J. van Dijk

Flexure Design Optimization for Foldable Structures

Pseudo-Rigid Body Modeling and Multi-Objective Optimization

Delft University of Technology

TU Delft

Flexure Design Optimization for Foldable Structures

Pseudo-Rigid Body Modeling and Multi-Objective Optimization

by

J. van Dijk

Research Assignment

in partial fulfilment of the requirements for the degree of

Master of Science
in Mechanical Engineering

at the Department Maritime and Transport Technology of Faculty Mechanical, Maritime and Materials Engineering of Delft University of Technology

Student number: 4262603
MSc track: Multi-Machine Engineering
Report number: 2021.MME.8568

Supervisor: Dr. J. Jovanova, TU Delft
Date: November 11, 2021

TU Delft

Preface

This research report is written as part of the second year of the Mechanical Engineering master at the Delft University of Technology. This assignment is done within the department of Maritime and Transport Technology, for the track Multi-Machine Engineering, within the theme of Large-scale Mechanical Systems Design. The report is aimed at researchers within the field of compliant mechanisms and large scale structures. This report can be used as a basis for further research into the design of flexures for large scale applications. This research was largely written from home, as the coronavirus restricted the possibilities for on campus work and physical meetings. I would like to thank my supervisor, Dr. J. Jovanova, for her flexible and cheerful support and guidance along the way, even when physical meetings were not possible. Her concrete advice was extremely useful in progressing through the research assignment.

J. van Dijk
Delft, November 2021

Abstract

Compliant mechanisms have found their way in more and more applications in recent years, particularly in the aerospace and micro-systems sectors. The benefits of compliant mechanisms stretch far, and can be exploited for large scale applications as well. Little research has been performed on the use of compliant mechanisms in large scale structures, where large deflections are in play. This while the interest in foldable structures is increasing, take for example the foldable container concept and origami inspired structures. The required hinges can be made compliant, such a hinge is called a flexure.

The aim of this research is to give insight in the design of flexures for large scale applications and large deflections, to aid researchers when designing large foldable structures. The research question that is answered in this report is: how can flexure design be optimized for foldable structures? To answer the research question a Pseudo-Rigid Body Model (PRBM) is made in Python, along with a multi-objective optimization.

Compliant mechanisms can offer increased performance, however, the design of compliant mechanisms is more complicated than rigid body mechanisms. Therefore, an intuitive method for designing compliant mechanisms is required. The Finite-Element Method is considered for this purpose, but is too complex and not suited for initial design stages. Therefore, a PRBM is selected for the design analysis, as it is less complex and well suited for large deflection members and ideal for initial design stages. A PRBM with two revolute joints is used, because it provides a higher accuracy than a model with one revolute joint and it simplifies the iterative process of a model with three revolute joints, while maintaining a similar level of accuracy. The load case that is evaluated in this research is a moment end load.

A sensitivity analysis is performed for the flexure behaviour regarding the flexure's Young's modulus, length, width, thickness and end moment load. This analysis shows that the flexure length and moment load, linearly influence the flexure deflection, i.e. twice the length, gives twice the deflection. Furthermore, the flexure deflection is reverse linearly dependant on the Young's modulus and width, i.e. twice the width, gives a twice as small deflection. The flexure is most sensitive to a change in thickness. For a thickness increase of factor n, the deflection decreases by a factor n^3.

Furthermore, the design of the flexure is evaluated for three material types, being aluminium, rubber and a Nickel–Titanium Shape Memory Alloy (SMA). The material changes the behaviour of the flexure because of the material's Young's modulus and yield strength, which prescribes the allowed load. Therefore, it is shown that the measure of flexibility, $F = \frac{\sigma_y}{E}$, is a good indicator for suitable flexure materials.

When taking the results from the sensitivity analysis and implementing the importance of the flexure material's yield strength, a design constant can be deduced for viable and intuitive flexure design. This constant considers all the variables influencing the flexure design and assumes maximum allowable load on the flexure. With this constant researchers can quickly identify feasible flexure designs and the result of parameter changes are made more intuitive.

The multi-objective optimization for the flexure design is performed with Non-dominated Sorting Genetic Algorithm II. The optimization can find the Pareto-optimal solutions for the objectives of shape error, stress and volume, while varying the flexure thickness. The evaluated configurations are for a single flexure and for flexures implemented in a large scale compliant mechanism. For the single flexure target shape, a clear Pareto-front is found, depicting the required trade-off between the objective functions. Furthermore, the optimization visualizes the super elastic properties of the SMA and its effect on the deflection when actuated optimally. The results for the use of SMA flexures in a compliant mechanisms are shown as well. It shows that with the right boundary conditions the superelasticity properties of the SMA can be exploited for optimal compliant structure design. From this it can be concluded that this optimization can be used for design of complicated foldable structures, while taking into account the accuracy of the compliant mechanism's motion.

For future research it is proposed to visualize the behaviour of a flexure with a different cross-section than a rectangular beam-like cross-section. Furthermore, more research is required into the practicality

of flexures for large scale applications, since the out of plane stiffness can become too low, risking the planar motion of flexures. Other directions that require more research, are the influence of micro slip and unloading effects on the flexure behaviour.

List of Abbreviations

2R	Revolute-Revolute
FEM	Finite Element Method
NiTi	Nickel–Titanium
PRBM	Pseudo-Rigid Body Model
SMA	Shape Memory Alloy

Contents

1 Introduction

Compliant mechanisms have found their way in more and more applications in recent years, particularly in the aerospace and micro-systems sectors. Better understanding of their behaviour and technological developments have accelerated this growth and have increased the awareness of the benefits [11]. Where previously compliant mechanisms needed to be designed through trail and error stages, or computationally intensive finite-element methods, now techniques exist to approximate the elastomeric structure and forces, making the calculations for analysis simpler [3].

Most of the compliant mechanism applications have been in the high-tech, small scale and light weight applications. This is because compliant mechanisms offer excellent properties for these areas, such as high precision, low part counts and the lack of need for lubrication. However, the benefits of compliant mechanisms stretch further, and can be exploited for larger scale applications as well. For example the lower cost, ease of fabrication and reduced need for maintenance, are properties that benefit every scale of application.

Little research has been performed on the use of compliant mechanisms in large scale structures, where large deflections are in play. This while the interest in foldable structures is increasing, take for example the foldable container concept [12] and origami inspired structures [17]. In applications like these the required hinges can be made compliant, thereby making it a compliant mechanism. Such a compliant hinge is called a flexure. The focus of this research is on this type of compliant mechanisms, particularly on the large scale application of flexures.

The aim of this research is to give insight in the design of flexures for large scale applications and large deflections, to aid researchers when designing large foldable structures. With these insights foldable structure design is made more intuitive. Furthermore, an optimization will be presented for large deflection flexure design in a foldable structure application.

The research question that is answered in this report is: how can flexure design be optimized for foldable structures? To get to a quantifiable answer, a model of a flexure is constructed in Python and evaluated for different parameters. Furthermore, an optimization for flexure thickness is made to demonstrate optimization in foldable structure design.

In the context of this research assignment, the term small scale applications refers to applications with a size ranging from millimeters up to several centimeters. Large scale application refers to sizes ranging from several centimeters up to multiple meters.

The structure of the report is as follows: firstly, in chapter two, the general concept of compliant mechanisms is introduced and their characteristics. Secondly, in chapter three, the method in which the research question is answered will be discussed. Subsequently, in chapter three the model that is used to model a flexure is described along with the optimization. Thereafter, the results are presented in chapter four. Lastly, in chapters five and six the discussion and conclusions are presented.

2

Introduction to Compliant Mechanisms

In this chapter the concept of compliant mechanisms is further introduced. First the general definition of compliant mechanisms is established, after which the application possibilities are discussed. Subsequently, the advantages and disadvantages surrounding compliant mechanisms are described. Thereafter the difficulties surrounding the design of compliant mechanism are discussed.

A single definition of compliant mechanisms is not universally utilized, commonly when talking about compliant mechanisms, the definition boils down to a mechanism which gets mobility through deformation of elastic components [11]. The mobility can come from flexible structural members or from an elastically flexible slender region, a flexure hinge, in between rigid bodies [13]. According to Lobontiu [13], flexures are the main type of compliant mechanisms and they perform a similar function to a bearing whilst having limited rotation capability. This research is focused on this specific compliant hinge subset of compliant mechanisms.

The idea of using compliant mechanisms instead of rigid-link devices is inspired by nature. Although early mechanisms already implemented compliant mechanisms, like hunting bows [11], the concept did not find its way into later structural and technological applications. Mechanisms with simple and more conventional joints and rigid links got preferred over the compliant mechanisms because of the more straightforward design and analysis [11]. In the last twenty years researchers have become better in analysing and designing compliant mechanisms due to improvements in computational software and hardware. This enabled better analysis of the mechanism motion and stresses. Furthermore, new materials and production techniques, well suited for use in compliant mechanisms, are available [11].

Currently compliant mechanisms are used in all sorts of industries. Compliant mechanisms feature a lot of properties that are attractive to the field of micro- and nano-engineering, opto-mechatronics, aerospace, semiconductor and health-care [6]. These properties come from the fact that compliant mechanisms are monolithic, which gives them high precision, ideal for mechanisms that require fine alignment [6].

Lobontiu [13] describes in his book much more applications, a few applications are mentioned here to give an idea of the broad applicability and scale of compliant mechanisms. In the automobile industry compliant mechanisms can be used as sensors and suspension systems, while the biomedical industry utilizes compliant mechanisms as vascular catheters. In fiberoptics and microdevices the compliant mechanisms are used as disc drive suspension, laser systems, load cells and optical switches [13].

It can be noted that all these applications are small scale implementations of compliant mechanisms, the largest of these mechanisms are several centimeters in size. Compliant mechanisms are being used in small scale applications for good reason, the benefits of compliant mechanisms are particularly useful in tight spaces and precision applications. However, the use of compliant mechanisms and its benefits are not limited to these scales. In the following sections the advantages and disadvantages are discussed in further detail.

