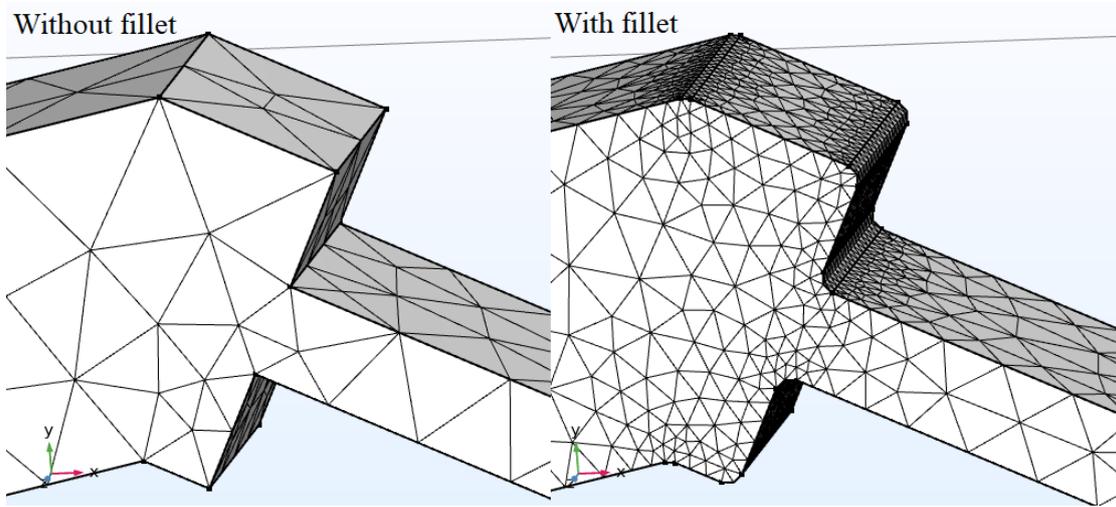


Figure 30: The mesh of the model without fillet and with fillet

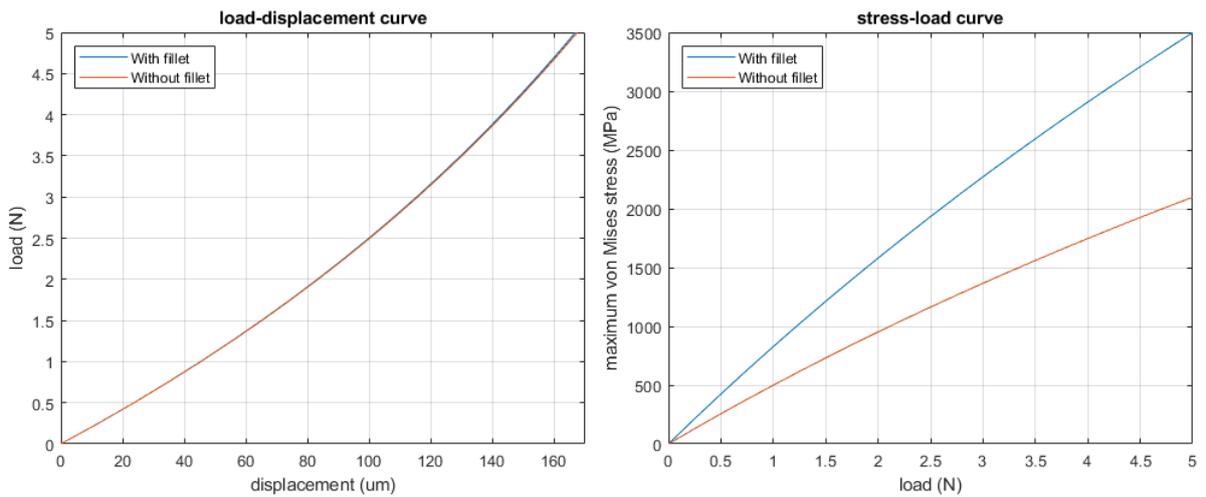


Figure 31: The simulation results of the model without fillet and with fillet

geometric parameters based on performance parameters and manually constructing a 3D model, reducing the manual effort required. Secondly, the designs generated through this method guarantee mechanical feasibility since the optimization process considers geometric constraints. Moreover, the inclusion of finite element analysis allows for the assessment of stress levels, enabling the quick identification of designs that cannot withstand excessive loads and their elimination from the optimization process.

### 3.3 Nonlinear spring design embodiment

Both design methods for the nonlinear spring can generate a multitude of models based on the provided basic parameters, However, the load-displacement curves of these springs do not form a perfect cubic curve. Notably, there are still non-negligible linear terms present, While the coefficient of the cubic term is adjustable, the range of modification is relatively limited.

#### 3.3.1 Result of spline-shaped mechanism nonlinear spring

The optimization process is performed for a nonlinear spring with a length of 132 mm along the  $x$  direction. The overall width (out-of-plane thickness) of the mechanism was set to 5 mm. The maximum applied load on the spring in the  $x$  direction was 11.25 N.

The number of splines in the system is set to be determined by the optimizer, with a maximum of 2. In order to simplify the design process and prevent excessive complexity, the configuration of the secondary spline involved manual adjustments. Specifically, one end of the secondary spline was fixed to connect with the middle control point of the primary spline, while the other end was left unattached. Additionally, the secondary spline was designed to have only four control points. The range within which these four control points were generated was limited to a smaller rectangular region centered around the middle control point of the primary spline. These constraints were implemented to reduce the number of design variables and ensure that the generated designs remained within a manageable level of complexity. By controlling the configuration of the secondary spline in this manner, the design process was made more efficient and the resulting designs were easier to handle.

Following the optimization process, a design was obtained, as shown in Figure 32. The spring was designed using a double-spline configuration. The secondary spline appears on one side of the primary spline. The spline partially overlaps with the primary spline, but does not intersect. The primary spline had a thickness (in-plane thickness) of 2.10 mm, while the secondary spline had a width of 1.75 mm.

The full load-displacement curve of the spring is presented in Figure 33, revealing a fitted curve that consists of both linear and cubic terms. The linear term, denoted as  $P_1$ , was found to be 188.89 N/m, while the cubic term, denoted as  $P_2$ , was determined to be 46,354 N/m<sup>3</sup>. Notably, when a load of +6.3 N or -6.7N was applied, the maximum von Mises stress reached the yield point of 124 MPa. Within the yield stress, the mean absolute error (MAE) of the FEM load and fitting curve is 0.256N, the mean absolute percentage error (MAPE) is 5.8%.

Despite undergoing multiple iterations, the design method did not result in a perfect design that

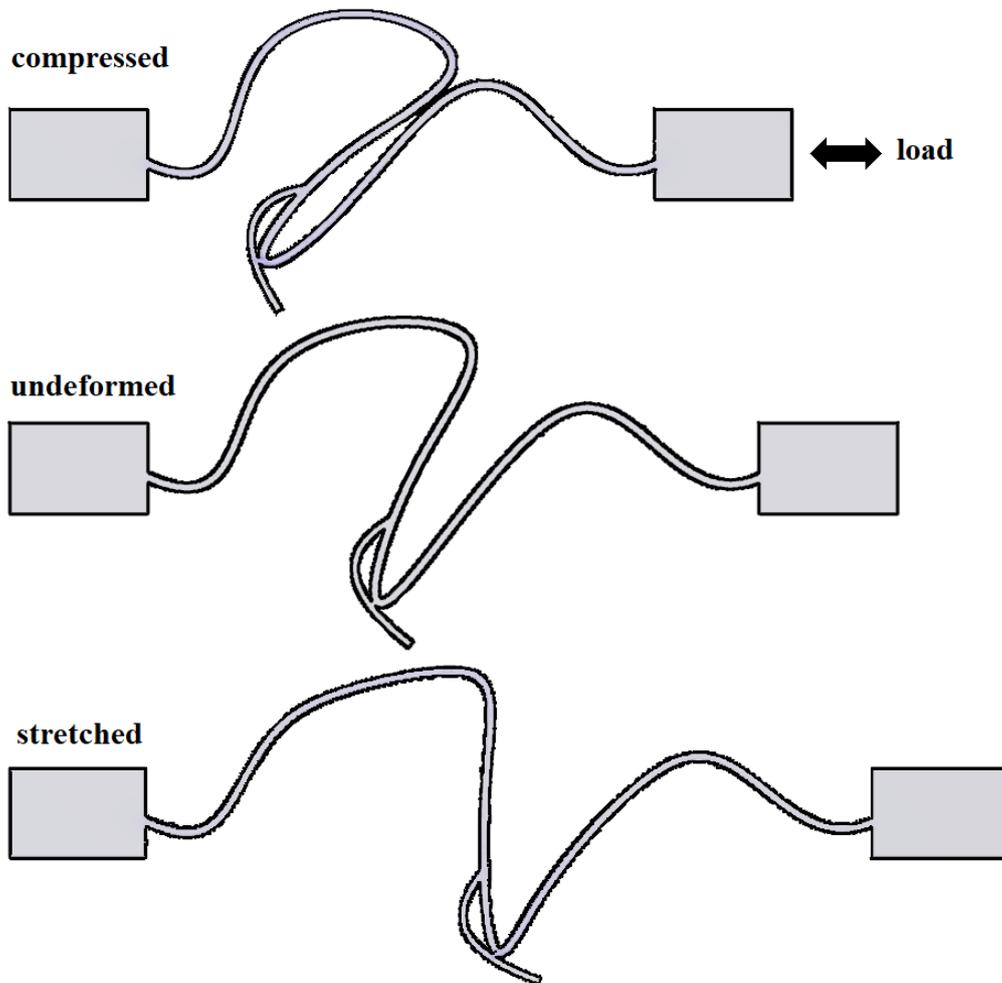


Figure 32: Generated spline-shaped nonlinear spring, the "small tail" in the middle of the primary spline is the secondary spline

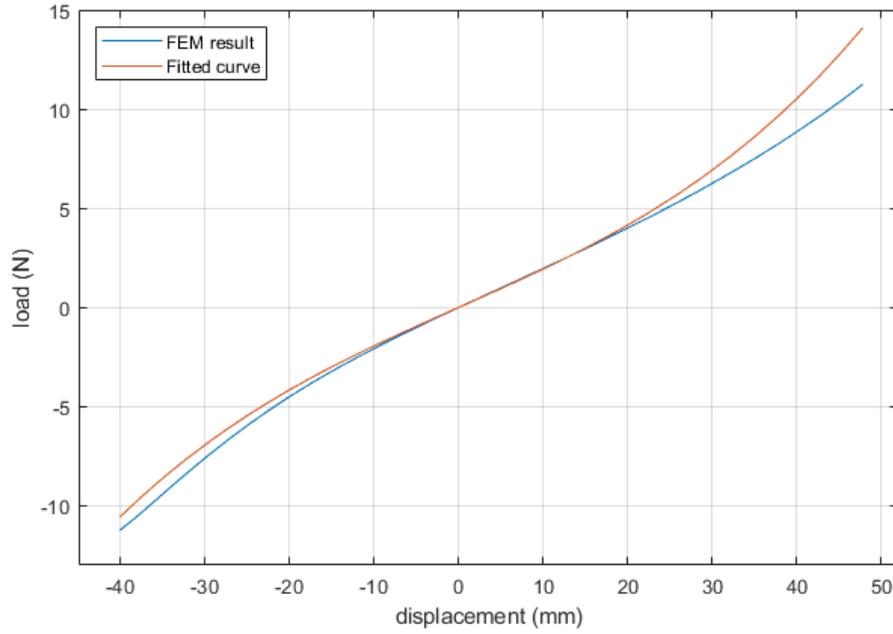


Figure 33: The load-displacement curve of the generated nonlinear spring design

exclusively exhibits a dominant cubic behavior in the load-displacement curve, with negligible contributions from other polynomial terms. The generated designs still possess non-negligible linear terms, indicating a partial linearity in their load-displacement characteristics. Further refinement of the design principle is necessary. Potential methods for improvement include increasing the number of splines, exploring different materials, adjusting the global size of the model, or enlarging the population size in the genetic algorithm. However, due to time constraints, these methods were not investigated in this study.

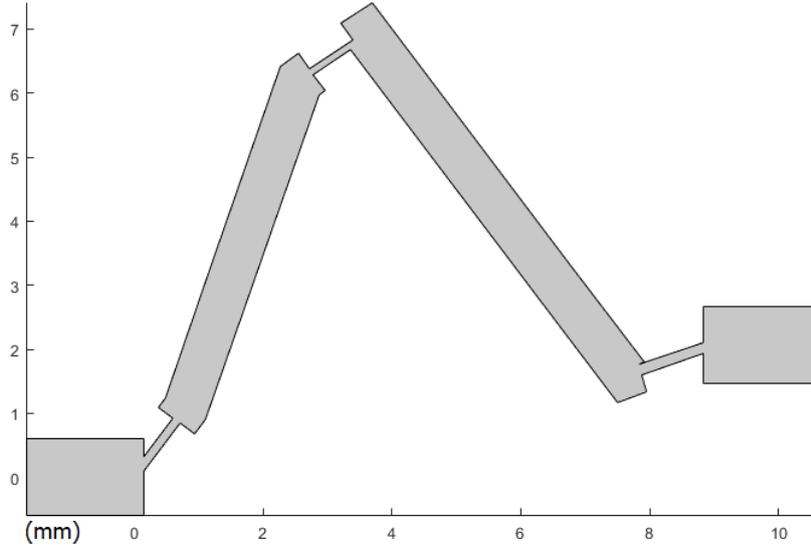


Figure 34: The optimized crank slider mechanism nonlinear spring design

### 3.3.2 Result of crank slider mechanism nonlinear spring

The optimization process was conducted for a crank slider mechanism fabricated from a  $525\ \mu\text{m}$  thick silicon wafer. The yield criterion for single crystal silicon was set at  $1.23\ \text{GPa}$  [20]. The cross-section of the mechanism was constrained to a rectangular shape with dimensions of  $15\ \text{mm}$  in length and  $8\ \text{mm}$  in width. A maximum load of  $10\ \text{N}$  was applied to the spring in the  $x$ -direction. Following the optimization, a design was obtained as depicted in Figure 34, featuring flexure thicknesses ranging from  $102$  to  $170\ \mu\text{m}$  and flexure lengths between  $807$  to  $1,208\ \mu\text{m}$ . The lengths of bars A and B were determined as  $5,420\ \mu\text{m}$  and  $6,673\ \mu\text{m}$ , respectively.

The resulting load-displacement curve of the spring, displayed in Figure 35, was fitted with a linear term of  $P_1 = 394.4\ \text{N/m}$  and a cubic term of  $P_2 = 3.44 \times 10^7\ \text{N/m}^3$ . The quality of the fit was also evaluated using the mean absolute error (MAE) and the mean absolute percentage error (MAPE), resulting in an MAE of  $1.48\ \text{N}$  and an MAPE of  $30\%$  between the FEM load and the fitted curve.

However, it is worth noting that the load-displacement curve exhibited asymmetry around the origin, as depicted in Figure 35. Despite multiple optimization iterations and design updates, this issue could not be completely resolved. Additionally, due to the high Young's modulus of silicon, significant deformation of the spring required a substantial load, which in turn led to high internal stresses exceeding the yield stress. For instance, even applying a displacement of  $800\ \mu\text{m}$  to the system in Figure 34 would reach the yield point, where the linear term still dominated the load-displacement relationship. Consequently, achieving a highly satisfactory design for a crank slider mechanism with a  $525\ \mu\text{m}$  silicon wafer of size  $15 \times 8\ \text{mm}$  proved

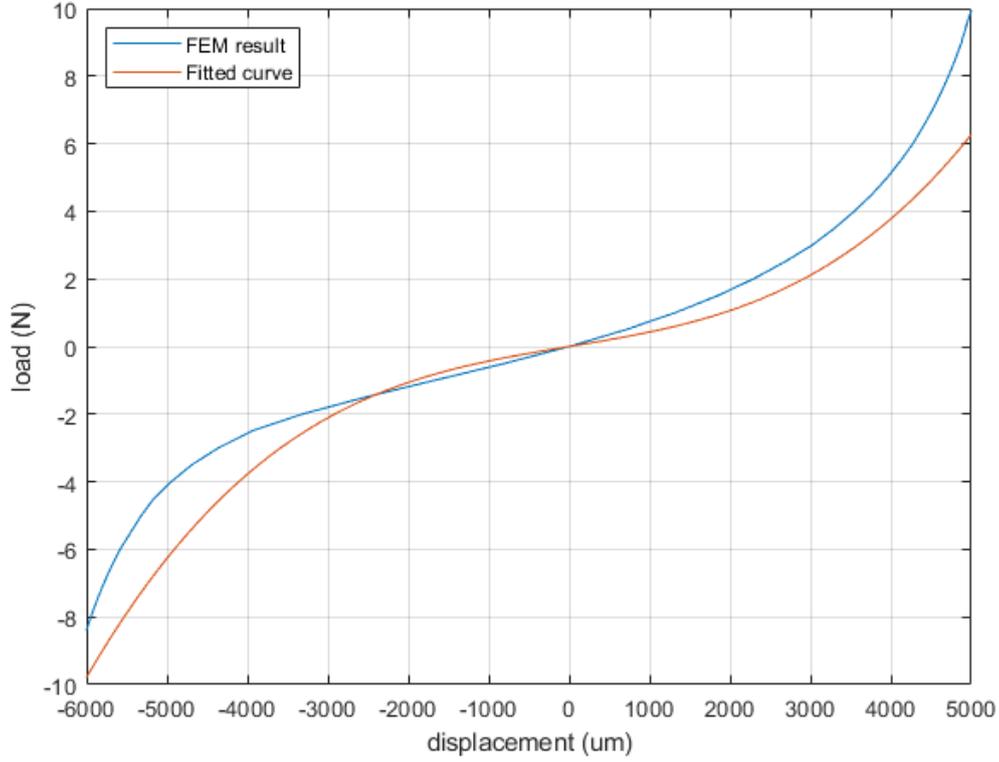


Figure 35: The load-displacement curve of the crank slider mechanism nonlinear spring design challenging under the current design strategy.