2.1. Advantages of Compliant Mechanisms

There are multiple advantages associated with compliant mechanisms which makes their use case really attractive. One of them is the potential for increased performance [11]. This can be achieved because of reduced surface and adhesive forces which are dominant at small scales, making compliant

mechanisms predictable, ideal for high precision applications [13]. Furthermore, the increased performance comes from the reduced wear, eliminating the need for lubricants, which is ideal for space applications, where lubricants can evaporate [4]. Reduced wear also means reduced maintenance, which saves cost and is beneficial for hard to reach locations. Then there is the benefit of reduced part count, which leads to easier assembly and fewer parts to stock [4]. Less parts, lead to more compact and light weight mechanisms as well.

2.2. Disadvantages of Compliant Mechanisms

There are some drawbacks associated with compliant mechanisms as well. The motion of compliant mechanisms is complex to define and is often non linear [4, 11]. For example, the rotation center of a compliant hinge is not fixed. Also, axial shearing and torsion loading ask for complicated equations to fully understand the motion [13]. Another drawback is their sensitivity to fatigue, due to repeated motion and large deflections fatigue can occur, compromising the mechanism's operation and lifetime. Furthermore, the range of motion of compliant mechanisms is much more constraint than their rigid body counterparts, only allowing for infinite rotations. Other disadvantages are their sensitivity to temperature variations [13] and their perceived flimsiness [11].

When these drawbacks can be overcome, compliant mechanisms are a promising design direction to be considered, for small scale and large scale applications. The design of compliant mechanisms is more complicated than rigid body mechanisms however. Components of compliant mechanisms often have to combine certain functions, that could earlier be assigned to multiple components. The linear paths and well defined guided motion of rigid body mechanisms have to be replaced by bending, non linear, unclear axis of rotation mechanisms. In chapter 3, the way to model these complexities for the design of compliant mechanisms is discussed, along with a method to optimize the design for large scale foldable structure applications.

3

Methods

In order to answer the research question, first a model needs to be selected to represent the flexure. This model needs to be as close to reality as possible without being too computationally complex. With the model it is possible to see how changes on certain parameters, change the behaviour of the flexure. These insights lead to efficient flexure design by making the influence of material choice and design parameters directly visible and intuitive.

Subsequently the parameters influencing the design need to be identified. A look is taken into the sensitivity of the flexure behaviour, to these parameters. When the sensitivity is known, one can predict what designs or materials are worth investigating further. To investigate the influence of the flexure material on the flexure behaviour, three materials are selected to be used in the model. These three materials are selected because they are commonly used in structural and bending applications.

Inherent to a change in material is the change in yield strength. The yield strength is an important property of a material, as it indicates the stress limit to where a material can be deformed until it deforms plastically. For flexures this is important because flexures need to be able to withstand multiple load cycles without losing its functionality. The yield strength has a significant influence on the flexure design because it is a direct measure of how much load the flexure of a particular material can carry. When the yield strength of a material is low, it means the flexure can not bare much load, so the deflection for a flexure of that material can be much lower.

After the model has been selected and the important parameters are identified, an optimization is performed for the flexure design. This is done through a multi-objective target shape optimization, where the optimal design is determined for a certain predefined shape. With this optimization the Pareto-optimal thickness of the flexure is determined. The thickness is optimized while taking into account the stresses and flexure volume.

3.1. Flexure model

As mentioned in chapter 2, compliant mechanisms are more complicated to design than conventional rigid body mechanisms. The complex motion of bending parts and the ability to perform multiple tasks with fewer parts, are the source for this complexity. The model that will be used to approximate this behaviour needs to capture these behaviours as accurately as possible. The main methods, of modeling flexure behaviour, are Finite Element Method (FEM) and Pseudo-Rigid Body Models (PRBM). Both methods are discussed below and the most suitable method is selected to be implemented for the use case.

3.1.1. Finite Element Method

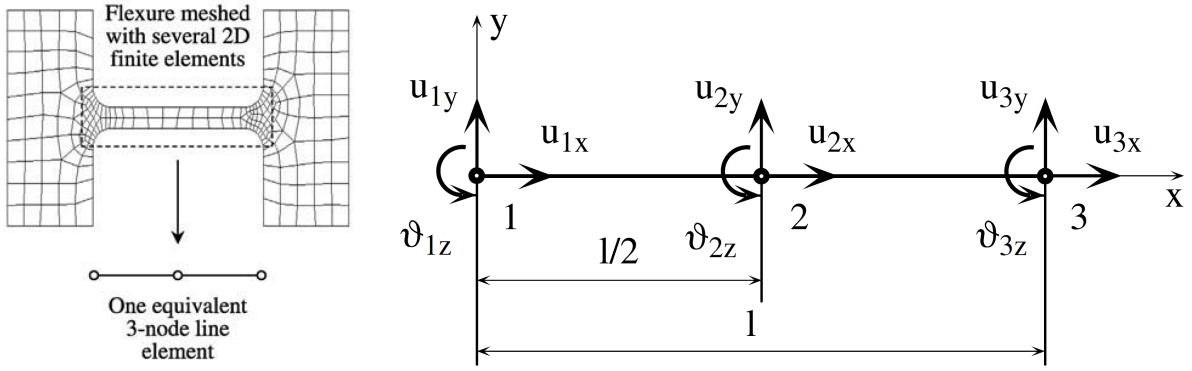
The Finite Element Method (FEM) is a numerical technique used in multiple engineering industries. It can be used for solid mechanics, fluid dynamics, heat transfer and electromagnetic problems [7]. A large benefit of the FEM is that it can be used for all sorts of problems, for complex materials, loads and geometries. The basic working principle is that a complex geometry can be divided into a finite set of discrete elements, in that way equilibrium requirements only have to be satisfied for the simple elements instead of entire complex geometries, making the equations simpler whilst still accurately representing the complex geometry.

The discretization of complex geometries has the benefit of being able to account for changes in

material properties along a structure, also, it can easily deal with changing geometries in a structure, for example a varying cross section of a beam. The changes in material properties and geometries can be modeled by assigning different properties per element.

The equilibrium requirements for each element make up the stiffness matrix for the element. This matrix gives the relation between the forces and displacements of the element. By combining the matrices of every element in the correct way, the global stiffness matrix can be defined. With the global stiffness matrix, the forces and displacements of the entire geometry can be determined, while capturing local effects.

The FEM method is very flexible, it provides a wide choice of analysis types and several levels of complexity to choose from [13]. A FEM model is useful when the geometry and loads are exactly known and an initial design is available. With a highly complex FEM model, very complex shapes and load types can be studied very accurately. This requires a lot of knowledge about FEM and multiple complex equations, which are integrated in commercially available software. When doing these equations yourself, a lower level of complexity is practically reachable, enabling for less accurate models. A depiction of an element made by software and a simpler equivalent element can be seen in figure 3.1a.



(a) Simple element to replace commercial software element

(b) Three-node, three-degrees-of-freedom-per-node FEM

Figure 3.1: FEM for single axis beam element flexure[13]

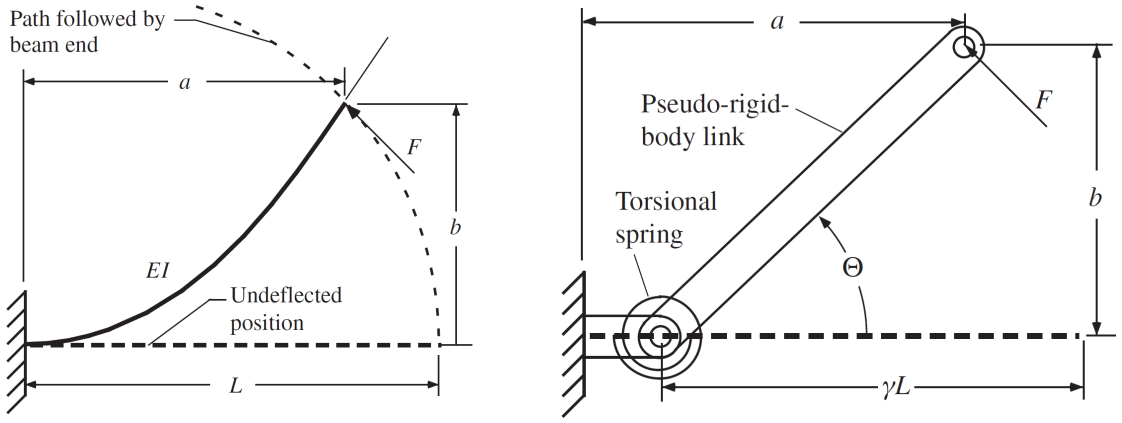
According to Lobontiu [13], the FEM model for a beam element flexure has to include three nodes per element to accurately describe the nonlinear behaviour of a flexure undergoing large deformation. Such a model is represented in figure 3.1b. Because of the third node, a fifth degree polynomial is required to provide continuity for the bending conditions, i.e. per node a boundary condition must be enforced for the deflection and for the slope. So already for such a simple element, complicated calculations are required.

3.1.2. Pseudo-Rigid Body Model

Another method of modeling the nonlinearities introduced by large deflections is through a Pseudo-Rigid Body Model (PRBM). A PRBM is a well suited for evaluation, optimization and visualization of compliant mechanism designs [10]. The PRBM is highly efficient in early design stages where many different designs need to be trialed, since it does not require an accurate initial design. Resulting designs can be refined by using other analysis types [4].

The PRBM simplifies the analysis of complex motion by replacing the flexible elements by rigid segments and torsion springs. Rigid segments and springs are easier to analyse and can be used to predict the force-deflection relationships. In a PRBM the classical rigid-body mechanism and compliant theories are unified [4]. Instead of modeling point by point variations, PRBMs only describe the behaviour of whole compliant segments [11]. A simple use case and its equivalent PRBM are depicted in figure 3.2. Here it is clearly visible how a flexible element can be replaced by a torsion spring and a rigid element to deduce the behaviour of the flexible element.