Upon inspecting the optimization results, it was observed that the majority of the designs exhibited improved cubic properties when subjected to purely tensile forces. Specifically, the load-displacement curves of the springs in the positive half of the x-axis demonstrated closer resemblance to a standard cubic curve. A design exemplifying favorable tension cubic properties is presented in Figure 36, where the fitted curve in Figure 37 featured a linear term of  $P_1 = 4,321 \text{ N/m}$  and a cubic term of  $P_2 = 1.06 \times 10^{10} \text{ N/m}^3$ . In this particular model, yielding takes place at a specific point where the displacement reaches  $240 \text{ }\mu\text{m}$ . Evaluating its load-displacement curve, the mean absolute error (MAE) was  $0.58 \text{ N}$ , with a mean absolute percentage error (MAPE) of  $5.0\%$ . Furthermore, by solely considering the portion of the curve where the load remained below  $7.8 \text{ N}$  for analysis, the MAE reduced to  $0.052 \text{ N}$ , with an MAPE of  $1.1\%$ . These results were significantly smaller in comparison to the analysis considering both tension and compression (MAE= $1.48 \text{ N}$ , MAPE= $30\%$ ).

Therefore, if the symmetry of load-displacement curve is ignored or not required, the crank slider mechanism can achieve high-quality cubic property when only tension load is applied.

In conclusion, based on the size and material limitations of the current study, the design strategy employed for the crank slider mechanism did not yield a perfect design that exhibits

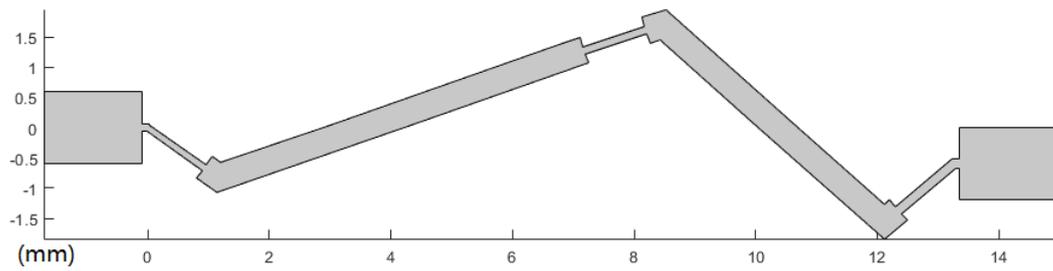


Figure 36: A crank slider mechanism design with good tension cubic property

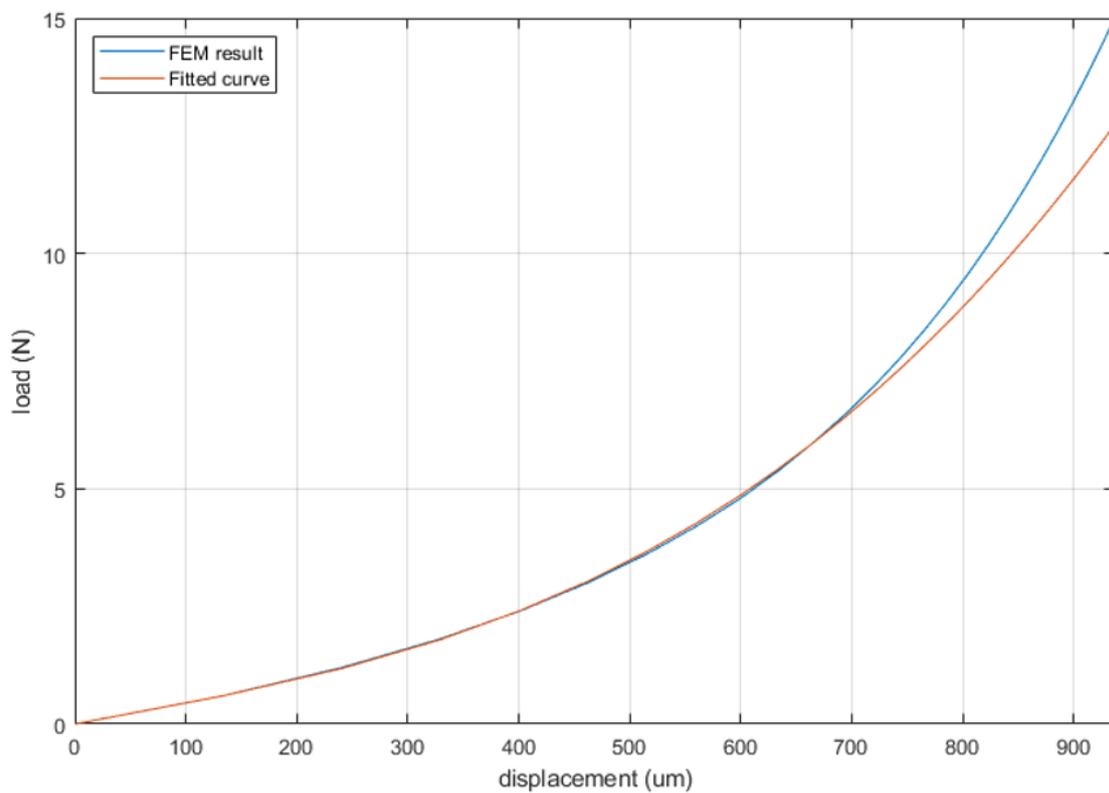


Figure 37: The load-displacement curve of a crank slider mechanism design with good tension cubic property

consistent cubic characteristics in both tension and compression directions. However, this design approach remains a promising idea for the development of nonlinear springs with load-displacement relationships other than cubic, or for cubic springs that predominantly experience tensile loads. Further exploration and refinement of the design strategy may lead to improved results in the future.

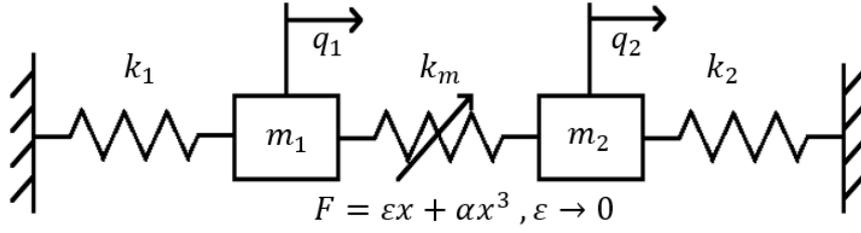


Figure 38: Schematic diagram of the 2-DoF system

### 3.4 Mode coupling

This section aims to investigate the potential of the designed nonlinear spring to facilitate mode coupling within a 2-DoF system. For this purpose, a mathematical model of a 2-DoF system connected by a purely nonlinear spring will be developed and analyzed. The conditions that could lead to mode coupling is also investigated.

#### 3.4.1 Equation of motion

The eventual goal of this thesis is to incorporate the nonlinear spring design discussed in previous sections in a 2-DoF system. For this system, both masses ( $m_1$  and  $m_2$ ) are connected to ground with two linear spring  $k_1$  and  $k_2$ . The two springs are actually two position stage based on leaf springs that makes two masses move only in  $x$  direction. The displacement of  $m_1$  and  $m_2$  in  $x$  direction are represented by  $q_1$  and  $q_2$ , respectively. A cubic linear spring with a very small linear term is used to couple the two masses.  $F_1$  and  $F_2$  are the load applied on  $m_1$  and  $m_2$ . The schematic of the system is shown in Figure 38. And the dynamics of such system can be explained using the following set of equations:

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} k_1 + \varepsilon & -\varepsilon \\ -\varepsilon & \varepsilon + k_2 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} + \begin{bmatrix} \alpha (q_1 - q_2)^3 \\ -\alpha (q_1 - q_2)^3 \end{bmatrix} = \begin{bmatrix} F_1(t) \\ F_2(t) \end{bmatrix} \quad (23)$$

To transform the system to modal space, the modal matrix  $[X]$  is introduced. The real coordinates  $q$  and modal coordinates  $\eta$  satisfy the following equation:

$$\begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = [X] \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} \quad (24)$$

Using the transformation Equation 24, Equations 23 can be transformed to the following system of equations, where the linear coupling coefficients are cancelled:

$$\begin{bmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{bmatrix} \begin{bmatrix} \ddot{\eta}_1 \\ \ddot{\eta}_2 \end{bmatrix} + \begin{bmatrix} \gamma_1 & 0 \\ 0 & \gamma_2 \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} + \begin{bmatrix} x_{(1)}^T & f_{nl}/\mu_1 \\ x_{(2)}^T & f_{nl}/\mu_2 \end{bmatrix} = 0 \quad (25)$$

$f_{nl}$  is the nonlinear term,  $\mu$  is modal mass and  $\gamma$  is modal stiffness. Then, the ratio of resonance frequencies satisfies:

$$\frac{\omega_2}{\omega_1} = \frac{\sqrt{\gamma_2/\mu_2}}{\sqrt{\gamma_1/\mu_1}} \quad (26)$$

Equation 26 is a very complex formula that involves many variables and parameters. However, when the linear term of the cubic spring becomes very small. The frequencies will satisfy:

$$\varepsilon \rightarrow 0, \omega_1 \rightarrow \sqrt{\frac{k_1}{m_1}}, \omega_2 \rightarrow \sqrt{\frac{k_2}{m_2}} \quad (27)$$

The ratio is then highly related to the eigenfrequency of one mass block. In this way, if the nonlinear spring linearity is very small, an internal resonance with prescribed ratio can be easily achieved by designing the system's mass and stiffness.

In order to make the equations of motion suitable for numerical integration, the equations are made to be dimensionless by considering  $Q$  as the characteristic length and  $\omega_1$  as the characteristic time as follows:

$$\tilde{x} = x/Q, \tau = \omega_1 t \quad (28)$$

Therefore, Equation 23 will become

$$\begin{cases} \ddot{\tilde{x}}_1 + \frac{(k_1+k_m)}{\alpha Q^2} \tilde{x}_1 - \frac{k_m}{\alpha Q^2} \tilde{x}_2 + (\tilde{x}_1^3 - \tilde{x}_2^3) = \frac{F_1(t)}{Q^3 \alpha} \\ \ddot{\tilde{x}}_2 + \frac{m_1(k_2+k_m)}{m_2 \alpha Q^2} \tilde{x}_2 - \frac{m_1 k_m}{m_2 \alpha Q^2} \tilde{x}_1 - \frac{m_1}{m_2} (\tilde{x}_1^3 - \tilde{x}_2^3) = \frac{m_1 F_2(t)}{m_2 Q^3 \alpha} \end{cases} \quad (29)$$

Taking into account the inherent damping effect present in the prototype, an examination of the system with the inclusion of a damping term is also conducted. This is shown in Figure 39. The equation of motion is accordingly modified and represented as followed:

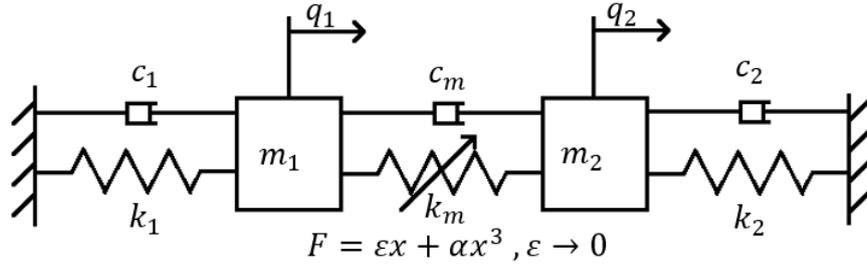


Figure 39: Schematic diagram of the mass-spring-damper mechanism

$$\begin{aligned}
 & \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} c_1 + c_m & -c_m \\ -c_m & c_2 + c_m \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} k_1 + \varepsilon & -\varepsilon \\ -\varepsilon & \varepsilon + k_2 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \\
 & + \begin{bmatrix} \alpha (q_1 - q_2)^3 \\ -\alpha (q_1 - q_2)^3 \end{bmatrix} = \begin{bmatrix} F_1(t) \\ F_2(t) \end{bmatrix} \quad (30)
 \end{aligned}$$

Similarly, the nondimensionalized version of Equation 30 becomes:

$$\begin{cases} \ddot{\tilde{x}}_1 + \frac{(c_1+c_m)}{Q\sqrt{\alpha m_1}} \dot{\tilde{x}}_1 - \frac{c_m}{Q\sqrt{\alpha m_1}} \dot{\tilde{x}}_2 + \frac{(k_1+k_m)}{\alpha Q^2} \tilde{x}_1 - \frac{k_m}{\alpha Q^2} \tilde{x}_2 + (\tilde{x}_1^3 - \tilde{x}_2^3) = \frac{F_1(t)}{Q^3 \alpha} \\ \ddot{\tilde{x}}_2 + \frac{(c_2+c_m)}{m_2 Q} \sqrt{\frac{m_1}{\alpha}} \dot{\tilde{x}}_2 - \frac{c_m}{m_2 Q} \sqrt{\frac{m_1}{\alpha}} \dot{\tilde{x}}_1 + \frac{m_1(k_2+k_m)}{m_2 \alpha Q^2} \tilde{x}_2 - \frac{m_1 k_m}{m_2 \alpha Q^2} \tilde{x}_1 - \frac{m_1}{m_2} (\tilde{x}_1^3 - \tilde{x}_2^3) = \frac{m_1 F_2(t)}{m_2 Q^3 \alpha} \end{cases} \quad (31)$$

For the 2-Dof mass-spring-damper system in this thesis, assuming only a sinusoidal excitation  $F = F_0 \sin(t)$  is applied to the mass  $m_1$ . By defining  $y_1 = \sin(\omega t)$  and  $y_2 = \cos(\omega t)$ , the system's response satisfies:

$$\begin{aligned}
 \dot{y}_1 &= y_1 + \omega y_2 - y_1 (y_1^2 + y_2^2) \\
 \dot{y}_2 &= y_1 - \omega y_2 - y_2 (y_1^2 + y_2^2) \\
 dx_1/dt &= \dot{x}_1 \\
 dx_2/dt &= \dot{x}_2 \\
 d\dot{x}_1/dt &= (-(c_1 + c_m) \dot{x}_1 + c_m \dot{x}_2 - (k_1 + \varepsilon) x_1 + \varepsilon x_2 - \alpha (x_1 - x_2)^3 + F y_1) / m_1 \\
 d\dot{x}_2/dt &= (-(c_2 + c_m) \dot{x}_2 + c_m \dot{x}_1 - (k_2 + \varepsilon) x_2 + \varepsilon x_1 + \alpha (x_1 - x_2)^3 + 0) / m_1
 \end{aligned} \quad (32)$$

### 3.4.2 Numerical simulation

In this subsection, Simulink Simscape software is used to model the full system, and the theoretical response of the system to different input signals under different design parameters is calculated and analyzed. The ODE solver used in the simulation is Ode23t, which is well

suiting for moderate stiff problems and does not lead to numerical damping [21]. With this solver, energy will not be reduced by numerical damping during simulations.

To effectively illustrate the impact of ideal mode coupling, a simulation is performed on a system with predetermined parameters. These parameters are chosen to exhibit an ideal integer relationship for the convenience of analysis and may not correspond to the actual prototype data. The system parameters are as follows: the masses  $m_1$  and  $m_2$  are both 0.01 kg, the linear spring stiffness values are  $k_1 = 39.4784$  N/m and  $k_2 = 191.076$  N/m, the linear term of the nonlinear spring is  $\varepsilon = 0.05$  N/m, the cubic term is  $\alpha = 80$  N/m<sup>3</sup>, and the damping coefficients  $c_1$  and  $c_2$  are both 0.