Through an optimization process, the exact force-deflection relationships are used to determine the model parameters for a specific use case. The exact relationships can be determined through elliptical integrals or other numeric methods [11]. The model parameters have to be selected so that



(a) Cantilever flexure with force at free end

(b) Equivalent PRBM

Figure 3.2: Simplification of large deflection compliant mechanisms[11]

the rigid links, connected by a characteristic pivot, approximate the flexure displacement, while the spring stiffness has to approximate the flexure stiffness. The location of the characteristic pivot is different for every load-case and geometry, just as the spring stiffness, these parameters need to be recomputed for every change in boundary conditions. This makes a PRBM difficult to use for dynamic simulations [19].

Shearing effects and axial loading are aspects that complicate the construction of PRBMs. These types of loading can not be modeled by a simple torsion spring. To account for this, mitigating additions have to be implemented in the model, such as extension springs. A beam geometry factor can give insight in the accuracy of the PRBM in complex load cases. The factor relates the extension of the beam element along the neutral axis with respect to the deformation due to bending [18].

The PRBM is developed under the assumption of flexures undergoing large deformations. However, oftentimes flexures are not used in large deflection situations, therefore the models are less accurate when the deflections are small [13]. Other limiting factors of the PRBM are, they currently only exist for constant cross section elements and can only describe planar motions [19].

3.1.3. Selected model

A PRBM has been chosen to be used in order to answer the research question. The main reason being its ease of use, visualization abilities and appropriateness for optimization purposes. Furthermore, local variations are not relevant for this research and boundary conditions are fixed. Also, the intended application of the model is to examine flexure behaviour for large deformations, eliminating inaccuracies of the model at small deformations. The insights of this research are meant to be used in early design stages when initials designs still feature simple geometries, making the exact and complicated equations of the FEM superfluous.

To accurately represent the flexure behaviour during the analysis, a suitable PRBM needs to be selected. A multitude of models exist, ranging from simple to complex and from inaccurate to accurate. Some models can be more accurate than others depending on the load case. For this research assignment, a PRBM with two revolute (2R) joints and three rigid segments is selected [21]. At the joints a torsion spring is implemented to simulate the flexure stiffness. A depiction of the model can be seen in figure 3.3. The flexure model will be based on a beam configuration.

The 2R model is selected because the added degree of freedom over a model with one revolute joint, leads to a significant reduction in error between the model and the exact solution [21]. Furthermore, the model with one revolute joint can only describe the path of the flexure tip, not the deflection angle. On the other hand, more degrees of freedom lead to higher computational cost, which at a certain point increases too much, without decreasing the error significantly. The 2R model strikes a good balance, since it provides a higher accuracy than a model with one revolute joint and it simplifies the iterative process of a model with three revolute joints, while maintaining a similar level of accuracy [22].

When extension effects are in play, the accuracy of a PRBM without a prismatic joint decreases.

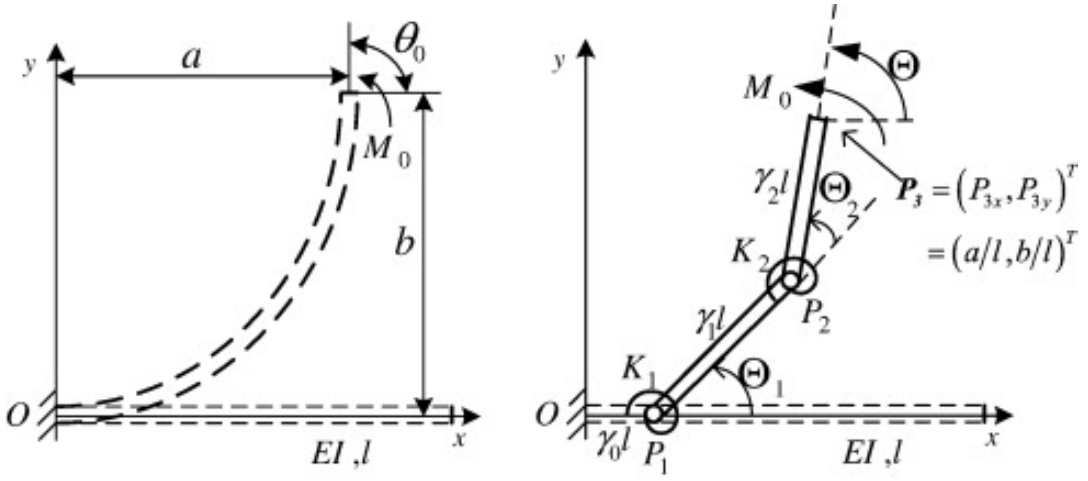


Figure 3.3: Cantilever beam with end moment and corresponding 2R PRBM [21].

The revolute joints with torsion spring are not able to account for the longitudinal forces in a flexure [19]. These effects are more significant when evaluating short, soft flexures, which are not being considered in this research [20]. Furthermore, the 2R model is accurate for long beam configurations.

The 2R model parameters are dependant on the load case. For this research the selected load case is a moment end load on the flexure. The model parameters associated with such a load case are displayed in table 3.1. These values are found through parametric approximations to the exact deflection path and by linear regression for the spring stiffness coefficients [21].

Table 3.1: Model parameters 2R PRBM with moment end load

γ_0	0.16
γ_1	0.66
γ_2	0.18
$K_{\theta 1}$	2.0571
$K_{\theta 2}$	1.9175

Here γ_i stands for the characteristic radius factor. The length of each rigid link is $\gamma_i * l$, where l is the length of the flexure, as can be seen: $\gamma_0 + \gamma_1 + \gamma_2 = 1$. $K_{\theta i}$, is a nondimensionalized torsion spring constant, called the stiffness coefficient.

Via the 2R PRBM the deflection of a flexure can be approximated in the following way: first the spring constants, K_i, need to be determined, that can be done through the characteristic radius factors and the stiffness coefficients, as follows:

$$K_i = \gamma_i K_{\theta i} \frac{EI}{l}, \quad (i = 1, 2) \tag{3.1}$$

Here E, stands for the Young's modulus of the flexure material. I, stands for the moment of inertia of a flexure, for a beam being $\frac{wt^3}{12}$, where w stands for the width of the beam and t for the thickness.

After the spring constants are determined, the deflection angles of the torsion springs can be determined, Θ_i respectively, via:

$$\begin{aligned} \Theta_1 &= \frac{M_0(1-\gamma_0)}{K_1} \\ \Theta_2 &= \frac{M_0\gamma_2}{K_2} \end{aligned} \tag{3.2}$$

Here M_0, is the moment end load applied to the flexure. The numerator in equations 3.2 gives the torque at the respective joint locations. The total deflection of the model can be calculated by the summation of both deflection angles, i.e. $\Theta_1 + \Theta_2 = \Theta$.

Subsequently, with both deflection angles known, the coordinates of the flexure tip can be calculated, being P_{3x} and P_{3y}:

$$P_{3x} = (\gamma_0 + \gamma_1 \cos\Theta_1 + \gamma_2 \cos\Theta)l \tag{3.3}$$

$$P_{3y} = (\gamma_1 \sin\Theta_1 + \gamma_2 \sin\Theta)l \tag{3.4}$$

3.2. Design parameters

To make the design of flexures intuitive, it is useful to perform a sensitivity analysis for the behaviour of the flexure. With the sensitivity analysis it is possible to see how the flexure reacts when certain parameters are changed. Several parameters have been identified to be of influence on the flexure behaviour, being: the flexure length, thickness and width, the moment end load and the flexure material. All of these parameters can be seen back in the equations forming the model. Another parameter of importance, not used in the model, is the yield strength of the flexure material.

The materials that are being used to research the flexure behaviour are aluminium, rubber and a Nickel–Titanium (NiTi) Shape Memory Alloy (SMA). Aluminium has been chosen because it is a common material in structural applications and widely available, furthermore, aluminium is a relatively stiff material and intuitively not best suited for large deflection applications. Table 3.2 shows the properties of the selected aluminium.

Rubber has been selected because of its flexible nature, it forms a contrast with aluminium in terms of the bending capability. It will be interesting to see how this influences the flexure design. The selected rubber type is polyurethane, since it is the strongest of the elastomers [8]. Its properties are listed in table 3.2.

The NiTi SMA has been selected because it has different properties than regular materials. The stress-strain curve is highly non-linear and includes a variation in Young's modulus. SMAs get their name from their shape memory properties, but next to that they exhibit unusual elastic properties, the superelastic properties. The superelasticity is related to the material phase transformation, from austenite to martensite. Due to the superelastic properties, SMAs can withstand large deformations relative to its weight. Furthermore, the SMA damping properties can be exploited for new design directions in flexures [2].

The SMA properties are interpreted according to the model of Auricchio et al. [1], and feature a stress value for the forward and reverse phase transformation of the SMA, defining the superelastic regime. This model is depicted in figure 3.4. The NiTi alloy properties are highly sensitive to the fabrication process, values used for this research are from Machado et al. [15], shown in table 3.2. For this research only the loading phase is considered. The yield strength values are obtained through lab tests on 50Ni50Ti at Delft University of Technology.

Table 3.2: Flexure material properties

Material	Young's modulus, E [GPa]	Yield strength, σ_y [MPa]
Aluminium	70	276
Rubber (Polyurethane)	0,03	51
NiTi SMA	60	1200
	Young's modulus, E [GPa]	Superelastic regime, σ [MPa]
NiTi SMA Superelastic	25	500-580

3.3. Optimization

When the model for flexure design has been established for quick and preliminary flexure design, the design can be optimized. An optimization can be made for a single flexure use case and for an use case in a structure. A target shape optimization is used to identify the optimal design. With this optimization it is possible to see for what value of the variables the target shape is best met. As a results, a design can me made, where for a given actuation load, the desired deflection is met as closely as possible.

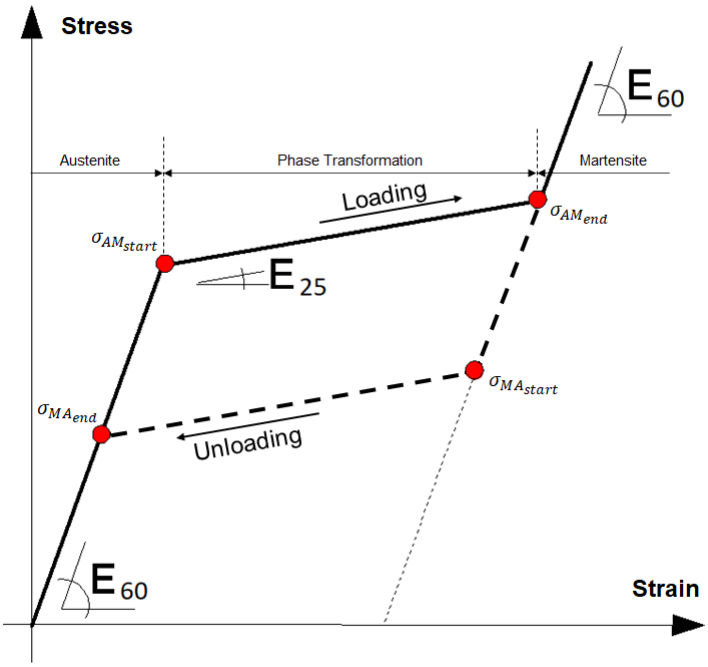


Figure 3.4: SMA material model. red dots represent stress levels of phase transformation

The optimization needs to be able to function when a desired deflection is given, and an initial configuration. The optimization has to take into account allowable stresses in the flexure and the flexure volume. The stresses need to be considered in the flexure because the resulting design has to be feasible, the flexure should not deform plastically under the desired deflection. Also, the volume needs to be considered, because design space is limited and less material means less costs. Moreover, the design should meet the target shape as close as possible. To satisfy these objectives a multi-objective optimization is required.

The algorithm used, is a Non-dominated Sorting Genetic Algorithm, called NSGA II [5]. NSGA II can find a large spread of solutions while converging near the true Pareto-optimal front, thereby representing the trade-off between multiple objectives. The way the algorithm works is by minimizing each objective function at the same time, giving a Pareto front. The optimal design is somewhere on the Pareto-optimal front, depending on the trade-off the user wants to make. A trade-off has to be made because a solution where the shape error is minimal and the stress in minimal, does not exist. The optimum lays somewhere in between.

The optimization is setup in such a way, that the thickness of the flexure is variable, so the algorithm will output an optimal thickness. Furthermore, in the case of the SMA, the Young's modulus is variable, depending on the stress levels in the flexure, as seen in figure 3.4. The thickness is set as the variable because the behaviour of the flexure is most sensitive to it. Moreover, implementing the flexure length as variable would give complications for determining the target shape. For the algorithm to be able to work, it requires the material properties, flexure length, flexure width, end moment and target coordinates as input. A flowchart of the used optimization algorithm can be seen in figure 3.5.

3.3.1. Objectives

For a flexure design to be optimal it has to satisfy multiple objectives. The first objective, f_1, accounts for the minimization of the shape matching error, $S_{e,p}$. $S_{e,p}$ is defined as the distance between the target point, specified by the target shape, and the corresponding model output point. The second objective, f_2, accounts for the minimization of the maximum overall bending stress, $\sigma_{max,n}$. The final objective, f_3, accounts for the minimization of the total flexure volume, V_n.

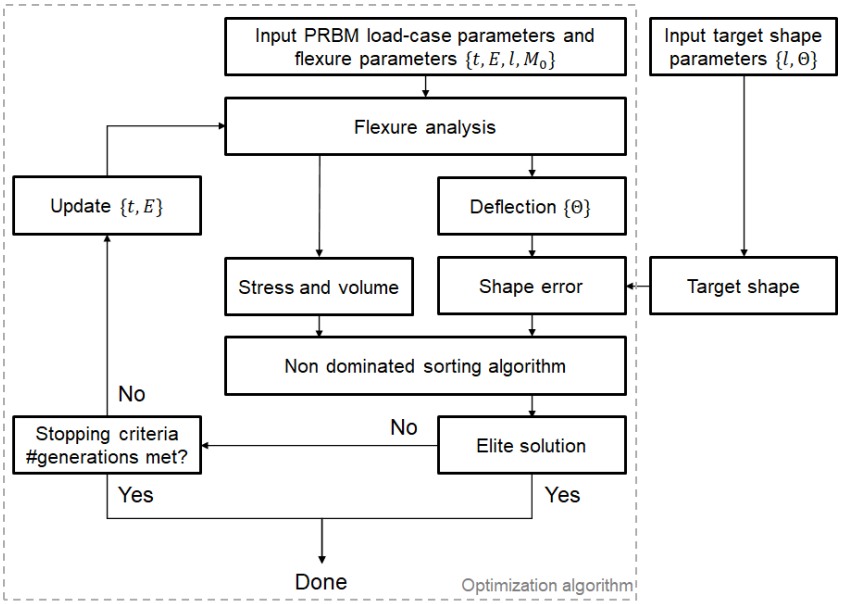


Figure 3.5: Flowchart of the flexure thickness optimization algorithm

The mathematical formulation of the algorithm is as follows:

$$Minimize(f_1, f_2, f_3)$$

Where,

$$f_1 = \sum_{p=1}^{p} S_{e,p} \tag{3.5}$$

$$f_2 = \sum_{n=1}^{n} \sigma_{max,n} \tag{3.6}$$

$$f_3 = \sum_{n=1}^{n} V_n \tag{3.7}$$

Here p, stands for the number of target points, and n, stands for the number of flexures. The algorithm is subject to a predefined maximum and minimum allowable flexure thickness, t_n. The allowable values are determined based on the geometry continuity and fabrication limitations, being one millimeter as minimum thickness and ten millimeter as maximum. This constraint is formulated as follows:

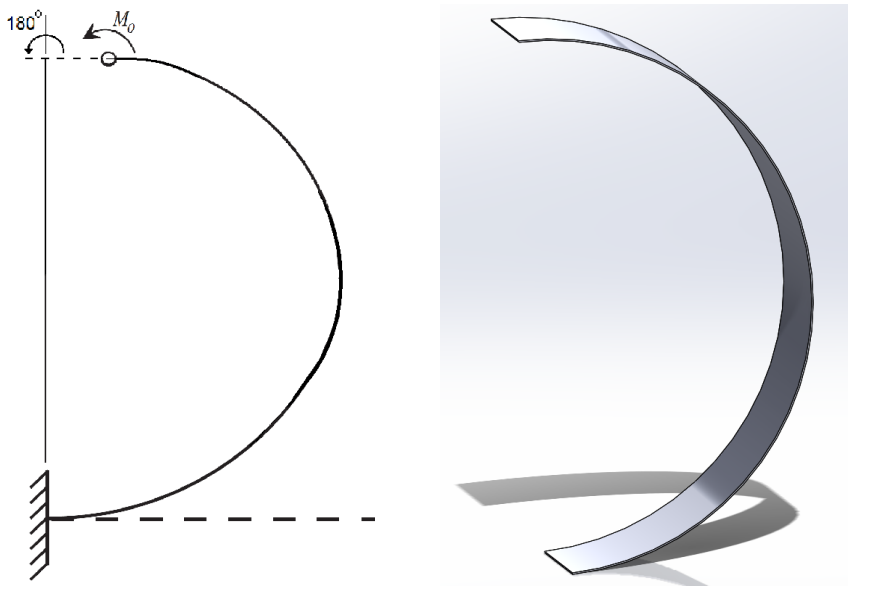
$$1(mm) \leq t_n \leq 10(mm) \tag{3.8}$$

3.3.2. Target Shape

For the optimization a target shape is chosen which reflects a potential application in a foldable structure. The target shape is set to represent a large flexure undergoing a large deformation. Hereby the behaviour of a flexure in extreme situations is simulated. In this way the model covers a multitude of application possibilities. In order to represent the design for a larger structure, multiple flexures are connected with rigid members in between. This structure shape will also be evaluated by the optimization in order to find the optimal design for the structure.

The target shape for a single flexure is set to represent a deflection of 180° upwards, a schematic depiction can be seen in figure 3.6a. In addition to the desired deflection, the flexure length influences the target shape coordinates. For every material the feasible deflection changes, since the material's Young's modulus influences the amount of deflection, and the yield strength limits the feasible deflection. Therefore, the material choice influences the flexure length, which influences the target shape

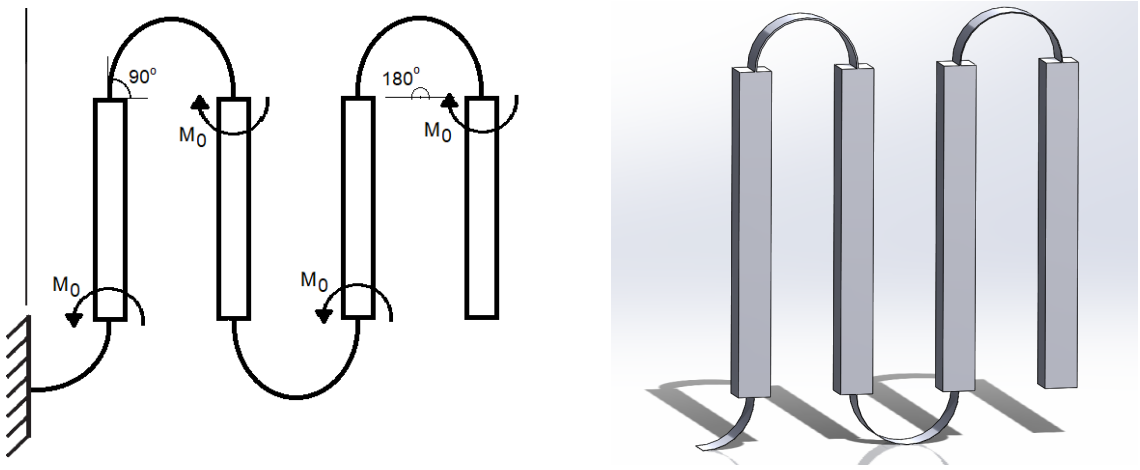
coordinates. The target coordinates are set as the end point of the flexure. Figure 3.6b, represent a one meter long flexure, with a width of twenty centimeters. This depiction can only be used to get a rough idea of what the target shape looks like and is not an exact representation of the flexure behaviour, since the center of rotation is not fixed for an actual flexure.