Therefore, the eigenvectors of the system are determined as  $[X_1] = [1.0000, 0.0003]$  and  $[X_2] = [-0.0003, 1.0000]$ , corresponding to the eigenfrequencies of 10.0 Hz and 22.0 Hz, respectively. The deliberate choice of the second eigenfrequency as 22 Hz is to avoid it being exactly double the frequency of 10 Hz, thereby preventing frequency overlap.

In the system's first mode,  $m_1$  vibrates at 10.0 Hz while  $m_2$  exhibits extremely small amplitude motion in the same direction as  $m_1$ . In the second mode,  $m_2$  vibrates at 22.0 Hz while  $m_1$  undergoes very small amplitude motion in the opposite direction to  $m_2$ . The initial displacements imposed on the two masses are 0.50000 m for  $m_1$  and 0.00015 m for  $m_2$ , implying that the system is driven in its first mode. The displacement response is presented in Figure 40. It is evident that the two objects oscillate either in the same or opposite directions at different instances.

A fast Fourier transform (FFT) is conducted on the response signals, as shown in Figure 41. The analysis reveals a larger amplitude for the first mode at 10 Hz, signifying its significance as the dominant component of the overall signal. However, in the spectrum of  $x_2$ , a distinct peak is observed at 22 Hz, corresponding to the second mode. This observation indicates that the contribution of the second mode cannot be neglected. Therefore, it can be concluded that there exists coupling between the two modes, resulting in energy transfer between them.

This analysis method can be used in preliminary theoretical simulations of new designs, quickly check whether the design meets the requirements, and save calculation time for subsequent finite element iterations.

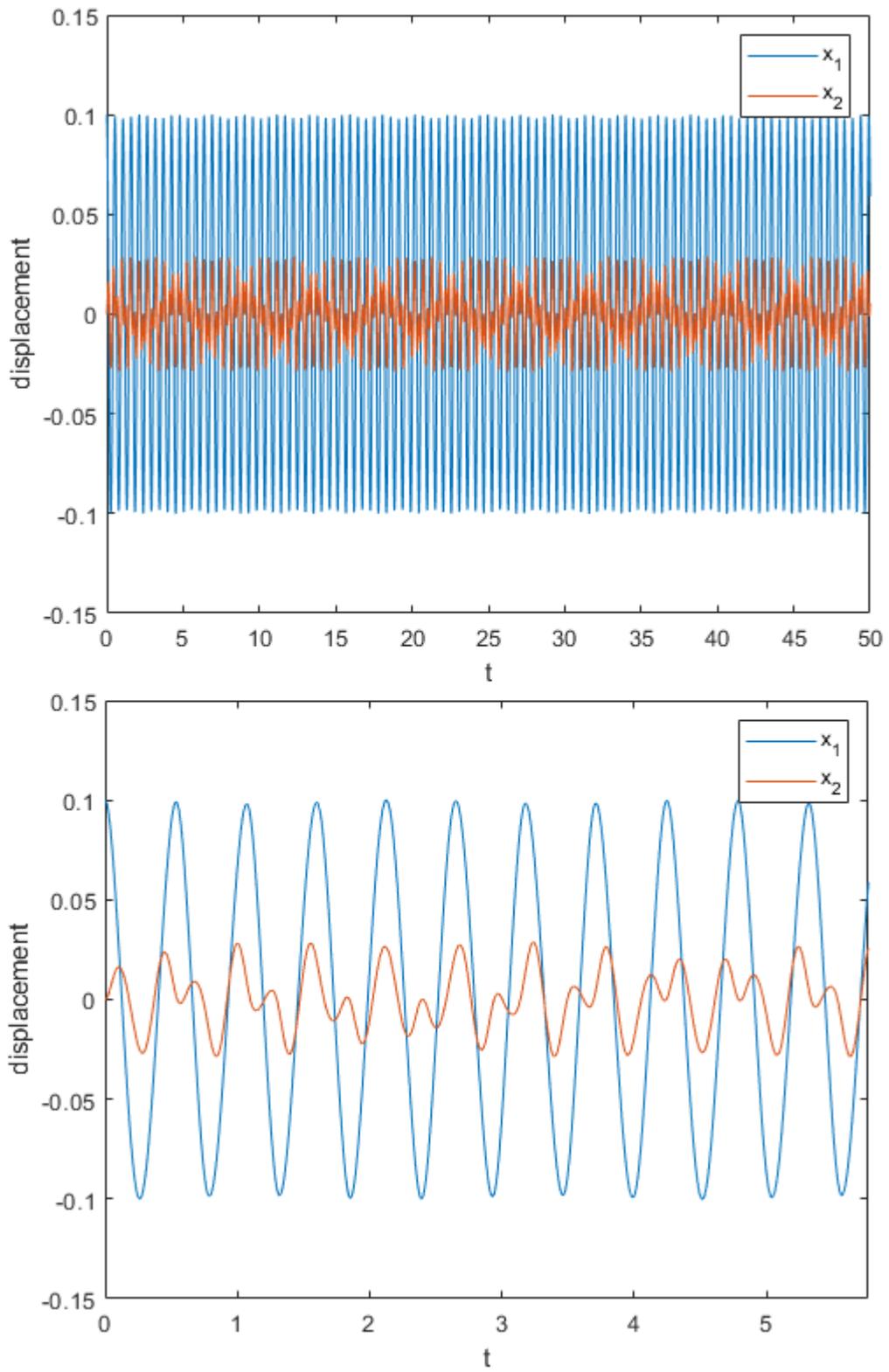


Figure 40: The displacement of two masses over time. The bottom image is an enlargement of the top image

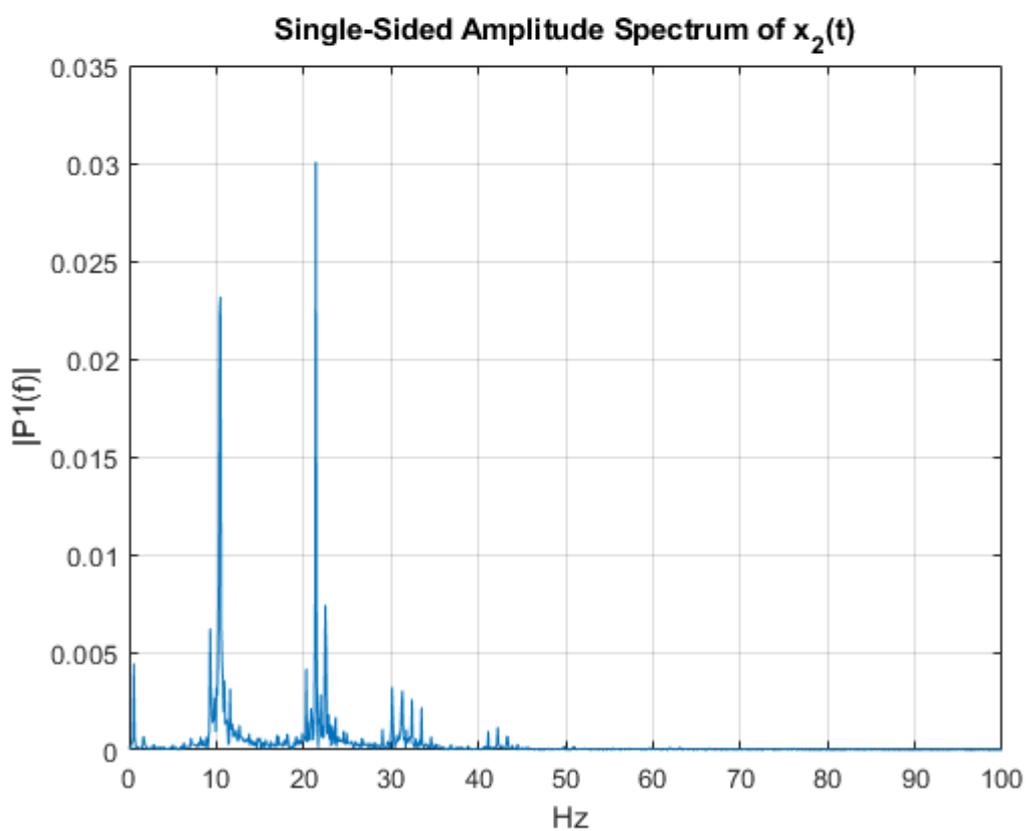
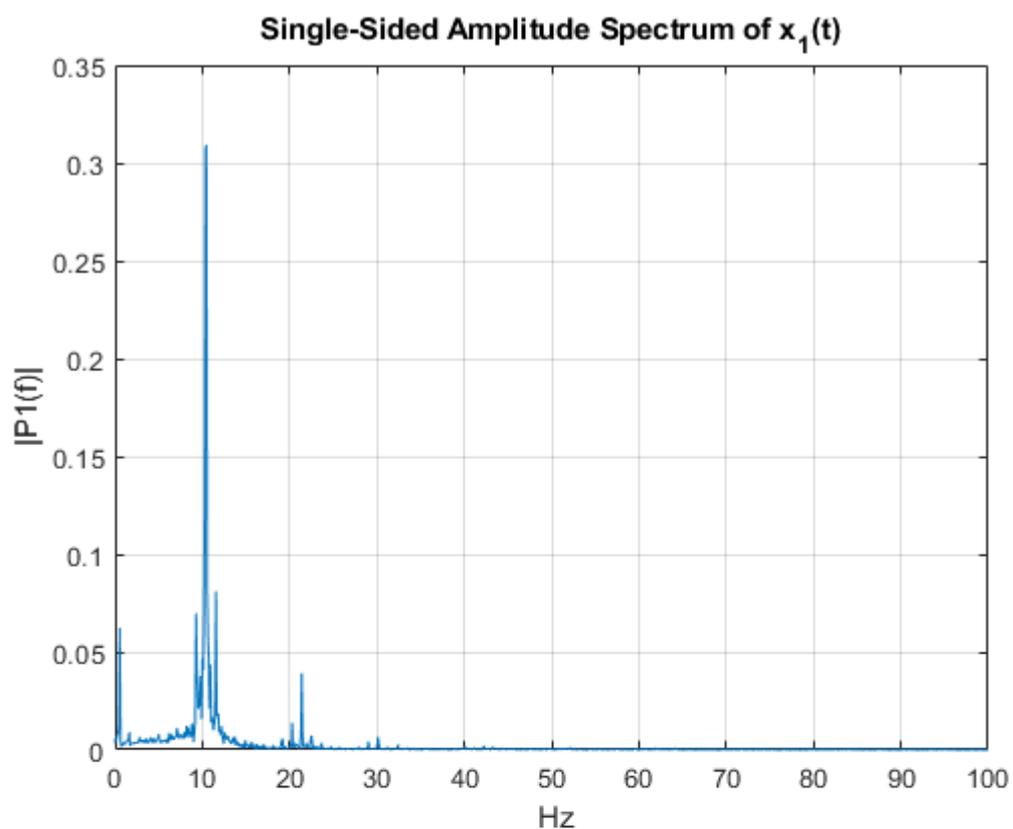


Figure 41: The time domain displacement FFT of two blocks. Please notice that the  $y$  axis-limits are different

## 3.5 Discussions and conclusions

Due to the limitations of time, equipment, and some of physical laws, many of the initial goals of this thesis were not finally realized. In this section, some challenges are discussed, along with their potential solutions and future research directions. Conclusions are also made in this section.

### 3.5.1 Challenges

Even though the theoretical calculations based on the obtained desired cubic nonlinearity showed sufficient coupling numerically, the practical demonstration in experiments was not achieved. The following are the challenges that need to be addressed in order to observe energy transfer through the engineering of nonlinear spring designs in practical applications:

**Optimization program** In the cubic spring design part, a considerable number of FEM calculations are called during the optimization process. To minimize CPU hours, it is preferable to employ small step sizes and a high mesh density in the FEM analysis. However, setting excessively coarse meshes or utilizing large step sizes can lead to non-convergence issues. Additionally, as mentioned earlier, the grid quality and model filleting can introduce errors in the determination of maximum von Mises stress. Achieving a balance between computational efficiency and optimization quality requires substantial iterative experimentation and fine-tuning efforts.

Furthermore, the investigation into the selection of the fitness function remains an area of focus. The load-displacement behavior exhibited by the spring presents a highly intricate curve, often necessitating the utilization of polynomial regression techniques of order 6-7 to achieve an optimal fit. The load-displacement curves of different springs may differ from the standard cubic curve in several ways. For example, some springs may have a larger linear term, and some springs may have a smaller linear term, but the curve as a whole is more like a 2.5 power curve. Methods to assess their quality are a challenge.

Additionally, the selection process for the so-called standard cubic curve utilized in fitness calculations also warrants exploration. Various approaches can be employed to generate the curves required for comparative analysis. For instance, one can derive simple cubic curves based on FEM data's maximum loads and displacements, using the relationship  $F_{\max} = \alpha x_{\max}^3$ . Alternatively, a standard cubic curve can be generated through polynomial regression. The similarity between the spring's FEM curve and the standard cubic curve can be quantified by

measuring the standard error between them. It is also possible to assign weights to errors observed at different points along the curves, corresponding to small and large loads.

Determining the linear term of the fitting curve can be accomplished through various techniques. One approach involves fitting the spring's load-displacement curve to a standard function incorporating solely linear and cubic terms. In this context, the linear term of the function can be regarded as the nonlinearity attributable to the spring.

When assessing the final fitness, both curve similarity and linear term value are of great importance, the weighting of them must be taken into consideration. Furthermore, it is possible that multiple sets of better designs will have very small difference in fitness, and appropriate methods need to be developed to increase discrimination.

In fact, for each specific given size and material instance, the optimal optimization settings (such as fitness function, number of generations, mesh density, load steps etc.) can vary. The process of determining the most appropriate parameters poses a significant challenge.

**Materials and processing equipment** The primary materials investigated in this thesis are resin and single crystal silicon. Resin is an advantageous choice for 3D printing due to its ease of use in model manufacturing; however, it exhibits a high damping ratio. Monocrystalline silicon possesses a larger Young's modulus and strength but has a limited capacity to withstand significant deformations. Consequently, achieving designed deformations and sustaining vibration becomes more challenging. Furthermore, during the design phase, the precision of the processing equipment becomes a crucial consideration. For instance, the currently available femtosecond laser cutting machine has a beam diameter of about 150  $\mu\text{m}$  and causes surface ablation, and the 3D printer also has a limit of 0.4 mm for the thinnest part, thereby imposing limitations on the minimum width of the flexure. These factors collectively constrain the potential of the design approach.

**Parameters picking** Additionally, for the theoretical calculations and simulations, variations in different system parameters (mass, linear spring stiffness, the linear and cubic term of the nonlinear spring) can exert a profound impact on the theoretical response. Figure 42 shows two example. When the initial displacement is too small, the disparity in maximum displacement between two masses may become excessively large, thereby obstructing effective observation. Similarly, increasing the values of masses alone necessitates adjusting the observation time span scale if the input initial displacement remains unchanged. Failure to do so may result in an insufficient number of vibration cycles for accurate FFT analysis. Hence, it becomes

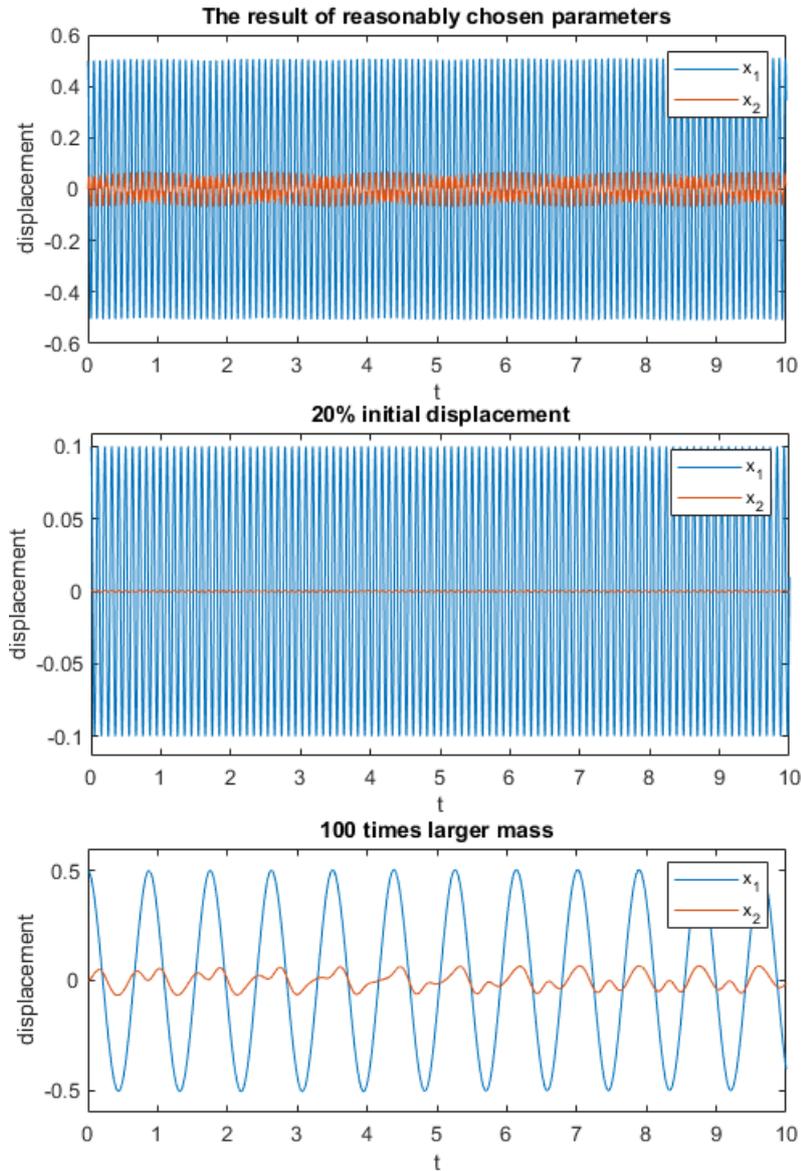


Figure 42: The response of systems with parameters of different orders of magnitude

imperative to preselect appropriate significant system parameters as design criteria for optimal spring design. Furthermore, careful consideration must be given to whether the nonlinear spring can endure these initial displacements without yielding.

**Linearity under small load** For the design of nonlinear springs, there is a prevalent issue related to the characteristics of their load-displacement curves. Figure 43 is an exaggerated schematic. In general, these curves share similarities with the standard cubic curve. However, during the initial section of the curve, specifically when the load is relatively small, the curve appears to be relatively flat, resulting in a more linear relationship between displacement and load. While the nonlinear component tends to dominate under larger loading conditions, the

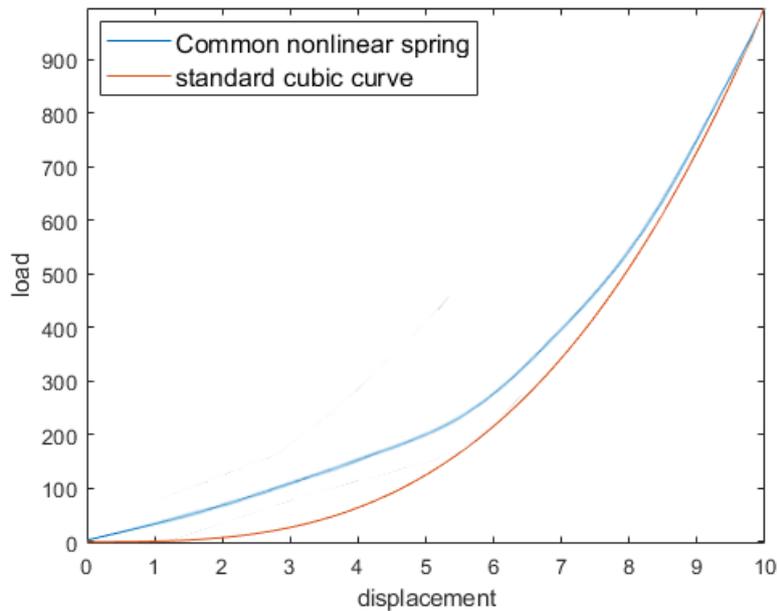


Figure 43: A exaggerated schematic showing the difference between a standard cubic curve and the load-displacement curve of a common nonlinear spring design

model is susceptible to yielding when subjected to excessively high loads. This behavior can be attributed to the fact that, at lower loads, the model experiences very deformation where the primary deformation is associated with simple stretching. Consequently, the significant deformations of the carefully engineered components (such as rigid bars and splines) of the model are not evident. To alleviate this problem, further optimization iterations can be carried out. The adoption of more precise manufacturing techniques to fabricate thinner flexures that are easier to deform is an alternative choice. Nevertheless, achieving a solution for this problem remains challenging.

**Asymmetry of the curve** For the obtained designs, the mechanical properties of nonlinear springs are not strictly symmetrical with respect to tension and compression. As shown in Figure 35, it is evident that the load-displacement curve is not strictly symmetrical about the origin. It is more difficult to achieve a nonlinear spring with good nonlinear behavior in both directions than in one side. The implications of this characteristic on mode coupling phenomena have yet to be investigated. This limitation can potentially restrict the applicability of certain precision systems that demand a high degree of symmetry. Since the deformation of the springs designs proposed in this thesis is inevitably asymmetrical in the tension and compression directions. This shortcoming may not be possible to eradicated. However, it is possible for other nonlinear spring designs to avoid this limitation by ensuring symmetrical deformations with respect to the initial position of the nonlinear spring.

### 3.5.2 Conclusions

A novel compliant mechanism with two major systems (nonlinear cubic spring and linear spring stage) was developed in this work. This mechanism is designed to achieve prescribed mode coupling. The whole workflow from setting aim, to modelling, and simulating is also developed.

Theoretically, when the cubic spring has low linear term and the linear stage parameters are well designed, the input energy can be transferred between the two modes, thus achieving designated mode coupling.

Two nonlinear spring design methods were developed. For the spline-shaped mechanism, in most cases, linear terms cannot be ignored, and the material requirements are very high. The crank-slider mechanism can achieve relatively good cubic load-displacement curve, but at that point, the yield point has been reached and it is therefore unavailable. However, when the symmetry of load-displacement curve is ignored or not required, the crank slider mechanism can achieve high-quality cubic property when only tension load is applied. Due to the commonality of the methods, they can also be used to design prescribed curves other than cubic ones.

## References

- [1] Kamala Rae Qalandar et al. “Frequency division using a micromechanical resonance cascade”. In: *Applied Physics Letters* 105.24 (2014), p. 244103.
- [2] XC Zhang et al. “Nanomechanical torsional resonators for frequency-shift infrared thermal sensing”. In: *Nano letters* 13.4 (2013), pp. 1528–1534.
- [3] Narjes Ghaemi, Amin Nikoobin, and Mohammad Reza Ashory. “A Comprehensive Categorization of Micro/Nanomechanical Resonators and Their Practical Applications from an Engineering Perspective: A Review”. In: *Advanced Electronic Materials* 8.11 (2022), p. 2200229.
- [4] Keivan Asadi, Jun Yu, and Hanna Cho. “Nonlinear couplings and energy transfers in micro-and nano-mechanical resonators: intermodal coupling, internal resonance and synchronization”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 376.2127 (2018), p. 20170141.
- [5] Huiliang Cao et al. “Sensing mode coupling analysis for dual-mass MEMS gyroscope and bandwidth expansion within wide-temperature range”. In: *Mechanical Systems and Signal Processing* 98 (2018), pp. 448–464.

- [6] Chunbo Lan, Weiyang Qin, and Wangzheng Deng. “Energy harvesting by dynamic instability and internal resonance for piezoelectric beam”. In: *Applied Physics Letters* 107.9 (2015), p. 093902.
- [7] Francesco Cottone, Helios Vocca, and Luca Gammaitoni. “Nonlinear energy harvesting”. In: *Physical review letters* 102.8 (2009), p. 080601.
- [8] Samuel C Stanton, Clark C McGehee, and Brian P Mann. “Nonlinear dynamics for broadband energy harvesting: Investigation of a bistable piezoelectric inertial generator”. In: *Physica D: Nonlinear Phenomena* 239.10 (2010), pp. 640–653.
- [9] Raghuram Tattamangalam Raman, Revathy Padmanabhan, and Arvind Ajoy. “Stable Deflection in Ferroelectric Negative-Capacitance Hybrid MEMS Actuator With Cubic Nonlinear Spring”. In: *IEEE Transactions on Electron Devices* 69.9 (2022), pp. 5162–5169.
- [10] Shane A Migliore, Edgar A Brown, and Stephen P DeWeerth. “Novel nonlinear elastic actuators for passively controlling robotic joint compliance”. In: (2007).
- [11] Shay Chemny and Sefi Givli. “The mechanical behavior of fixed-angle bows”. In: *Acta Mechanica* 232.8 (2021), pp. 3215–3232.
- [12] Guangbo Hao. “A framework of designing compliant mechanisms with nonlinear stiffness characteristics”. In: *Microsystem Technologies* 24.4 (2018), pp. 1795–1802.
- [13] Sebastien Boisseau, Ghislain Despesse, and Bouhadjar Ahmed Seddik. “Nonlinear H-shaped springs to improve efficiency of vibration energy harvesters”. In: *Journal of Applied Mechanics* 80.6 (2013).
- [14] Christine Vehar Jutte and Sridhar Kota. “Design of nonlinear springs for prescribed load-displacement functions”. In: (2008).
- [15] Ben Rivlin and David Elata. “Design of nonlinear springs for attaining a linear response in gap-closing electrostatic actuators”. In: *International Journal of Solids and Structures* 49.26 (2012), pp. 3816–3822.
- [16] Xuhan Dai et al. “Tuning of nonlinear vibration via topology variation and its application in energy harvesting”. In: *Applied Physics Letters* 100.3 (2012), p. 031902.
- [17] Brian P Trease, Yong-Mo Moon, and Sridhar Kota. “Design of large-displacement compliant joints”. In: (2005).
- [18] Larry L Howell. “Compliant mechanisms”. In: *21st century kinematics*. Springer, 2013, pp. 189–216.

- [19] Larry L Howell, Spencer P Magleby, and Brian M Olsen. “Handbook of Compliant Mechanisms”. In: (2013), pp. 55–76.
- [20] Jianqiang Mou, Shixin Chen, and Yi Lu. “Single-crystal silicon MEMS microactuator for high-density hard disk drive”. In: *Device and Process Technologies for MEMS and Microelectronics II*. Vol. 4592. SPIE. 2001, pp. 443–449.
- [21] Lawrence F Shampine and Mark W Reichelt. “The matlab ode suite”. In: *SIAM journal on scientific computing* 18.1 (1997), pp. 1–22.

# Appendices

## A Prototyping

This chapter describes the prototyping of linear stages. Iteration on 3D printing and laser cutting was carried out, to find out the compensation value of CAD model for processing errors (resin deformation, laser beam diameter).

### A.1 3D printed prototype

Fabrication was performed using a Formlabs stereolithography (SLA) 3D printer. The material used is Tough 1500 resin, which has good elasticity and strength in the available material library of this project, and can be used to manufacture prototypes with isotropy mechanical properties [40].

Firstly, the design method of the 3D printed linear stage model is explored. The vibration frequency of the model should be small, otherwise it is difficult to observe its movement with a high-speed camera during the experimental stage. Therefore, thinner and longer flexures and heavier mass block should be used. The thickness of the flexure is the most significant parameter for the nature frequency. However, considering the precision and size limitations of the 3D printer, the thickness of the flexures should be at least 400  $\mu\text{m}$ . Similarly, the mass block size cannot be too large, because the largest print size of the available Formlabs printer is  $144 \times 144$  mm. Although the prototype of the complete model is not produced in this thesis, the model must still be designed on the premise that the complete model can be printed with this printer. Therefore, the length of the linear stage block is limited to 25 mm, instead of the maximum length of 144 mm that the printer can provide. This allows for a design space of 94 mm for the length of the nonlinear spring.

Several through holes are designed in the center of the mass block. These through holes can be used to reduce the weight of the mass block, or increase the weight by fixing on some metal screws.

A typical CAD model that is printed is shown in Figure 44(a). The width (out-of-plane thickness) of the mechanism is 6 mm, the thickness (in-plane thickness) of the flexure is 0.6 mm, the length of the flexure is 35 mm, the length of the block is 24 mm, the width of the block is 10 mm, the diameters of the holes are all 5 mm. There are two blocks, one of them can be used as the ground. When using the one with one hole as the mass block, the nature frequency

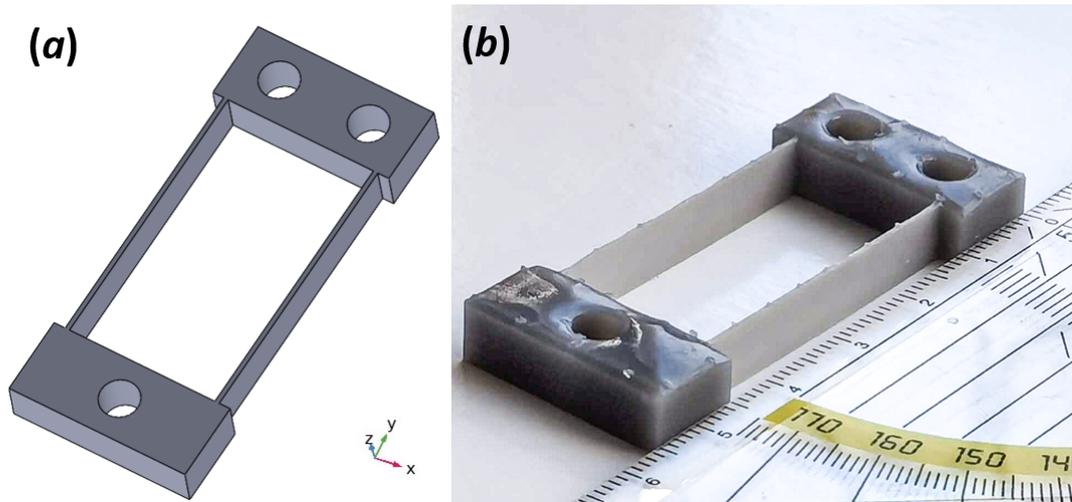


Figure 44: (a) CAD model of 3D printing prototype, (b) 3D printing result

would be 40.3 Hz. Vibration under this frequency can be safely recorded with a 1920 frames per second camera.

The 3D printing result is shown in Figure 44(b). Some other models with different sizes were also printed. Three main problems were found:

**Problem 1** When 3D printing a compliant mechanism, the bottom of the model should be integrated with some thin pillar supports, as shown in Figure 45. When the supporting pillars are being removed after the printing process, the fragile flexures may be damaged, and the connection spots will remain on the model, making the surface of the model not smooth. The spots can be seen in Figure 44(b). However, the remaining plastic spots are very small, and the impact on mass is almost negligible.

**Problem 2** For relatively slender and rigid parts (such as the ground part of a full model), they are often deformed and warped. This is especially noticeable on large-sized models. This might be related to the stress or deformation caused by shrinkage during resin solidification. The deformation may cause translation, vibration, and load to be non-coaxial, thereby affecting mechanism performance. However, the specific effects on performance have not yet been evaluated.

**Problem 3** There is a convex surface where the support is connected to the model. As Figure 44(b) shown, compared with the smooth side surfaces, the top surfaces of the mass blocks are convex. It is possible that the deformation is caused by gravity during the resin post-curing. For this classical prototype with 6 mm width in the  $z$  direction. The height of the convex

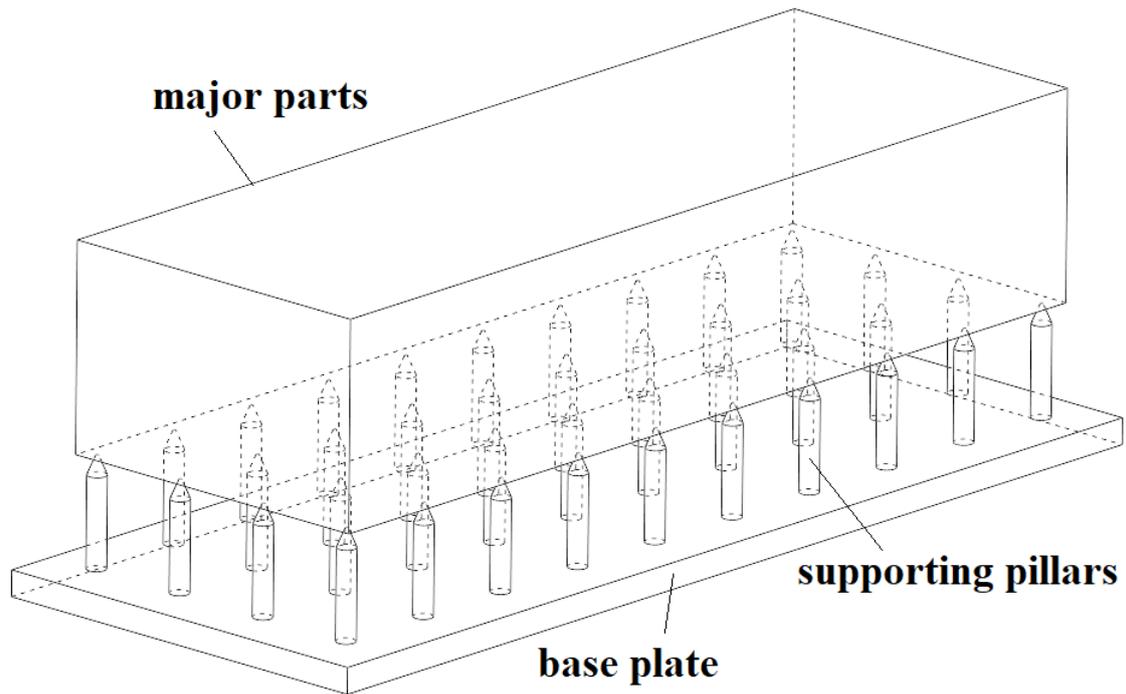


Figure 45: The supports of the mechanism

surface is about 0.9 mm. In solidworks, a relatively simple estimative modeling is performed, as shown in Figure 46. The volume of a perfect mass block with one hole is  $1,322.19 \text{ mm}^3$ . The volume of convex block is  $1,401.51 \text{ mm}^3$ . This convex surface brings a volume change of about 6.0%, which is an acceptable value. However, taking this error into consideration in design and analyze stage is still suggested.