(a) Schematic single flexure target shape (b) Single flexure target shape (fixed center of rotation)

Figure 3.6: Target shape 180° deflection

Figure 3.7a and 3.7b visualize the implementation of flexures in a potential mechanism application. The mechanism represents a compliant arm which can extend or detract under influence of an applied moment load to the end of the flexures. For this configurations it is assumed that the moment loads are equal at every flexure tip point. Furthermore, the flexures have the same length, except for the first flexure, connected to the ground, which is half the length of the other flexures. In figure 3.7b, the flexures have a length of one meter. The segments connecting the flexures are assumed rigid and have a length of two meters. This configuration is evaluated by the optimization as well, to get to an optimal design. The start and end point of every rigid segment are set as the target coordinates.



(a) Schematic connected flexures target shape (b) Connected flexures target shape

Figure 3.7: Compliant structure target shape

4

Results

The results of the sensitivity analysis and the optimization are presented in this chapter. First the functionality of the model is shown, with a depiction of how the model generates the total deflection when multiple bodies are connected. Thereafter, the model response is shown for the different variables. From these results the sensitivity can be analyzed. Subsequently, the model deflections are presented while taking into account the yield strength of the material. This gives the feasible deflection per material type. When combining this with the sensitivity analysis, a constant can be deduced, which can be used for intuitive design of flexures.

Furthermore, the optimization results of a flexure design for a specific use case are presented. First the results are shown for the design of a single flexure, thereafter, the results are shown for the flexure design in a structure application. To determine the best design a trade-off has to be made between the different objective functions, this process is discussed as well.

4.1. Model output

Figure 4.1, shows the output of the model, the complete code can be found in appendix A. What can be seen is the behaviour of a flexure for varying positive end moment loads. The red lines represent the configuration of the flexure at different loads, the start load, the end load and, where applicable, a load in between. The red dots represent the location of the characteristic pivots for each configuration and the end point of the flexure. As can be seen, since the model is a 2R PRBM, there are two characteristic pivots per flexure. The blue line represents the locus of the flexure tip, where each dot represents a discrete load level. As can be seen, each configuration has the first rigid member in common, starting from the origin.

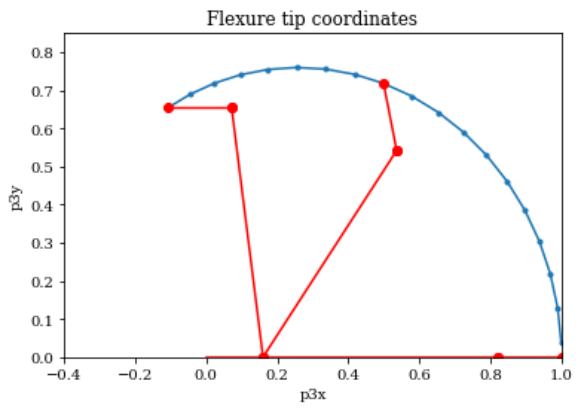
Figure 4.1b shows in what way the model gets to the overall deflection when multiple flexures are connected. The blue and orange lines depict the local behaviour of a single flexure. The green line shows the global behaviour of the system, where it combines the deflections of both local solutions.

Figure 4.1c shows the output of the model for the compliant structure use case, the code for this model can be found in appendix B. In this depiction the flexure length is one meter, except for the first flexure, which is half that. For this figure the material properties of aluminium are used, $E = 70$ (GPa) and a flexure width of 20 centimeters. For this configuration for the flexure to reach 180° deflection, a moment end load of 184.2 (Nm) is required.

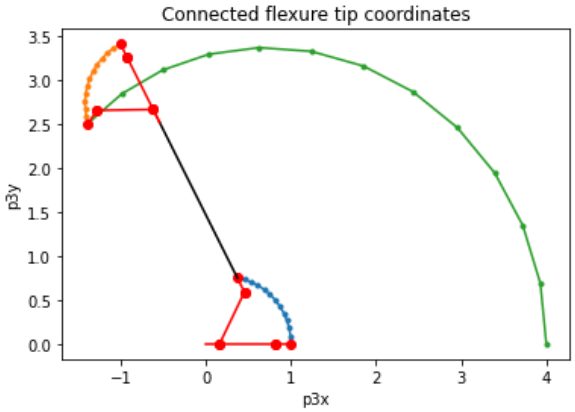
4.1.1. SMA flexure

Figures 4.1a - 4.1c, show the model results for materials with a constant Young's Modulus, i.e. materials with a linear stress-strain curve. The Young's modulus is not constant for NiTi SMAs, resulting in a nonlinear deflection curve. Figure 4.1d shows the nonlinear deflection path of a NiTi SMA subject to varying loads. Since the Young's modulus of NiTi depends on the bending stress it changes under specific loads.

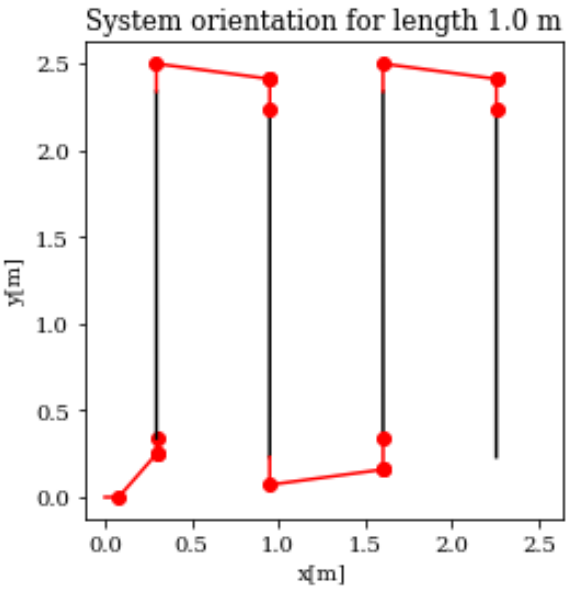
In figure 4.1d, the red lines represent the flexure deflection in the first austenite phase, here the Young's modulus is equal to 60 (GPa). Then the bending stress reaches the limit where the NiTi SMA phase changes from austenite to martensite, reducing the Young's modulus to 25 (GPa), the changeover phase is depicted by the green lines. The moment end load in the changeover phase ranges from 17-19 (Nm). Subsequently, when the bending stresses become too high, the NiTi SMA



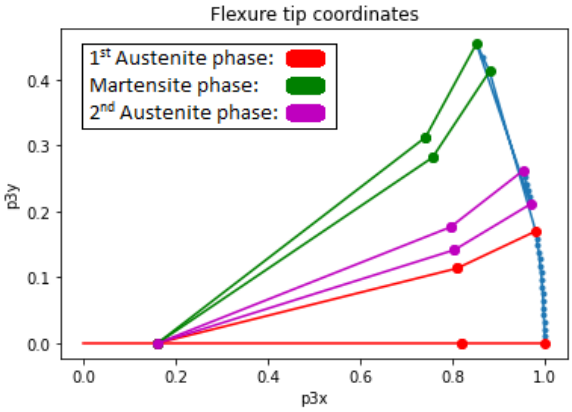
(a) Single flexure varying load



(b) Connected flexures varying load



(c) Multiple connected flexures



(d) Flexure behaviour NiTi SMA 0Nm ≤ M_0 ≤ 25Nm

Figure 4.1: 2R PRBM output

reaches the martensite phase, depicted by the magenta lines, where the Young's modulus is equal to 60 (GPa) again. In this phase the material behaves like any other material, since the Young's modulus is constant.

As can be seen, the NiTi SMA has a superelastic property in a certain load range. This can be exploited in compliant mechanisms when the flexure is designed so that it operates in this region.

4.2. Sensitivity Analysis

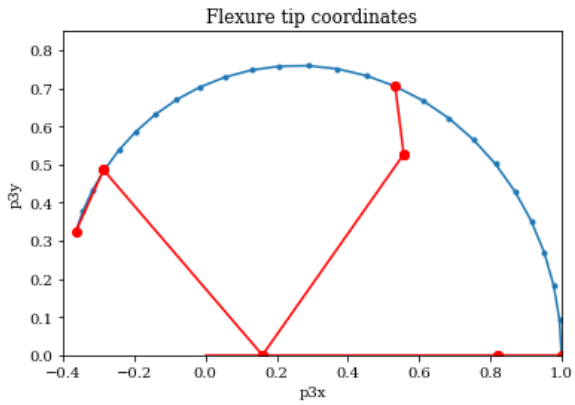
In this section the results of the sensitivity analysis are presented. The sensitivity of the flexure deflection to each variable is analyzed. For each plot in figure 4.2, the variables are the same, except for the variable of which the sensitivity is being investigated. The model variables used for the plots are listed in table 4.1. For the plots where the variables are not fixed, the range and step size are denoted in table 4.1 as well.

From the plots and the step sizes some interesting insights can be gained. Firstly, from the end moment and flexure length sensitivity, figures 4.2a and 4.2c, it can be seen that for larger values, the deflection is larger in a linear way. It is good to note that figure 4.2c represents the relative deflection of the flexure. The Young's modulus and the flexure width, figures 4.2b and 4.2d, are revers linearly related to the flexure deflection, i.e. when the values become larger, the deflection decreases linearly.

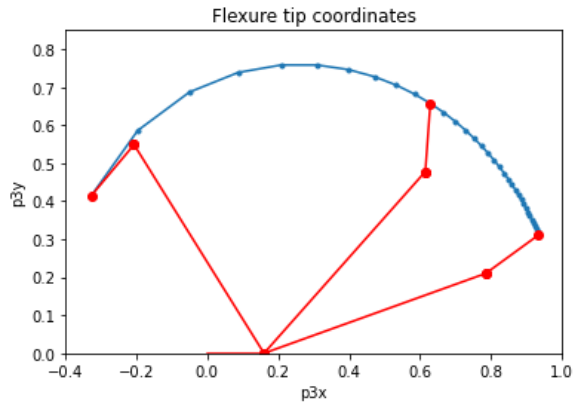
The influence of the flexure thickness on the deflection is more complicated however. It can be seen that when the thickness increases with a factor n, the deflection decreases by a factor of n^3.