## A.2 Laser cut prototype

Fabrication is performed with LASEA femtosecond laser cutter. The processing material is (100) silicon wafer with a diameter of 100mm and a thickness of 525  $\mu\text{m}$ . Since laser cutting can achieve higher precision, and monocrystalline silicon has higher stiffness. Models with smaller size compared with 3D printing can be achieved. This method is an ideal processing solution for the final product.

Similarly, the design method of the laser cut linear stage model is explored. Since the plan of the follow-up experiment plans is to use piezoelectric vibration stage and laser Doppler vibrometer to test the prototype and these devices have high measurement accuracy. Therefore, it is not necessary to make stricter restrictions on the natural frequency of mass blocks. Considering that miniaturization is an important aim of this thesis, this model should be as small as possible based on the laser cutter accuracy. And due to the size limitation of the vibrometer, the length

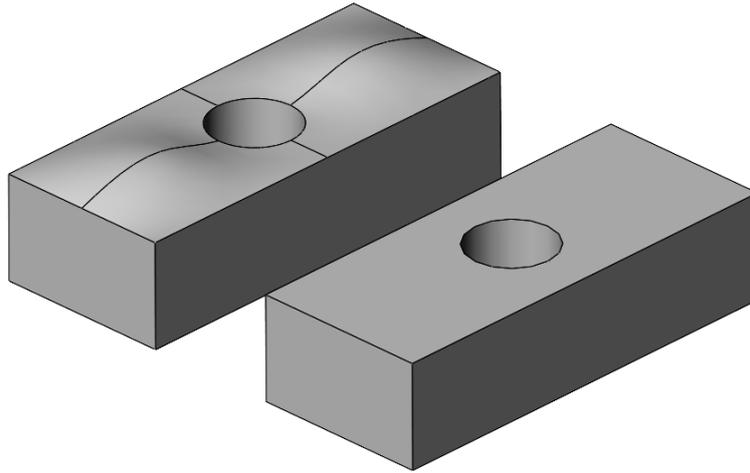


Figure 46: The mass blocks with and without convex surface

of the linear stage should be less than 15mm, and the width should be less than 5mm.

Since the smallest parts that can be processed by a laser cutting machine are related to the parameters of the laser beam (such as laser speed, repetitions, laser pulse rate, laser power), multiple rounds of prototype iterations are required to find different combinations of processing parameters that can achieve acceptable processing quality within an appropriate time.

Also, since flexures are very fragile. It may be necessary to design a protective border for the linear stage. And it is necessary to leave a path channel for the laser beam of the vibrometer in the subsequent experiment stage.

First, in order to test the effect of laser beam thickness on the model edge, a common linear stage was designed for processing test, as Figure 47(a) shown. The thickness (in-plane thickness) of the flexure is 400  $\mu\text{m}$ , the length of the flexure is 8 mm, the lengths ( $x$  direction) of the blocks are 6 mm, the widths ( $y$  direction) of the two blocks are 4 mm and 5mm.

The main laser cutting parameters are: laser speed, which is the moving speed of the laser focal point on the silicon wafer. When the scanning speed is small, local heat dissipation may not be timely, resulting in overheating and severe ablation. But if the movement is too fast, the light source may change direction too fast, making the polygon movement path of the laser source inaccurate, and a fillet may be formed, as Figure 48 shown. The maximum power of the Laser source is 50w. When the power is too high, it may cause overheating, but this can also reduce the processing time. The pulse rate that controls the laser pulse frequency have a similar effect as the power. Scanning the laser along the DXF outline path only once is not enough to cut through the silicon wafers, multiple repetitions are required. To check whether the silicon wafer

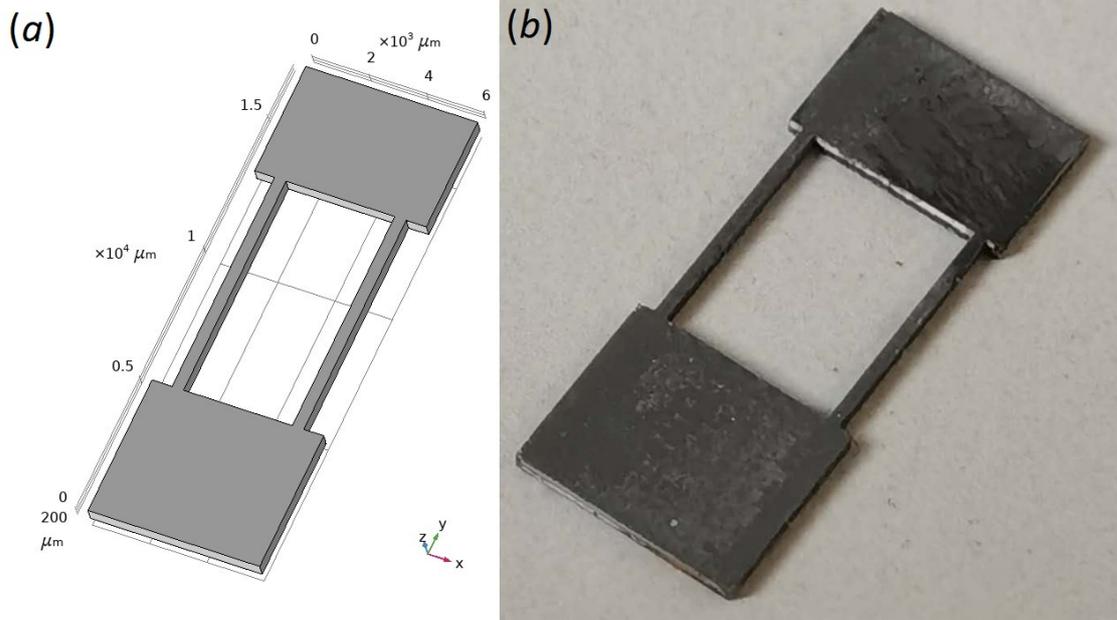


Figure 47: (a) CAD model of the testing laser cut prototype, (b) laser cut result

is cut through, the sample must be picked up from the working stage. After moving the wafer, it is very difficult to reposition the wafer and continue processing this sample. Therefore, the silicon wafer must be cut through in a single cut. Therefore, there should be more repetitions to ensure that the silicon is cut through, although this will increase the length of the process. In addition, since the thickness of the silicon wafer cannot be ignored, the laser focal point should not be on the upper surface of the silicon wafer, but under it. And it is better to cut several rounds. Each round has different focus point depth. Two cutting methods are available. One is to perform line cutting along the outline of the model, as shown in Figure 49(a). The other is to use a beam of light to hatch the excess part of the silicon wafer other than the model. The Hatching method is similar to using a pencil to paint on paper, as shown in Figure 49(b). The hatching method can ensure that the model is separated away, but hatching greatly increase the processing time.

After many tests, the optimized laser cutting parameters are: laser speed 400 mm/s, laser power 50% (15 W), pulse rate 37,000 Hz, 2,500 repetitions. Focusing on 200 μm, 300 μm, 400 μm below the wafer upper surface and executing once respectively. Under these parameters, the total processing time is about 25 minutes. The final product is shown in Figure 47(b). There is a little dust on the surface, but the overall shape is duplicated without obvious distortion. The cutting quality of the edge is shown in Figure 50. The compensation of the laser beam radius is 37 μm, and the fillets caused by high scanning speed have radius of 83 μm.

Then, the model size is compensated using the parameters obtained earlier, and the design is

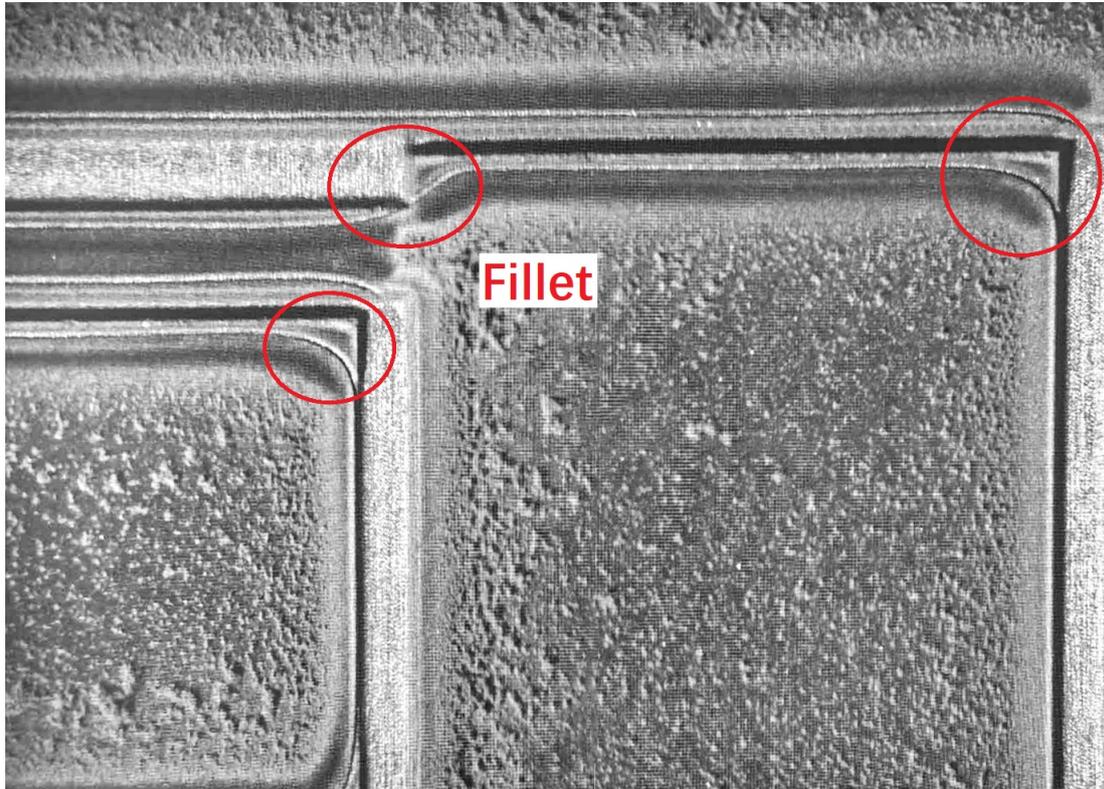


Figure 48: The fillet caused by high laser scanning speed

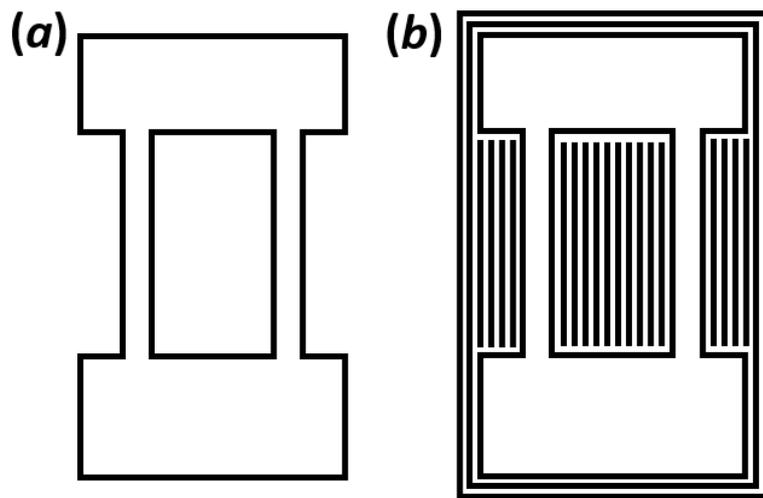


Figure 49: Laser scanning path. (a) cutting through the outline, (b) hatching

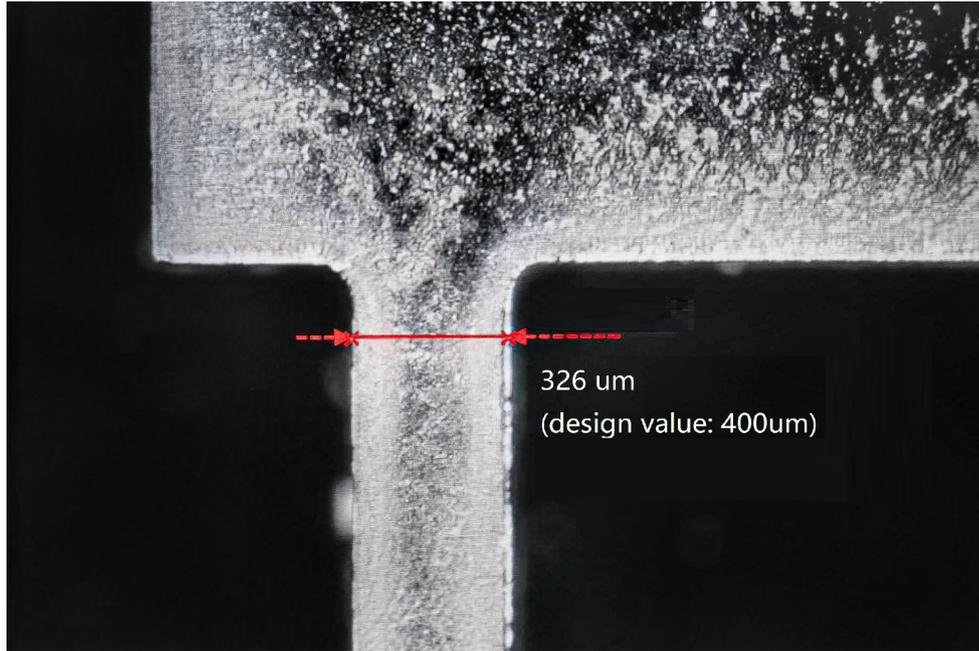


Figure 50: The edge quality under optimized laser cutting parameters

updated. Several models of smaller sizes were designed. they have a protective border  
 Then, the CAD models are updated based on the compensation parameters obtained earlier.  
 Several complex models with smaller sizes were designed. they have protective borders.

A typical linear stage model that can achieve good edge processing quality and relatively small size is shown in Figure 51. The laser beam radius compensation is taken into consideration in the CAD model. The thickness (in-plane thickness in  $x$  direction) of the flexure is 200  $\mu\text{m}$ , the length of the flexure is 4 mm, the The length ( $x$  direction) of the block is 1 mm, the width ( $y$  direction) of the block is 2 mm. The distance between protective border and the major parts is 333  $\mu\text{m}$ . The border thickness is 1 mm. The channel for the laser Doppler vibrometer near the mass block is 2 mm wide . There is a 4.5 mm long ground part at the bottom to fix the linear stage to the vibration stage. Theoretically, the natural frequency of this linear stage is 12,237 Hz.

It is worth noting that, according to the experience of the later vibration experiment, the side surface of the block used to reflect the laser vibrometer beam should be milled to increase the laser reflectivity. Various milling parameters were tested. The optimized laser cutting parameters are: laser speed 13 mm/s, laser power 5% (1.5 W), pulse rate 1,119 Hz, 200 repetitions. Focusing on the top surface of the wafer and executing once. Since the surface being milled is perpendicular to the plane of the wafer, wedges are used to hold the prototype.

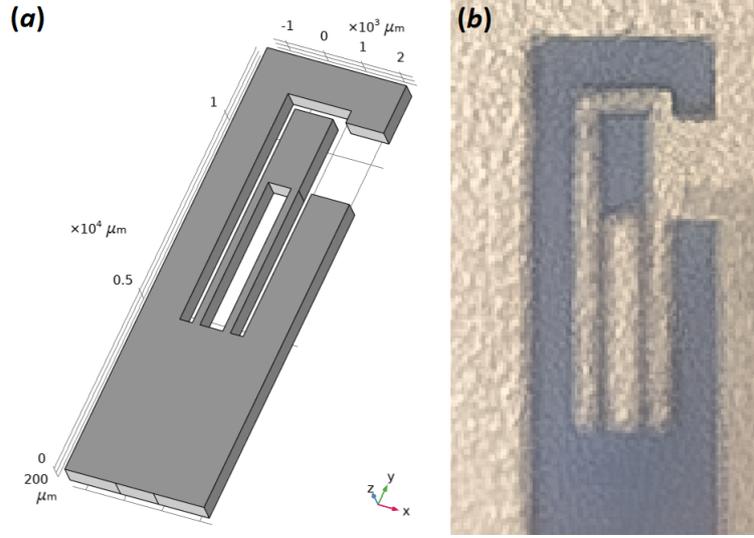


Figure 51: (a) CAD model of the laser cut linear stage, (b) laser cut result

## B Experimental validations

In this chapter, the performance evaluation of the linear stage prototypes from last chapter is carried out. For the larger 3D printed prototype, its natural frequency was tested. For the smaller laser cut prototype, its response to inputs of different frequencies were tested.

### B.1 3D printed prototype

In order to find out how well the natural frequency of the prototype matches the theoretical value, the typical 3D printed linear stage prototype in Figure 44 is tested. The mass block with two holes is fixed to the ground. The flexures are perpendicular to the ground. In this way, the theoretical nature frequency would be 40.3 Hz.

Input the initial displacement to the mass block with one hole, then release the block and record the translation of the mass block with a 1920 frames per second high-speed camera. The initial displacement in horizontal direction is 16.0 mm, which is 67% of block length and 46% of flexure length. Due to the flexure deformation, this input will lead to a vertical initial displacement of 4 mm.

The block displacement response is shown in Figure 52. The fitting function for the displacement is shown in Equation 33. Where the initial displacement  $A = 0.0163$  m. The damping ratio  $\zeta = 0.0429$ . Damping coefficient  $c = 0.0304$  kg/s. The quality factor  $Q = 11.66$ . The nature frequency without damping  $f_n = 40.77$  Hz. Its corresponding angular velocity  $\omega_n = 256.17$  rad/s. The real frequency of vibration in the experiment  $f_d = 40.73$  Hz. Its corresponding

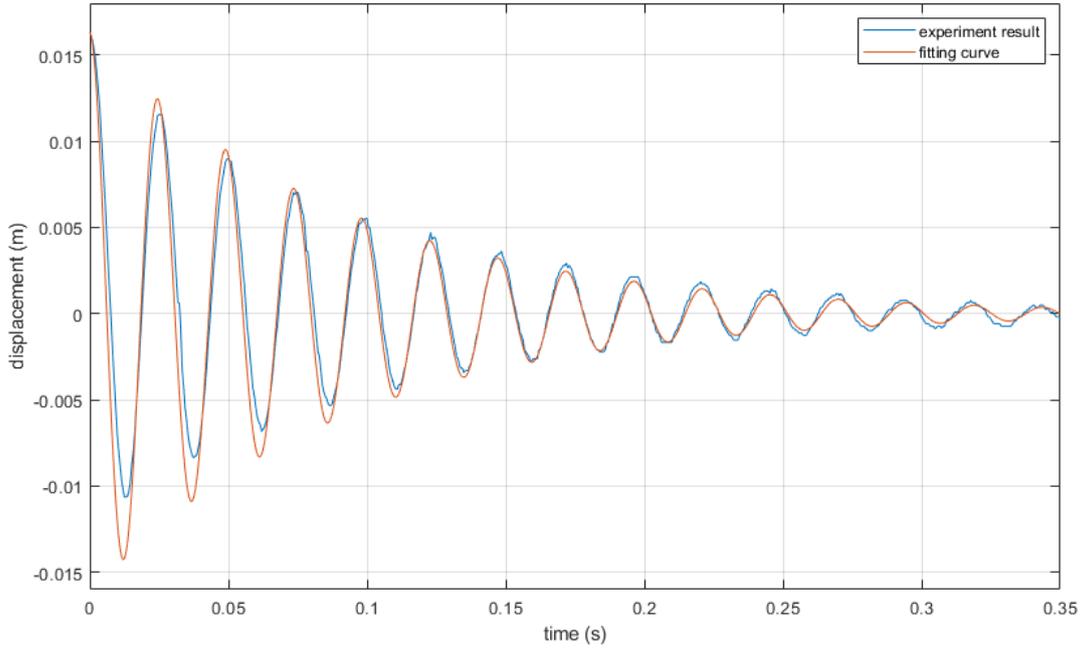


Figure 52: The displacement response of 3D printed prototype

angular velocity  $\omega_d$  satisfies Equation 34,  $\omega_d = 255.93$  rad/s.

$$x(t) = Ae^{-\zeta\omega_n t} \cdot \cos(\omega_d t) \quad (33)$$

$$\omega_d = \sqrt{1 - \zeta^2} \cdot \omega_n \quad (34)$$

The error between the theoretical value of the natural frequency and the experimental data is only 1.1567%. It can be seen that the linear stage manufactured by the stereolithography 3D printing method using tough 1500 resin can achieve very accurate natural frequency. However it is worth noting that the 3D printed models have non-negligible damping. The source of damping has not been investigated. Due to the limitation of equipment conditions, the vibration experiment was not carried out in vacuum. but according to a study by Jenna Gietl et al [41], the loss due to the fluid is not the dominant damping mechanism. Another potential source of damping is clamps. Prototypes have been clamped as much as possible to reduce this loss. Therefore, the main source of damping is most likely from within the resin material. Since the the natural frequency is closely related to design of the model, and the total size of the linear stage is very limited, it has not been studied whether the damping can be reduced by optimizing the model design.

One potential use of the whole system in this thesis is energy harvesting. Such high damping

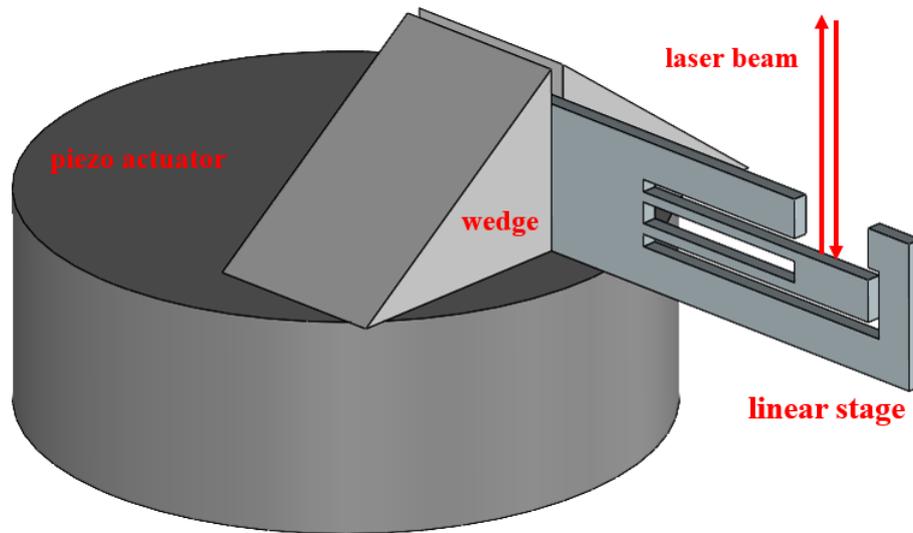


Figure 53: The fixing of the linear stage and the actuator

is not acceptable for the large amount of energy conversion involved in the process. Taking the typical model involved in the experiment as an example, after inputting a large initial displacement, the vibration only maintained for about 1 second, and then the vibration is difficult to observe with the naked eye. Therefore, whether the method of 3D printing is suitable for this thesis remains to be studied. But this method is very suitable if the linear stages do not require very low damping.

## B.2 Laser cut prototype

The typical laser cut linear stage prototype in Figure 51 is tested as well. For the CAD model, the theoretical nature frequency calculated with COMSOL build-in eigenfrequency analyzer is 12237 Hz.

Experiment system components include: Polytec OFV 512 fiber interferometer, MSA-400 junction box, OFV-5000 vibrometer controller, and Keithley 2400 sourcemeter. Since the incident direction of the laser beam is vertical to the ground, the prototype is fixed to the piezo actuator by being glued with two wedges, as Figure 53 shown. A frequency sweep until 20 kHz was carried out.

The response of the linear stage after fast Fourier transform is shown in Figure 54. Multiple peaks can be seen in the plot. The one at 11067 Hz has the highest magnitude of 50.0  $\mu\text{m/s}$ . The error between this frequency and the COMSOL simulation result of the origin model is 12.8%.

Since the actual prototype was not measured to get the real dimension data, the CAD model of the prototype was not rebuilt. Therefore, the theoretical first order eigenfrequency of the prototype is not available.

Considering that the size of the sample is still relatively large for the experimental devices, the excitation force that the actuator can provide is not great enough, and the sample had to be exposed to the air. Therefore, the signal-to-noise ratio is low, so that the plot is still far from the perfect curve. In addition, since the constraint method of this experiment is gluing with wedges. Large errors could be caused.

The prototypes and experimental procedures were not further optimized due to time constraints. But according to the results obtained so far, laser cutting method can obtain great processing quality for the prototype and can be used to manufacture linear stage with accurate nature frequency.

### B.3 Prospects

For the 3D printed prototypes, research on how to reduce the impact of damping by optimizing the geometric size and topology of the model can be carried out. For example, changing the thickness and length of the flexures while keeping the natural frequency unchanged to study the damping. It is also possible to study the effect of different materials on damping and, if possible, to test prototypes obtained using metal additive manufacturing methods.

For the laser-cut models, the processing limit of existing equipment for models remains to be explored. Some smaller models may be possible. And the experimental process of this model still needs to be optimized, especially the clamping method.

## C Potential nonlinear spring designs

In this chapter, some other potential designs for the nonlinear spring are proposed.

A concept utilizing force decomposition and Taylor expansion is shown in the Figure 55. The two mass blocks are connected together by two ordinary tension coil linear springs with a stiffness of  $K_0$  and an equilibrium length of  $L$ . The connection is ordinary revolute joint. Mass block  $m_1$  is fixed to the ground. When applying a load  $F$  on  $m_2$  in  $x$  direction, the displacement of  $m_2$  would be  $\Delta x$ . The relation between  $F$  and  $\Delta x$  satisfies Equation 35. Applying Taylor expansion at  $\Delta x = 0$ , the load becomes Equation 36. If  $\Delta x$  is relatively small ( $-0.3L < \Delta x < 0.3L$ ), Equation 36 becomes Equation 37, which is a perfect cubic relation.

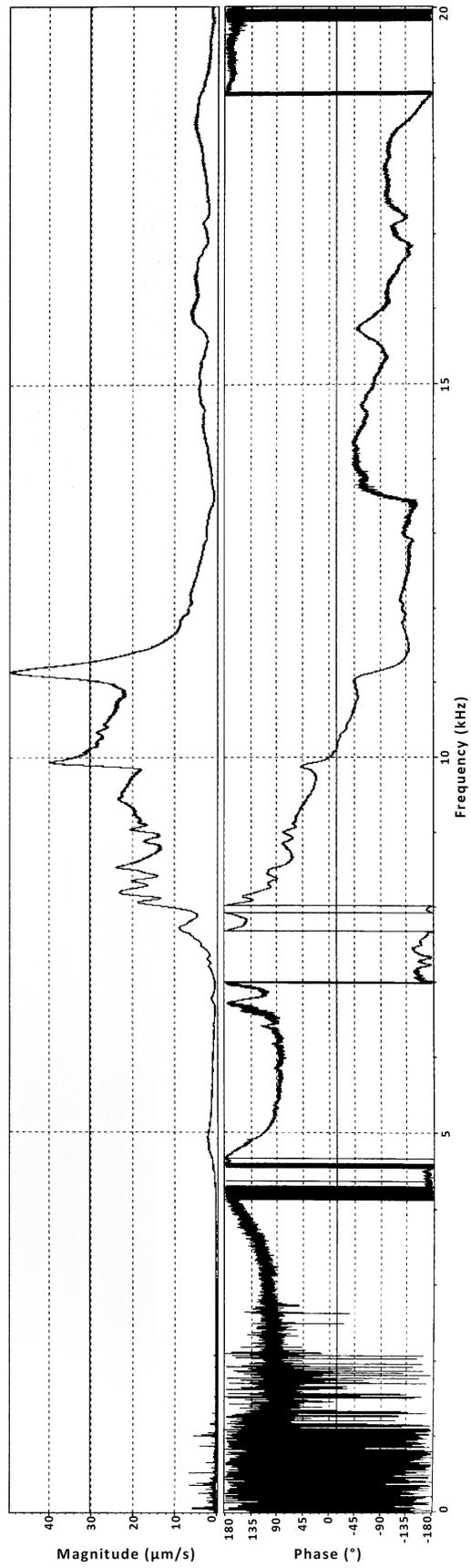


Figure 54: The frequency response of laser cut prototype after FFP

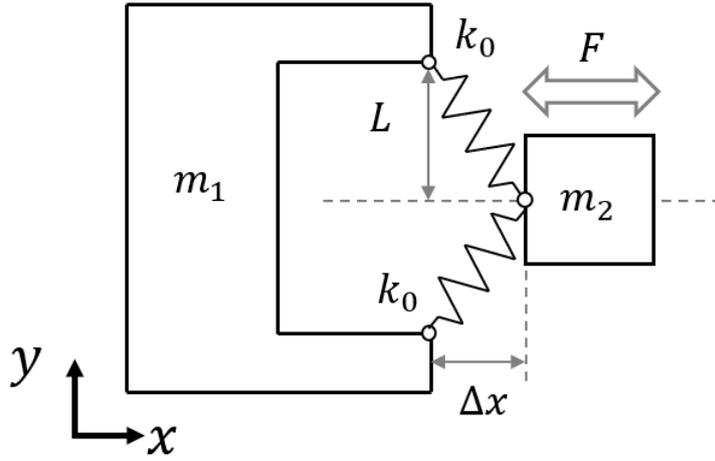


Figure 55: The principle of a possible concept

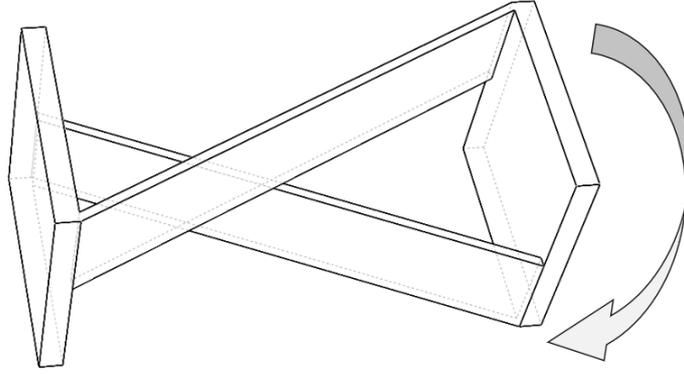


Figure 56: Cross-axis flexural pivot

This concept can achieve great stroke and perfect nonlinearity. However, the requirement of revolute joints makes it very difficult to convert it to a compliant mechanism design. Using cross-axis flexural pivots (Figure 56) to replace revolute joint is method to integrate mass blocks and linear springs. But the overhanging components of the flexural pivots in  $z$  direction will make this integrated model impossible to be laser cut.

$$F = 2k_0x - \frac{2k_0L\Delta x}{\sqrt{\Delta x^2 + L^2}} \quad (35)$$

$$F = \frac{k_0\Delta x^3}{L^2} - \frac{3k_0\Delta x^5}{4L^4} + \frac{5k_0\Delta x^7}{8L^6} + o(\Delta x^9) \quad (36)$$

$$F \approx \frac{k_3\Delta x^3}{L^2} \quad (-0.3L < \Delta x < 0.3L) \quad (37)$$

In addition, the continuum topology optimization commonly used in the architecture field [42] can also be used to optimize the geometric layout of compliant mechanism structure. These structures usually have very complex geometric layout with many elements. It may take a large number of iterations to generate a new structure with a cubic load-displacement curve. However, optimizing based on an existing spline-shaped or crank slider design might save a lot of time. Combining the two optimization strategies together is also very promising.

## D Nonlinear spring optimization scripts

This chapter describes the work flow of the nonlinear spring optimization and the corresponding codes.

The workflow for this project involved several steps. Initially, a Comsol Multiphysics (.mph) file was created and used to generate a standard model. This model was then run and debugged within the Comsol software to calculate the displacement-load curve data of the spring. An alternative method involved saving the model as a separate .m file, which can be opened in Matlab for code verification and modification.