These results correspond with the expected behaviour of the model. The model behaviour can be predicted by combining equations 3.1 and 3.2, and taking for I the moment of inertia of a beam, $\frac{wt^3}{12}$. When removing the constants from the equations, the formulation of the deflection boils down to:

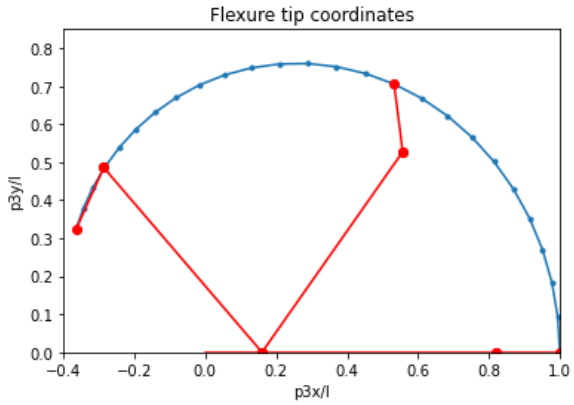
$$\Theta = \frac{M_0 l}{E w t^3} \tag{4.1}$$



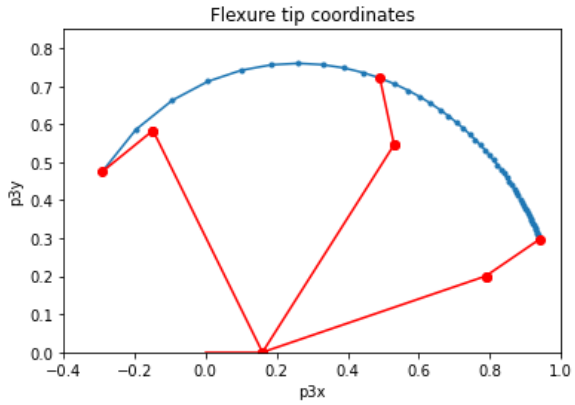
(a) Flexure behaviour 0Nm ≤ M_0 ≤ 250Nm



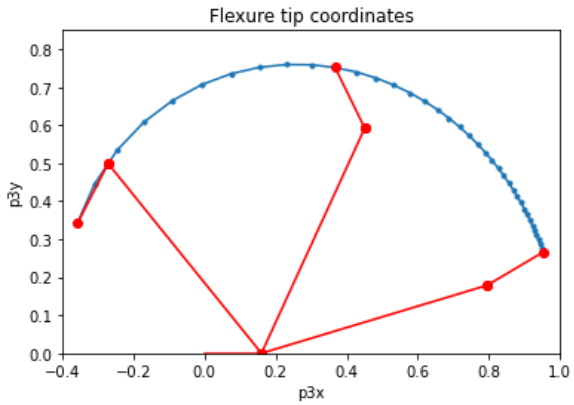
(b) Flexure behaviour 30GPa ≤ E ≤ 200GPa



(c) Flexure behaviour 0m ≤ l ≤ 2.5m



(d) Flexure behaviour 9cm ≤ w ≤ 60cm



(e) Flexure behaviour 0.74mm ≤ t ≤ 1.5mm

Figure 4.2: 2R PRBM sensitivity analysis

From the sensitivity analysis results it can be seen that for a change in flexure material, i.e. a change

in Young's modulus, the moment end load has to scale equally to reach the same deflection of the flexure. Furthermore, when the material properties allow a low moment end load, due to the material's yield strength, a large deflection can still be reached by enlarging the flexure length. Therefore, the yield strength plays a significant role for the design of flexures. Currently no measure exists which combines the flexure material's yield strength, Young's modulus, flexure length and thickness for the design of flexures.

Table 4.1: Model variable values

	Fixed value	Value range	Step size
End moment, M_0 [Nm]	100	0-250	10
Young's modulus, E [GPa]	70	25-200	5
Flexure length, l [m]	1	0-2.5	0.1
Flexure width, w [cm]	20	9-60	1
Flexure thickness, t [mm]	1	0.74-1.5	0.02

4.3. Flexure Design

For actual flexure design the sensitivity analysis alone is not enough. For a flexure design to be actually feasible, the yield strength of the flexure material needs to be considered as well. The yield strength plays a significant role for the design of flexures, because it sets a limit for the allowable stress in the flexure. When the yield strength is surpassed, the flexure will deform plastically, making the deformation permanent. To mitigate surpassing the yield strength at a certain deflection, other flexure parameters can be changed. For example, the length of the flexure can be increased, causing the required moment end load to be decreased, lowering the bending stress in the flexure.

4.3.1. Material influence

When considering the materials mentioned in table 3.2, the feasible flexure length for a desired deflection of 180° varies quit significantly. The required length for a flexure of any of those materials to reach a deflection of 180°, with a thickness of one millimeter, a width of twenty centimeters, without surpassing the yield strength, is denoted in table 4.2. The allowable moment is directly related to the material's yield strength, as can be seen through equation 4.2. The feasible deflection for a flexure of one meter is also denoted in table 4.2. The values are not exact, because they only serve an indication purpose.

Table 4.2: Feasible values of length and moment for deflection of 180°, and feasible deflection for length of 1m and moment of 100Nm, here t=1mm, w=20cm

Material	Allowable moment, M_0 [Nm] (approx.)	Required length, l [m] (approx.)	Feasible deflection, Θ [°] (approx.)
Aluminium	10	20	9.7
Rubber (Polyurethane)	1.72	0.05	3954
NiTi SMA	38	4.5	43
NiTi SMA (Superelastic)	18	4	49

$$\sigma_{max} = \frac{M_0 \frac{t}{2}}{I} \tag{4.2}$$

4.3.2. Design constant

As can be seen from table 4.2, there are enormous differences in required lengths and feasible deflections. For people who are not accustomed to working with flexure design and the influence of material properties on flexures, the results may look completely random. Actually it is possible to deduce a relation between the material properties, deflection angle, flexure length and thickness. Such a relation can be used to design flexures fast and intuitively.

First, it is necessary to distinguish the importance of the flexure material's yield strength and Young's modulus to the flexure design. Moreover, the ratio between them. This ratio can be used as an indicator for the flexibility of materials [16]. With a higher ratio of strength to Young's modulus, being better [11]. The best suited materials are both flexible and strong. To quickly identify what materials meet this criteria, an Ashby plot can be used, as seen in figure 4.3, a larger example of the figure can be found in appendix E. An Ashby plot is a scatter plot which can be used to display multiple material properties of all sort of materials. It is ideal for finding materials with a good strength to Young's modulus ratio, where the best materials are positioned at the top left.

Along the diagonal index lines, in figure 4.3, the materials have an equal strength to Young's modulus ratio. As can be seen the index lines associated with polyurethane rubber and aluminium are represented. The material properties for the NiTi SMA are added manually in the plot, represented by the two orange circles, according to the material properties from Machado et al. [15]. What can be seen from this is that the NiTi SMAs perform extremely well, compared to other alloys. Furthermore, the SMA in the martensite phase, performs slightly better on the material index, than the SMA in the austenite phase. This corresponds with the difference in required length for 180° deflection from table 4.2.

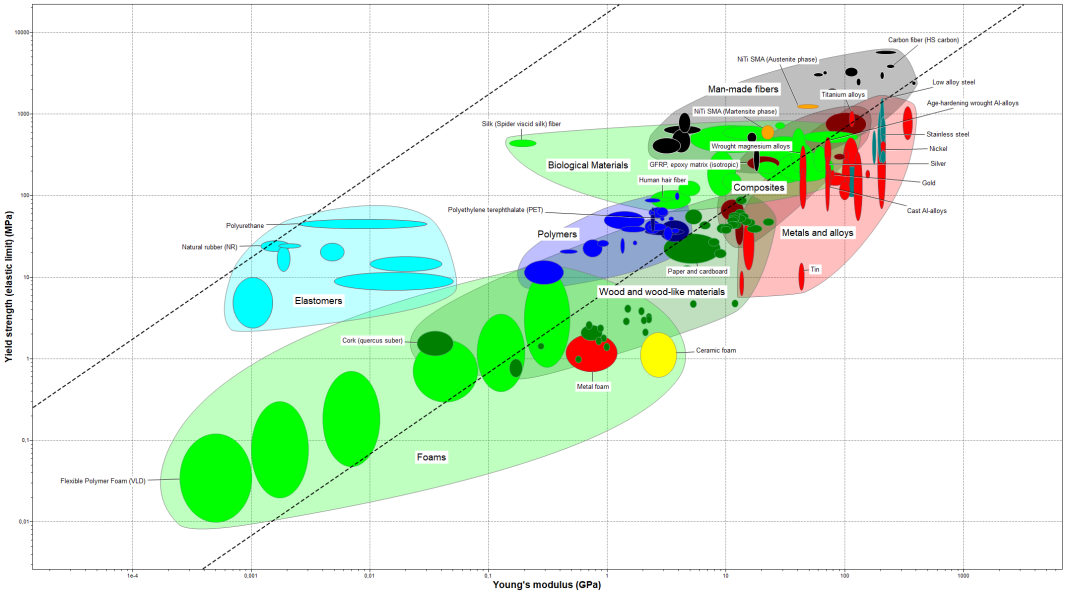


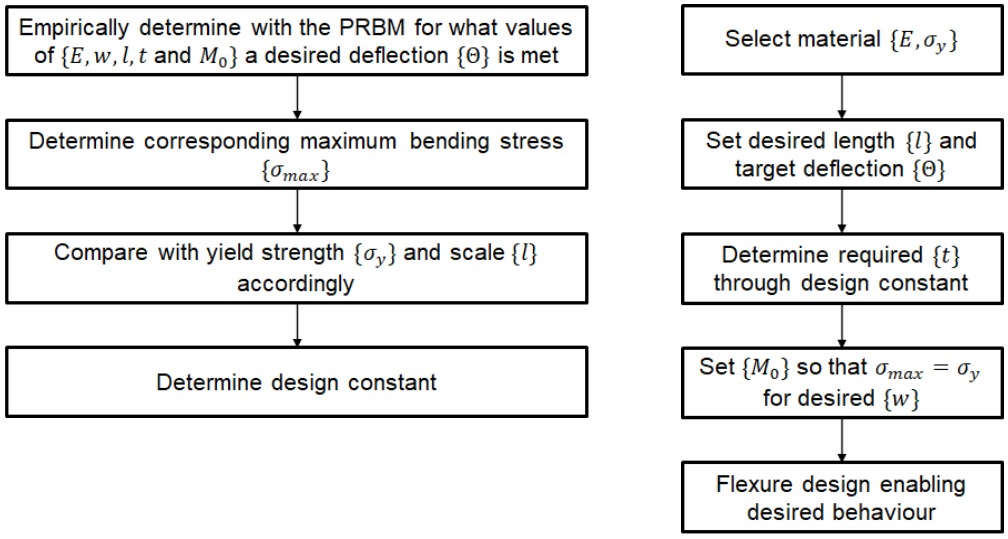
Figure 4.3: Ashby plot ratio yield strength to Young's modulus (enlarged in appendix E)