Next, the LiveLink for Matlab was used to load the .mph file and modify the parameters of the model in the Matlab code. This allowed for adjustments to the magnitude of the applied force, size parameters of the model, and other variables without launching the Comsol software's graphical user interface. The resulting displacement-load data of the model was outputted and saved as a .txt file.

Afterwards, a genetic algorithm was employed to optimize certain parameters, such as the length of a specific part. The displacement-load data of the model under these parameters was calculated and compared to a perfect cubic curve. The optimization process was repeated for several generations until an acceptable level of nonlinearity was achieved.

The parameter settings of the genetic algorithm itself, the parameter settings of the finite element model, and the result analysis are all integrated into one file.

The whole script is roughly divided into 5 parts. In Part 1, initializing the system, inputting the initial value and range of optimization variables and importing the project file. In the subsequent Part 2, the first-generation individuals are randomly generated and their fitness is calculated. In Part 3, chromosome exchange and mutation were performed on the first-generation individuals. In Part 4, executing loops, calculating the fitness of each generation, and doing the exchange and mutation according to the results. In Part 5, outputting the final

optimization result and saving it as some .txt files. The design variables and displacement load curves of each individual generation are stored separately and can be used for subsequent manual analysis.

The script is as follows:

```
1 %% Part 1 %%
2     %mph_modify
3     close all
4     %clc
5     cd D:\Files\fin\importantPrograms\SpringIterate
6     model=mphopen('basic_comsol_file.mph')%import the model
7 tic
8
9     %% comsol basic parameters
10    %
11    %load range, it is best not to change the following
12    model.param.set('force', '4[N]');
13    model.study('std1').feature('stat').set('plistarr', {'range
14        (0,force/20,force)'});
15
16    model.study('std2').feature('stat').set('plistarr', {'range
17        (0,-force/20,-force)'});
18
19    %%Material properties, it is best not to change the following
20    %model.component('comp1').coordSystem('sys2').set('base',
21        {'1/(6^0.5)' '-1/(6^0.5)' '0'; '1/(6^0.5)' '1/(6^0.5)'
22        '0'; '0' '0' '(2^0.5)/(6^0.5)'});
23
24    %Grid, 5 is normal, the smaller the more detailed. Level 1
25    takes about 54s, level 4 takes about 12s.
26    model.component('comp1').mesh('mesh1').autoMeshSize(3);
27
28    %% GA parameters
29    setnum=50; %population size 5,10,15,20,25,etc
30    gennum=8;
```

```

25     pc = 0.8;% crossover rate
26     pm = 0.11;% mutation rate
27
28     L=15;%lenth of DNA ,also the num of parameters
29     exSetNum=(floor(pc* setnum/2))*2; %number of sets to be
        exchanged. even number
30
31     quality_table=zeros(gennum ,setnum);
32     quality_gen_mean=zeros(gennum ,1);
33     %% %change parameters2
34
35     % % % %model.param.set('t5', '0.4[mm]', 'min distance');
36
37 %     model.param.set('t1', '130[um]'); 100-200
38 %     model.param.set('t2', '130[um]'); 100-200
39 %     model.param.set('t3', '130[um]'); 100-200
40 %     model.param.set('L1', '900[um]'); 800-1200
41 %     model.param.set('L2', '900[um]'); 800-1200
42 %     model.param.set('L3', '1000[um]'); 800-1200
43 %     model.param.set('LA', '3000[um]'); 2000-7000
44 %     model.param.set('LB', '5000[um]'); 2000-7000
45 %     model.param.set('theta1', 'pi/4[rad]'); 10*pi/180 - 70*pi
        /180
46 %     model.param.set('theta3', 'pi/6[rad]'); 10*pi/180 - 70*pi
        /180
47 %     model.param.set('t0', '500[um]'); 300-700
48 %     model.param.set('L0', '500[um]'); 150-400
49 %     model.param.set('theta4', 'pi/4[rad]'); -45*pi/180 - 45*pi
        /180
50 %     model.param.set('theta5', 'pi/6[rad]'); -45*pi/180 - 45*pi
        /180
51 %     model.param.set('theta6', 'pi/6[rad]'); -45*pi/180 - 45*pi
        /180
52

```

```

53
54 %% Part 2 %%
55
56 %% generation 1
57 %% case 1 random
58     gen1cp=zeros(setnum,12);
59     gen1cp(:,01)=100+100*rand(setnum,1);
60     gen1cp(:,02)=100+100*rand(setnum,1);
61     gen1cp(:,03)=100+100*rand(setnum,1);
62     gen1cp(:,04)=800+400*rand(setnum,1);
63     gen1cp(:,05)=800+400*rand(setnum,1);
64     gen1cp(:,06)=800+400*rand(setnum,1);
65     gen1cp(:,07)=2000+5000*rand(setnum,1);
66     gen1cp(:,08)=2000+5000*rand(setnum,1);
67     gen1cp(:,09)=10*pi/180+60*pi/180*rand(setnum,1);
68     gen1cp(:,10)=10*pi/180+60*pi/180*rand(setnum,1);
69     gen1cp(:,11)=300+400*rand(setnum,1);
70     gen1cp(:,12)=150+250*rand(setnum,1);
71     gen1cp(:,13)=-45*pi/180+90*pi/180*rand(setnum,1);
72     gen1cp(:,14)=-45*pi/180+90*pi/180*rand(setnum,1);
73     gen1cp(:,15)=-45*pi/180+90*pi/180*rand(setnum,1);
74
75 %% case 1 random
76 %% case 2 load old stuff,
77 %     % save('gen0LDcp.txt','gen1cp','-ascii');
78 %     load gen1cp.txt
79 %     save('full_parameters_gen1.txt','gen1cp','-ascii');
80 %% case 2 load old stuff
81
82 %% edit generation 1 control points and do calculation
83 %% edit generation 1 control points and do calculation
84
85 for i=1:setnum
86     %% save parameters

```

```

87     the15para=gen1cp(i,:);
88     filename1=['paragen1set',num2str(i),'.txt'];
89     save(filename1,'the15para','-ascii');
90     %% do the settings for comsol
91         bridge=[num2str(the15para(1)),'[um]'];
92         model.param.set('t1', bridge);
93
94         bridge=[num2str(the15para(2)),'[um]'];
95         model.param.set('t2', bridge);
96
97         bridge=[num2str(the15para(3)),'[um]'];
98         model.param.set('t3', bridge);
99
100        bridge=[num2str(the15para(4)),'[um]'];
101        model.param.set('L1', bridge);
102
103        bridge=[num2str(the15para(5)),'[um]'];
104        model.param.set('L2', bridge);
105
106        bridge=[num2str(the15para(6)),'[um]'];
107        model.param.set('L3', bridge);
108
109        bridge=[num2str(the15para(7)),'[um]'];
110        model.param.set('LA', bridge);
111
112        bridge=[num2str(the15para(8)),'[um]'];
113        model.param.set('LB', bridge);
114
115        bridge=[num2str(the15para(9)),'[rad]'];
116        model.param.set('theta1',bridge);
117
118        bridge=[num2str(the15para(10)),'[rad]'];
119        model.param.set('theta3', bridge);
120

```

```

121     bridge=[num2str(the15para(11)), '[um]'];
122     model.param.set('t0', bridge);
123
124     bridge=[num2str(the15para(12)), '[um]'];
125     model.param.set('L0', bridge);
126
127     bridge=[num2str(the15para(13)), '[rad]'];
128     model.param.set('theta4', bridge);
129
130     bridge=[num2str(the15para(14)), '[rad]'];
131     model.param.set('theta5', bridge);
132
133     bridge=[num2str(the15para(15)), '[rad]'];
134     model.param.set('theta6', bridge);
135     %% for this design, do a study in positive load
136     model.study('std1').run;% run study once
137
138     %% get the results
139     str1 = mphtable(model, 'tbl1'); % extract probe table
140     tbl_1 = str1.data; % extract probe table[U+FFFD]column 1
141         is load, column 2 is displacement
142     rawdataload=tbl_1(:,1);
143     rawdatadisp=tbl_1(:,2);
144
145     str2 = mphtable(model, 'tbl2'); % extract probe table
146     tbl_2 = str2.data; % extract probe table[U+FFFD]column 1
147         is load, column 2 is von mises
148     rawdatastress=tbl_2(:,2);
149
150     tbl_3 = [rawdataload,rawdatadisp,rawdatastress];
151
152     %% for this design, do a study in negative load
153     model.study('std2').run;% run study once
154     str4 = mphtable(model, 'tbl1'); % extract probe table

```

```

153     tbl_4 = str4.data; % extract probe table[U+FFFD]column 1
        is load, column 2 is displacement
154     rawdataload=tbl_4(:,1);
155     rawdatadisp=tbl_4(:,2);
156     str5 = mphtable(model,'tbl2'); % extract probe table
157     tbl_5 = str2.data; % extract probe table[U+FFFD]column 1
        is load, column 2 is von mises
158     rawdatastress=tbl_5(:,2);
159     tbl_6 = [rawdataload,rawdatadisp,rawdatastress];
160     tbl_7=flip(tbl_6);
161     tbl_7(end,:)=[];
162     tbl_8=[tbl_7;tbl_3];
163     rawdataload=tbl_8(:,1);
164     rawdatadisp=tbl_8(:,2);
165     rawdatastress=tbl_8(:,3);
166     filename3=['rawdatagen1set',num2str(i),'.txt'];
167
168     save(filename3,'tbl_8','-ascii');
169
170     %% judging the quality of the design
171
172     maxload=tbl_8(end,1);%N y
173     maxdisp=tbl_8(end,2);%mm x
174     mindisp=tbl_8(1,2);%mm x, is negative
175
176     standard_alpha= 0.5*(maxload/(maxdisp^3)-maxload/(
        mindisp^3));
177     errorstack=0;
178     for ijk=1:length(tbl_8)
179         %errorstack=errorstack+abs(yload(ijk)-standard_alpha*
            xdisp(ijk)^3);
180         standard_load(ijk)=standard_alpha*rawdatadisp(ijk)^3;
181         errorsingle(ijk)=rawdataload(ijk)-standard_load(ijk);
182         errorstack=errorstack+abs(errorsingle(ijk));

```

```

183     end
184     qlt=(maxload*length(tbl_8))/errorstack;
185     qlt=qlt^6; %The bigger the better, the worst pure
           straight line is about 5937
186     quality_table(1,i)=qlt;
187     maxVon_table(1,i)=tbl_8(end,2);
188 end
189
190
191 %% survived guys of gen1:
192 quality_temp=quality_table(1,:);
193
194 [Rankedquality1,RankSeedquality1]=sort(quality_temp) ; %front
           : value small, bad
195 surviveRate=quality_temp/sum(quality_temp); %looks like
           0.0007    0.0072    0.0099    0.4278    0.5544
196
197 %survival rate of this generation
198 %looks like 0.0007    0.0072    0.0099    0.4278    0.5544
199
200 for i=1:setnum
201     %surviveRate(i)
202     ratebreak(i)=sum(surviveRate(1:i));%looks like
           0.3,0.4,0.7,0.9,1.0
203 end
204
205 for ii=1:setnum
206     sjs = rand;
207     for i=1:setnum
208         if sjs <= ratebreak(i) %If the survival rate of group
           i is greater than the random number
209             survivedsets(ii)=i;
210             break
211         end

```

```

212         end
213     end
214     %survivedsets;% looks like [1,4,1,3,1] Here are the original
        survivors, not the ascending ones
215
216     if std(survivedsets)==0
217         %RankSeedquality1(setnum-1) is a good but not the best
            choice
218         survivedsets(2)=RankSeedquality1(setnum-1);
219     end
220     %survived first generation control points
221
222     gen1cp=gen1cp(survivedsets,:);%the design parameters of the
        new generation
223
224     %% above: first generation
225     %% below: TRANSFORM FIRST generation TO NEW generation
226
227
228 %% Part 3 %%
229
230 %% EXCHANGE
231 %1and3, 2and 4 exchange
232 for i = 1:exSetNum/2                                %, 1,2
233
234         breakpointA=randi(L-2); %1~10, if bpA=7
235         breakpointB=breakpointA+randi(L-1-breakpointA);%
            randi=1or2or3or4, if randi=2,bpB=9
236         tempCPA= gen1cp(i,:); %set1. if i=2, set2
237         tempCPB= gen1cp(i+exSetNum/2,:);%set3. if i=2,
            set4
238
239         tempCPAbackup=tempCPA;
240         tempCPBbackup=tempCPB;

```

```

241     gen1cp(i,:)=[tempCPB(1:breakpointA), tempCPA(breakpointA+1:
           breakpointB) ,tempCPB(breakpointB+1:end)];%1,2.
242 %1~7,8~9,10~end
243     gen1cp(i+exSetNum/2,:)=[tempCPA(1:breakpointA), tempCPB(
           breakpointA+1:breakpointB) ,tempCPA(breakpointB+1:end)];%
           3,4
244 %-----AVOID INTERSECTION is not needed
245
246     end
247 % gen1cp NOW HAS FINISHED EXCHANGE (no intersection)
248
249 %% MUTATION
250     for i = 1: setnum
251         if rand < pm
252             randSeed=randi(L);
253             backupValue=gen1cp(i, randSeed);
254 %             gen1cp(i, randSeed) = backupValue -5 + (5+5)*
           rand;
255 % %
256             switch randSeed
257                 case 1
258                     gen1cp(i, randSeed)=100+100*rand(1);
259                 case 2
260                     gen1cp(i, randSeed)=100+100*rand(1);
261                 case 3
262                     gen1cp(i, randSeed)=100+100*rand(1);
263                 case 4
264                     gen1cp(i, randSeed)=800+400*rand(1);
265                 case 5
266                     gen1cp(i, randSeed)=800+400*rand(1);
267                 case 6
268                     gen1cp(i, randSeed)=800+400*rand(1);
269                 case 7
270                     gen1cp(i, randSeed)=2000+5000*rand(1);

```

```

271         case 8
272         gen1cp(i, randSeed)=2000+5000*rand(1);
273         case 9
274         gen1cp(i, randSeed)=10*pi/180+60*pi/180*rand
           (1);
275         case 10
276         gen1cp(i, randSeed)=10*pi/180+60*pi/180*rand
           (1);
277         case 11
278         gen1cp(i, randSeed)=300+400*rand(1);
279         case 12
280         gen1cp(i, randSeed)=150+250*rand(1);
281         case 13
282         gen1cp(i, randSeed)=-45*pi/180+90*pi/180*rand
           (1);
283         case 14
284         gen1cp(i, randSeed)=-45*pi/180+90*pi/180*rand
           (1);
285         case 15
286         gen1cp(i, randSeed)=-45*pi/180+90*pi/180*rand
           (1);
287         end
288     end
289 end
290
291     % gen1cp NOW HAS FINISHED MUTATION (no intersection)
292 %toc
293 quality_gen_mean(1)=sum(quality_table(1,:))/setnum;
294 save('full_quality_gen_mean.txt','quality_gen_mean','-ascii');
295 save('full_quality_gen_set.txt','quality_table','-ascii');
296 save('full_maxVon_gen_set.txt','maxVon_table','-ascii');
297
298
299 %% Part 4 %%

```

```

300 %% Generation loops start here
301
302 genOLDcp=gen1cp;
303
304 for j=2:gennum
305     %% settings and calculation for this generation---
306     filename4=['full_parameters_gen',num2str(j),'.txt'];
307     save(filename4,'genOLDcp','-ascii');
308
309     for i=1:setnum
310         the15para=genOLDcp(i,:);
311         filename1=['paragen',num2str(j),'set',num2str(i),'.txt'];
312         save(filename1,'the15para','-ascii');
313
314         %% do the settings for comsol
315         bridge=[num2str(the15para(1)),'[um]'];
316         model.param.set('t1', bridge);
317
318         bridge=[num2str(the15para(2)),'[um]'];
319         model.param.set('t2', bridge);
320
321         bridge=[num2str(the15para(3)),'[um]'];
322         model.param.set('t3', bridge);
323
324         bridge=[num2str(the15para(4)),'[um]'];
325         model.param.set('L1', bridge);
326
327         bridge=[num2str(the15para(5)),'[um]'];
328         model.param.set('L2', bridge);
329
330         bridge=[num2str(the15para(6)),'[um]'];
331         model.param.set('L3', bridge);
332
333         bridge=[num2str(the15para(7)),'[um]'];