Next to the measure for flexibility, the deflection angle, flexure length and thickness are important to the flexure design. When combining equations 4.1 and 4.2 with the deflection angles acquired through the 2R PRBM, the following relation can be found:

$$\frac{F * l}{t * \Theta} = 0.438 \tag{4.3}$$

Where $F = \frac{\sigma_y}{E}$, the measure for the flexibility of the material, l, is the flexure length, t, is the flexure thickness and Θ, is the desired/feasible deflection. Here σ_y and E, stand for the flexure material's yield strength and Young's modulus respectively. The flexure width has no influence on the constant, since a larger flexure width results in a lower stress, but also a smaller deflection. The applied moment end load M_0, also has no influence on this design constant because the allowed moment is already accounted for by the value of the yield strength, assuming maximum allowable load. The flowchart depicted in figure 4.4a, shows how the design constant can be determined, regardless of the PRBM that is used.

With this constant it is possible to determine what the flexure length needs to be in order to safely facilitate a desired deflection when the material properties and flexure thickness are known. Similarly the flexure thickness and reachable deflection can be determined when the other variables are known. An example design workflow is presented in figure 4.4b, where the desired deflection and flexure length are known.



(a) Flowchart for finding the design constant

(b) Example of design process

Figure 4.4: Design flowcharts

4.4. Optimization results

The outputs from the 2R PRBM model and the design constant, serve as ideal basis for the optimization of the flexure design. With the design constant it can be determined what realistic input values are for the optimization. The inputs required for the optimization are: the target shape, the applied moment, Young's modulus, flexure length and width.

The optimization results for aluminium flexures and for SMA flexures will be shown. For materials with a constant Young's modulus the results will be the same as for the aluminium flexure, when the inputs are scaled to allow for the change in material properties, i.e. length scaled for the strength to Young's modulus ratio.

The optimization results make little sense when the target shape can not even be matched, due to a too little moment or too large thickness. The input variables are taken so that the material's yield strength is not surpassed when reaching zero shape error, furthermore, for the lowest possible thickness the target shape can be surpassed. While keeping this in mind the following inputs are used, seen in table 4.3. For the optimization the population size was set to 100, and the number of generations was set to 100 as well.

Table 4.3: Optimization variable values

	Aluminium	NiTi SMA (superelastic)	NiTi SMA
Target deflection [°]	180	180	180
End moment, M_0 [Nm]	10	18	38
Young's modulus, E [GPa]	70	25	60
Flexure length, l [m]	20	4	4.5
Flexure width, w [cm]	20	20	20
Flexure thickness range, t [mm]	1-10	1-10	1-10

4.4.1. Single flexure

The optimization results for a single flexure of aluminium can be seen in figure 4.5, the code for this optimization can be seen in appendix C. The blue dots represent the Pareto-optimal front. The other colored dots represent extreme optimal points and can be used to describe the trade-off between the multiple objectives. The significance of the extremes and the corresponding optimization results are denoted in table 4.4a. The target coordinates are from the tip of the flexure, as seen in the configuration of figure 4.1a, where the maximum deflection is 180°.

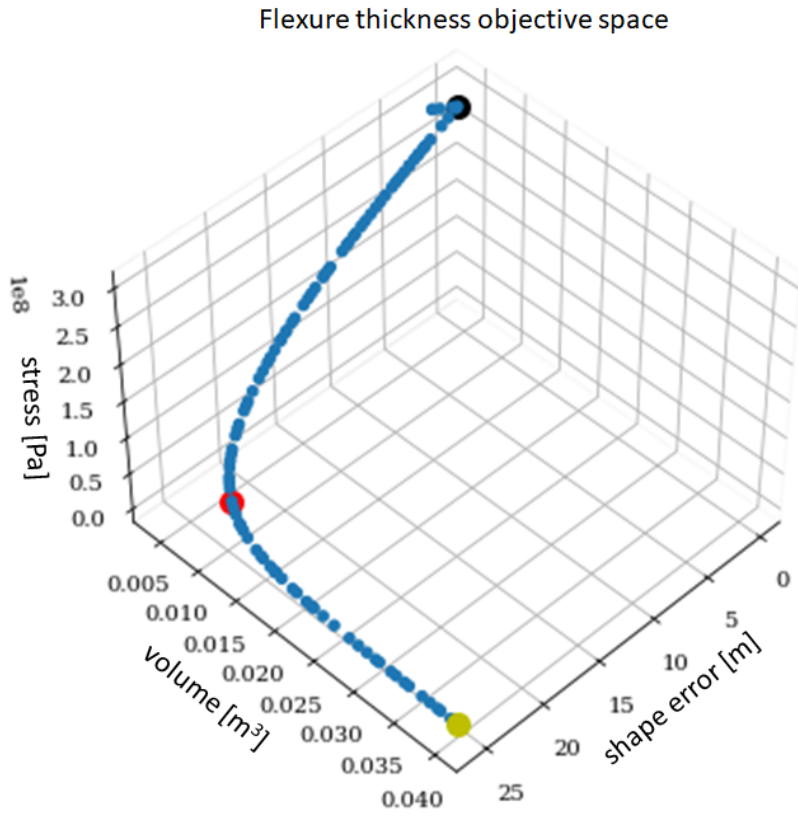


Figure 4.5: Single aluminium flexure thickness optimization

The trade-off that has to be made here is: either the shape error is zero (black solution), but the flexure is stressed to its yield limit, or the stress stays well below the yield limit (yellow solution), but the shape error is large. Both the shape error and the bending stress can not be optimal for the same configuration. The optimal solution can be in between these extreme solutions as well, since all solutions on the Pareto-front are optimal. To help in this trade-off, the volume is considered as well. The red solution represent the optimal solution between the volume and stress objectives. What solution is chosen in the final design of a flexure, depends on the needs of the user. The optimal thickness for each extreme solution is presented in table 4.4a.

The optimization results for the NiTi SMA flexure are presented in figure 4.6. The inputs to the optimization for this material are a little more complicated than for a material with constant Young's modulus, since for both material phases (normal and superelastic) the ideal length for 180° deflection is different, as seen in table 4.2. This has an influence on the target coordinates for the optimization.

When considering both the ideal lengths for the NiTi SMA when a deflection of 180° is required, some interesting results are seen in the optimization. First, when the bending stress is such, that it is between the stress boundaries for superelasticity, a jump in flexibility can be seen in the optimization, figure 4.6a. When this superelasticity can be exploited in a structure, it can be ensured that the target shape is met while maintaining low levels of stress in the flexure.

Furthermore, in figure 4.6b, the jump in flexure performance is visible as well. Here the flexure is stressed past the superelastic phase, to the martensite phase. Apart from the jump in the Pareto-front the material behaves like any other material, the material only behaves superelastically when the thickness is between 1.42013 and 1.49162 (mm). The closest the target shape can be approached, while utilizing the superelastic property is 0.56316(m). The thickness optimization values are given in tables 4.4c

Table 4.4: Optimization results

(a) Single aluminium flexure optimization results

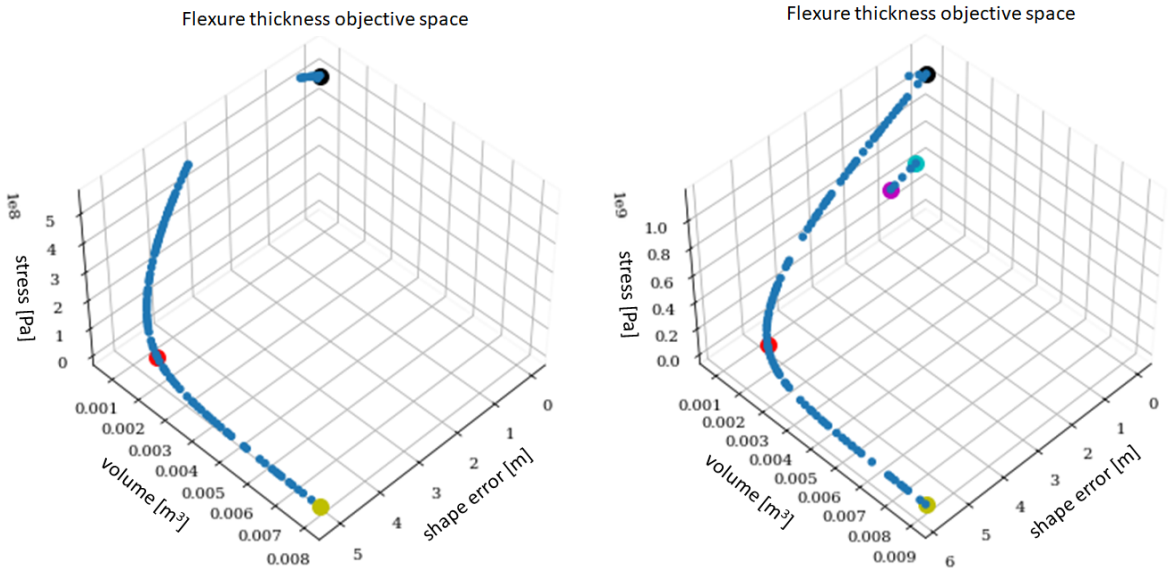
Color	Significance	Thickness [mm]
Black	Min. shape error	1.02791
Yellow	Minimum stress	10
Red	Optimal trade-off	2.52404