```

```

334     model.param.set('LA', bridge);
335
336     bridge=[num2str(the15para(8)), '[um]'];
337     model.param.set('LB', bridge);
338
339     bridge=[num2str(the15para(9)), '[rad]'];
340     model.param.set('theta1', bridge);
341
342     bridge=[num2str(the15para(10)), '[rad]'];
343     model.param.set('theta3', bridge);
344
345     bridge=[num2str(the15para(11)), '[um]'];
346     model.param.set('t0', bridge);
347
348     bridge=[num2str(the15para(12)), '[um]'];
349     model.param.set('L0', bridge);
350
351     bridge=[num2str(the15para(13)), '[rad]'];
352     model.param.set('theta4', bridge);
353
354     bridge=[num2str(the15para(14)), '[rad]'];
355     model.param.set('theta5', bridge);
356
357     bridge=[num2str(the15para(15)), '[rad]'];
358     model.param.set('theta6', bridge);
359     %% for this design, do a study in positive load
360     model.study('std1').run([U+FFFD][U+FFFD][U+FFFD]study
361     %% get the results
362     str1 = mphtable(model, 'tbl1');
363     tbl_1 = str1.data;
364     rawdataload=tbl_1(:,1);
365     rawdatadisp=tbl_1(:,2);
366
367     str2 = mphtable(model, 'tbl2');

```

```

368     tbl_2 = str2.data;
369     rawdatastress=tbl_2(:,2);
370     tbl_3 = [rawdataload,rawdatadisp,rawdatastress];
371     %% for this design, do a study in negative load
372     model.study('std2').run;%
373     str4 = mphtable(model,'tbl1');
374     tbl_4 = str4.data; %
375     rawdataload=tbl_4(:,1);
376     rawdatadisp=tbl_4(:,2);
377
378     str5 = mphtable(model,'tbl2'); %
379     tbl_5 = str2.data; %
380     rawdatastress=tbl_5(:,2);
381
382     tbl_6 = [rawdataload,rawdatadisp,rawdatastress];
383     tbl_7=flip(tbl_6);
384     tbl_7(end,:)=[];
385     tbl_8=[tbl_7;tbl_3];
386     rawdataload=tbl_8(:,1);
387     rawdatadisp=tbl_8(:,2);
388     rawdatastress=tbl_8(:,3);
389     filename3=['rawdatagen1set',num2str(i),'.txt'];
390
391     save(filename3,'tbl_8','-ascii');
392
393     %% judging the quality of the design
394
395     maxload=tbl_8(end,1);%N y
396     maxdisp=tbl_8(end,2);%mm x
397     mindisp=tbl_8(1,2);%mm x, is negative
398
399     standard_alpha= 0.5*(maxload/(maxdisp^3)-maxload/(
400         mindisp^3));
        errorstack=0;

```

```

401     for ijk=1:length(tbl_8)
402         %errorstack=errorstack+abs(yload(ijk)-standard_alpha*
           xdisp(ijk)^3);
403         standard_load(ijk)=standard_alpha*rawdatadisp(ijk)^3;
404         errorsingle(ijk)=rawdataload(ijk)-standard_load(ijk);
405         errorstack=errorstack+abs(errorsingle(ijk));
406     end
407     qlt=(maxload*length(tbl_8))/errorstack;
408     qlt=qlt^6;%The bigger the better, the worst pure straight
           line is about 5937
409     quality_table(1,i)=qlt;
410     maxVon_table(1,i)=tbl_8(end,2);
411     quality_table(j,i)=qlt;
412 end
413 %% settings and calculation for this generation---
414
415 %% survived guys of gen j:
416 quality_temp=quality_table(j,:);
417 [Rankedquality1,RankSeedquality1]=sort(quality_temp) ; %front
           : value small, bad
418 surviveRate=quality_temp/sum(quality_temp); %looks like
           0.0007    0.0072    0.0099    0.4278    0.5544
419
420 %%survival rate of this generation
421 %looks like 0.0007    0.0072    0.0099    0.4278    0.5544
422
423 for i=1:setnum
424     %surviveRate(i)
425     ratebreak(i)=sum(surviveRate(1:i));%looks like
           0.3,0.4,0.7,0.9,1.0
426 end
427
428 for ii=1:setnum
429     sjs = rand;

```

```

430     for i=1:setnum
431         if sjs <= ratebreak(i) %If the survival rate of group
           i is greater than the random number
432             survivedsets(ii)=i;
433             break
434         end
435     end
436 end

437
438 if std(survivedsets)==0
439     %RankSeedquality1(setnum-1) is a good but not the best
           choice
440     survivedsets(2)=RankSeedquality1(setnum-1);
441 end

442
443 genOLDcp=genOLDcp(survivedsets ,:);%the design parameters of
           the new generation
444
445 %% repetation of Part 3 %%
446 %% EXCHANGE
447 %1and3, 2and 4 exchange
448 for i = 1:exSetNum/2           %, 1,2
449
450         breakpointA=randi(L-2); %1~10, if bpA=7
451         breakpointB=breakpointA+randi(L-1-breakpointA);%
           randi=1or2or3or4, if randi=2,bpB=9
452         tempCPA= genOLDcp(i,:); %set1. if i=2, set2
453         tempCPB= genOLDcp(i+exSetNum/2,:);%set3. if i=2,
           set4
454
455         tempCPAbackup=tempCPA;
456         tempCPBbackup=tempCPB;
457 genOLDcp(i,:)=[tempCPB(1:breakpointA), tempCPA(breakpointA+1:
           breakpointB) ,tempCPB(breakpointB+1:end)];%1,2.

```

```

458 %1~7,8~9,10~end
459 genOLDcp(i+exSetNum/2,:)= [tempCPA(1:breakpointA), tempCPB(
    breakpointA+1:breakpointB) ,tempCPA(breakpointB+1:end)];%
    3,4
460 %-----AVOID INTERSECTION is not needed
461 end
462 % gen1cp NOW HAS FINISHED EXCHANGE (no intersection)
463
464 %% MUTATION
465     for i = 1: setnum
466         if rand < pm
467             randSeed=randi(L);
468             backupValue=genOLDcp(i, randSeed);
469 %             gen1cp(i, randSeed) = backupValue -5 + (5+5)*
    rand;
470 % %
471         switch randSeed
472             case 1
473                 genOLDcp(i, randSeed)=120+80*rand(1);
474             case 2
475                 genOLDcp(i, randSeed)=120+80*rand(1);
476             case 3
477                 genOLDcp(i, randSeed)=120+80*rand(1);
478             case 4
479                 genOLDcp(i, randSeed)=800+400*rand(1);
480             case 5
481                 genOLDcp(i, randSeed)=800+400*rand(1);
482             case 6
483                 genOLDcp(i, randSeed)=800+400*rand(1);
484             case 7
485                 genOLDcp(i, randSeed)=2500+5500*rand(1);
486             case 8
487                 genOLDcp(i, randSeed)=2500+5500*rand(1);
488             case 9

```

```

489         genOLDcp(i, randSeed)=10*pi/180+70*pi/180*
           rand(1);
490     case 10
491         genOLDcp(i, randSeed)=10*pi/180+70*pi/180*
           rand(1);
492     case 11
493         genOLDcp(i, randSeed)=300+400*rand(1);
494     case 12
495         genOLDcp(i, randSeed)=150+250*rand(1);
496     end
497 end
498 end
499 quality_gen_mean(j)=sum(quality_table(j,:))/setnum;
500
501 save('full_quality_gen_mean.txt','quality_gen_mean','-ascii');
502     % genOLDcp (genJcp) NOW HAS FINISHED MUTATION (no
           intersection)
503 %toc
504 save('full_quality_gen_set.txt','quality_table','-ascii');
505 save('full_maxVon_gen_set.txt','maxVon_table','-ascii');
506
507 end
508 %% Loops END here
509
510 %% Part 5 %%
511
512 toc
513 save('full_quality_gen_mean.txt','quality_gen_mean','-ascii');
514 save('full_quality_gen_set.txt','quality_table','-ascii');
515 save('full_maxVon_gen_set.txt','maxVon_table','-ascii');
516
517 best_q=max(max(quality_table));
518 mean_error=best_q^(-1/6)
519 oldqlt=1/mean_error

```

```
520 [best_gen, best_set]=find(quality_table==best_q)
```

## Bibliography

- [1] Fan Xiangdong et al. “Coupled modes of the torsion pendulum”. In: *Physics Letters. A* 372 (2008).
- [2] Leonid Manevitch and Arkadiy I Manevich. *The mechanics of nonlinear systems with internal resonances*. World Scientific, 2005.
- [3] Francesco Clementi, Stefano Lenci, and Giuseppe Rega. “1: 1 internal resonance in a two dof complete system: a comprehensive analysis and its possible exploitation for design”. In: *Meccanica* 55.6 (2020), pp. 1309–1332.
- [4] Keivan Asadi, Jun Yu, and Hanna Cho. “Nonlinear couplings and energy transfers in micro-and nano-mechanical resonators: intermodal coupling, internal resonance and synchronization”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 376.2127 (2018), p. 20170141.
- [5] Aliyu Aliyu Babayo, Mohammad Hossein Anisi, and Ihsan Ali. “A review on energy management schemes in energy harvesting wireless sensor networks”. In: *Renewable and Sustainable Energy Reviews* 76 (2017), pp. 1176–1184.
- [6] Luca Gammaitoni. “There’s plenty of energy at the bottom (micro and nano scale nonlinear noise harvesting)”. In: *Contemporary Physics* 53.2 (2012), pp. 119–135.
- [7] Samir A Emam and Ali H Nayfeh. “Non-linear response of buckled beams to 1: 1 and 3: 1 internal resonances”. In: *International Journal of Non-Linear Mechanics* 52 (2013), pp. 12–25.
- [8] Chunbo Lan, Weiyang Qin, and Wangzheng Deng. “Energy harvesting by dynamic instability and internal resonance for piezoelectric beam”. In: *Applied Physics Letters* 107.9 (2015), p. 093902.
- [9] S-J Jang et al. “Design of a 2DOF vibrational energy harvesting device”. In: *Journal of Intelligent Material Systems and Structures* 22.5 (2011), pp. 443–448.
- [10] Jianxiong Zhu et al. “Development trends and perspectives of future sensors and MEMS/NEMS”. In: *Micromachines* 11.1 (2019), p. 7.
- [11] Tianyi Zhang et al. “Sensitivity enhancement of a resonant mass sensor based on internal resonance”. In: *Applied Physics Letters* 113.22 (2018), p. 223505.
- [12] Larry L Howell. “Compliant mechanisms”. In: *21st century kinematics*. Springer, 2013, pp. 189–216.

- [13] P Thiruvengatanathan et al. “Enhancing parametric sensitivity using mode localization in electrically coupled MEMS resonators”. In: *TRANSDUCERS 2009-2009 International Solid-State Sensors, Actuators and Microsystems Conference*. IEEE. 2009, pp. 2350–2353.
- [14] Ming Lyu et al. “Design and modeling of a MEMS accelerometer based on coupled mode-localized nonlinear resonators under electrostatic actuation”. In: *Communications in Non-linear Science and Numerical Simulation* 103 (2021), p. 105960.
- [15] Xiang-Dong Fan et al. “Coupled modes of the torsion pendulum”. In: *Physics Letters A* 372.5 (2008), pp. 547–552.
- [16] Abu Raihan Mohammad Siddique, Shohel Mahmud, and Bill Van Heyst. “A comprehensive review on vibration based micro power generators using electromagnetic and piezoelectric transducer mechanisms”. In: *Energy Conversion and Management* 106 (2015), pp. 728–747.
- [17] Yunus Uzun and Erol Kurt. “Performance exploration of an energy harvester near the varying magnetic field of an operating induction motor”. In: *Energy conversion and management* 72 (2013), pp. 156–162.
- [18] Alireza Khaligh, Peng Zeng, and Cong Zheng. “Kinetic energy harvesting using piezoelectric and electromagnetic technologies—state of the art”. In: *IEEE transactions on industrial electronics* 57.3 (2009), pp. 850–860.
- [19] Adnan Harb. “Energy harvesting: State-of-the-art”. In: *Renewable Energy* 36.10 (2011), pp. 2641–2654.
- [20] Chongfeng Wei and Xingjian Jing. “A comprehensive review on vibration energy harvesting: Modelling and realization”. In: *Renewable and Sustainable Energy Reviews* 74 (2017), pp. 1–18.
- [21] Steve P Beeby et al. “A micro electromagnetic generator for vibration energy harvesting”. In: *Journal of Micromechanics and microengineering* 17.7 (2007), p. 1257.
- [22] Jiawen Xu and Jiong Tang. “Modeling and analysis of piezoelectric cantilever-pendulum system for multi-directional energy harvesting”. In: *Journal of Intelligent Material Systems and Structures* 28.3 (2017), pp. 323–338.
- [23] Ya Zhang et al. “Giant enhancement in the thermal responsivity of microelectromechanical resonators by internal mode coupling”. In: *Physical Review Applied* 14.1 (2020), p. 014019.

- [24] Farbod Khoshnoud and Clarence W de Silva. “Recent advances in MEMS sensor technology-mechanical applications”. In: *IEEE Instrumentation & Measurement Magazine* 15.2 (2012), pp. 14–24.
- [25] Wei Pang et al. “Piezoelectric microelectromechanical resonant sensors for chemical and biological detection”. In: *Lab on a Chip* 12.1 (2012), pp. 29–44.
- [26] CK O’sullivan and GG Guilbault. “Commercial quartz crystal microbalances—theory and applications”. In: *Biosensors and bioelectronics* 14.8-9 (1999), pp. 663–670.
- [27] G Sauerbrey. “Use of the vibrating quartz for thin film weighing and microweighing”. In: *Z. Phys.* 155 (1959), p. 206.
- [28] Christopher R Kirkendall, Daniel J Howard, and Jae W Kwon. “Internal resonance in quartz crystal resonator and mass detection in nonlinear regime”. In: *Applied Physics Letters* 103.22 (2013), p. 223502.
- [29] YX Liu et al. “Design of a digital closed control loop for the sense mode of a mode-matching MEMS vibratory gyroscope”. In: *The 9th IEEE International Conference on Nano/Micro Engineered and Molecular Systems (NEMS)*. IEEE. 2014, pp. 199–203.
- [30] Atabak Sarrafan, Behraad Bahreyni, and Farid Golnaraghi. “Design and characterization of microresonators simultaneously exhibiting 1/2 subharmonic and 2: 1 internal resonances”. In: *2017 19th International Conference on Solid-State Sensors, Actuators and Microsystems (TRANSDUCERS)*. IEEE. 2017, pp. 102–105.
- [31] Atabak Sarrafan et al. “Demonstration of a nonlinear angular rate sensor based on internal resonance”. In: *2020 IEEE Sensors Applications Symposium (SAS)*. IEEE. 2020, pp. 1–6.
- [32] Warner J Venstra, Ronald van Leeuwen, and Herre SJ van der Zant. “Strongly coupled modes in a weakly driven micromechanical resonator”. In: *Applied Physics Letters* 101.24 (2012), p. 243111.
- [33] Nor Hayati Saad et al. “Analysis of MEMS mechanical spring for coupling multimodal micro resonators sensor”. In: *Microelectronic Engineering* 86.4-6 (2009), pp. 1190–1193.
- [34] Juan A Gallego and Just Herder. “Synthesis methods in compliant mechanisms: An overview”. In: *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Vol. 49040. 2009, pp. 193–214.
- [35] D Farhadi Machekposhti, N Tolou, and JL Herder. “A review on compliant joints and rigid-body constant velocity universal joints toward the design of compliant homokinetic couplings”. In: *Journal of Mechanical Design* 137.3 (2015), p. 032301.

- [36] Sangamesh R Deepak et al. “A comparative study of the formulations for topology optimization of compliant mechanisms”. In: *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Vol. 43260. 2008, pp. 205–216.
- [37] Christine Vehar Jutte and Sridhar Kota. “Design of nonlinear springs for prescribed load-displacement functions”. In: (2008).
- [38] Brian P Trease, Yong-Mo Moon, and Sridhar Kota. “Design of large-displacement compliant joints”. In: (2005).
- [39] Ming Lyu et al. “Exploiting nonlinearity to enhance the sensitivity of mode-localized mass sensor based on electrostatically coupled MEMS resonators”. In: *International Journal of Non-Linear Mechanics* 121 (2020), p. 103455.
- [40] Francesca Cosmi and Alberto Dal Maso. “A mechanical characterization of SLA 3D-printed specimens for low-budget applications”. In: *Materials Today: Proceedings* 32 (2020), pp. 194–201.
- [41] Jenna Gietl et al. “Characterization of damping properties in 3D printed structures”. In: *Journal of Physics: Conference Series*. Vol. 1149. 1. IOP Publishing. 2018, p. 012002.
- [42] Krystyna Januszkiewicz and Marta Banachowicz. “Nonlinear shaping architecture designed with using evolutionary structural optimization tools”. In: *IOP Conference Series: Materials Science and Engineering*. Vol. 245. 8. IOP Publishing. 2017, p. 082042.