(b) Single superelastic SMA flexure optimization results

Color	Significance	Thickness [mm]
Black	Min. shape error	1.03055
Yellow	Minimum stress	10
Red	Optimal trade-off	2.58259

(c) Single SMA flexure optimization results

Color	Significance	Thickness [mm]
Black	Minimum shape error	1.02692
Yellow	Minimum stress	10
Red	Optimal trade-off	2.42931
Cyan	End superelastic phase	1.42013
Magenta	Start superelastic phase	1.49162



(a) SMA loaded up to superelastic phase

(b) SMA loaded passed superelastic phase

Figure 4.6: Single SMA flexure thickness optimization

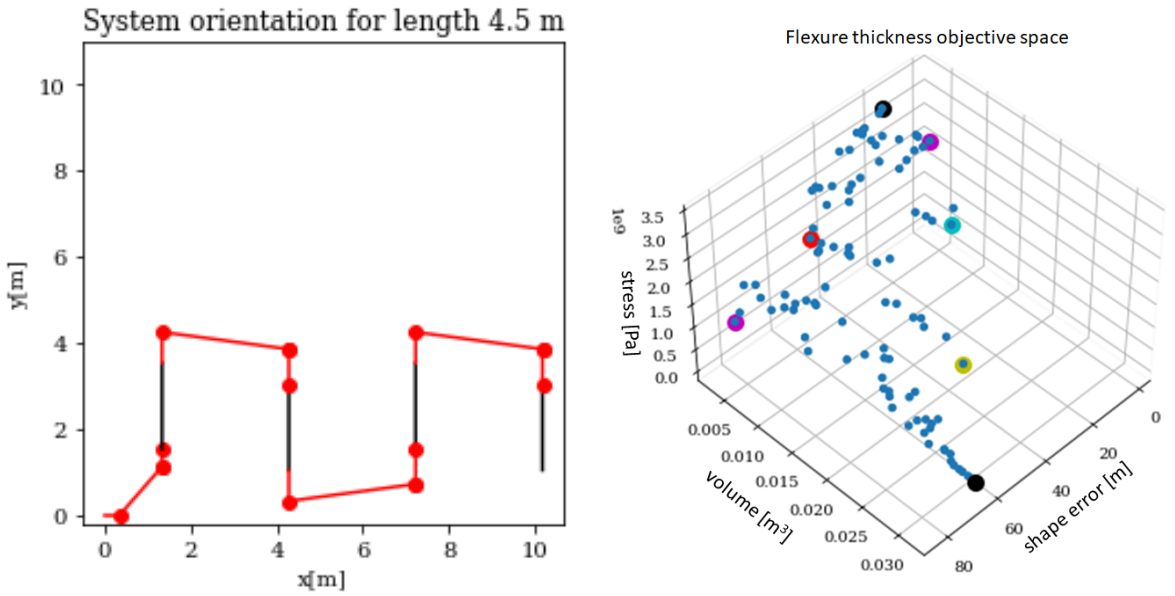
4.4.2. Connected flexures

The results are presented here below for the connected flexure use case, as seen in figure 4.1c, the code for this optimization can be seen in appendix D. The results are presented for the NiTi SMA flexures since these results show the ability of the optimization to take into account the phase change of SMAs.

As input for the optimization the values are set to allow for a 180° deflection of a SMA flexure without deforming plastically, i.e. without the yield strength being surpassed. The end moment and flexure length are set to 28 (Nm) and 4.5 (m) respectively. The end moment differs from the value given in table 4.2, because the new value allows a set of fully superelastic flexures to be the optimal solution. For this configuration the target coordinates shape is depicted in figure 4.7a. The target coordinates are taken from the start and end points of the rigid segments, which are taken to be 2 meters long.

The Pareto-optimal solution for the thicknesses is presented in figure 4.7b. The highlighted points represent solutions with a significance and are listed and evaluated in table 4.5. The numbers one to four, stand for the respective flexures, where the first flexure is the most left one. In the phase column, the letters stand for the phase the associated flexure is in, where N stands for normal (austenite/martensite) and S stands for superelastic.

From these results it can be seen that with the right boundary conditions the superelasticity proper-



(a) Target shape for flexure length of 4.5 m

(b) Multiple SMA flexures

Figure 4.7: Single SMA flexure thickness optimization

ties of the NiTi SMA can be exploited for optimal flexure design. As a matter in fact the solutions with the lowest error, being magenta and cyan, utilize the superelasticity of the martensite phase to reach the target objective. With this optimization it is possible to find solutions for the best thickness, while considering nonlinear material behaviour. That is something which is hard to achieve since multiple objectives need to be considered and the flexure behaviour is highly sensitive to material thickness. Therefore, this optimization can be used for design of complicated foldable structures, while taking into account the accuracy of the compliant mechanism's motion.

Table 4.5: Multiple SMA flexures results

Color	Significance	Thickness [mm]	Shape error [m]	Phase
Black	Minimum stress	1: 9.97033 2: 9.97184 3: 9.98609 4: 9.9852	62.6871	1: N 2: N 3: N 4: N
	Maximum stress	1: 1.00003 2: 1.00103 3: 1.0001 4: 1.00022	18.4879	1: N 2: N 3: N 4: N
Magenta	Minimum error	1: 1.23009 2: 1.23703 3: 1.25118 4: 1.26127	1.17185	1: S 2: S 3: S 4: S
	Maximum error	1: 1.94103 2: 1.28957 3: 1.69898 4: 1.61986	83.332	1: N 2: S 3: N 4: N
Red	Low volume, med. error	1: 1.73472 2: 1.21301 3: 1.23825 4: 1.3594	51.0493	1: N 2: S 3: S 4: N
Yellow	Low stress, med. error	1: 9.17153 2: 9.20635 3: 1.2847 4: 8.91481	38.7499	1: N 2: N 3: S 4: N
Cyan	Low error, med. stress	1: 1.2463 2: 1.23874 3: 1.25315 4: 9.93703	14.9009	1: S 2: S 3: S 4: N

5 Discussion

When analysing the results some remarks can be made. Firstly, the proposed target deflection of 180° brings some difficulties with it. This large deflection was chosen because it represents an extreme use case and by choosing a large deflection, a lot of use cases are covered. However, it resulted in infeasible flexure designs, when considering aluminium and rubber flexures.

For an aluminium flexure the large target deflection resulted in unrealistically long flexures. Beside the fabrication difficulties this brings, it introduces instability in the flexure when forces are not perfectly planar. The stiffness in the plane perpendicular to the deflection becomes too low, allowing for sideways deflection, which is undesirable. Furthermore, the PRBM can only describe planar deflections.

The results for the polyurethane rubber flexure were in practical as well, when the shortest feasible flexure length is assumed. The flexure could be as short as 5 (cm), which causes difficulties for large deflection. Because the flexure is only 5 (cm) long, the connected rigid members would collide when facilitating 180° deflection. Moreover, for short flexures, the 2R PRBM becomes less accurate, since extension effects weigh in more than for long flexures [20]. To model such short soft flexures accurately, a model with a prismatic joint should be considered.

Lastly on the large deflection assumption, the selected 2R PRBM becomes less accurate for high deformations. The authors of the 2R PRBM paper have identified a maximum parametric angle, for which the error between the exact tip deflection and the model output is less than 1%, to be 124.7° [21]. To accurately model deflection beyond this point, another more complex model should be selected, although even then, the results for large deflections will be limited by an allowable error.

For the material choice and description some remarks can be made as well. The choice for the polyurethane rubber type was based on it being the strongest rubber type. In hindsight it is not the best suited rubber type material for flexures. When looking at the Ashby plot, figure 4.3, it can be seen that natural rubber has a better index on the strength to Young's modulus scale. Therefore, it is able to reach the same deflection for a smaller flexure length.

When considering the SMA, it is good to evaluate the assumptions that are made. The material behaviour model has been taken from Auricchio et al. [1]. This model assumes an isotropic material which is subject to an uniaxial tensile stress, therefore, only two phases exist, the austenite and the martensite phase. Furthermore, temperature differences have an influence on the SMA properties, however, in this research, temperature changes are not considered.

The unloading of the flexures are also not considered. Along with the hysteresis and damping effects of SMAs. These elements are outside the scope of this research and can be considered in future research.

To evaluate on the found design constant, equation 4.3, it has to be considered that the constant value is derived from results gained with the 2R PRBM. When another model is used to find the design constant, the value can differ. Furthermore, the constant is derived from configurations with a large deflection. As previously mentioned the model accuracy decreases at large deflections, therefore the inaccuracy is carried over to the design constant. That leads us to conclude that the design constant can only be used as an initial indication for the dimensions of a feasible flexure or to get a feel for the consequences of varying certain flexure parameters.

Furthermore, the evaluated flexure configurations are assumed to be loaded up to their yield strength.

In theory this is a save limit because no plastic deformations should show up. However, when actuated frequently beyond 20% of the yield strength, micro slip can occur [9]. This can influence the accuracy and safety of the flexure. More research is required to investigate the exact consequences of micro slip. Moreover, fatigue is a danger to flexure operation, even when staying below the yield strength of materials, because flexures can have imperfections where fatigue failure can propagate.

The PRBM used for this research is based on the assumption that the flexure is only loaded with a pure moment end load. In reality this is hard to achieve, since there will always be axial loads present. These axial loads influence the accuracy of the model. For loads other than pure moment loads, different models exist. These models have a smaller accuracy range however, since the load cases are more complex [21].

For the system configuration, where multiple flexures are connected by rigid links, it is assumed that the moment end load on each flexure is equal. For practical applications this does not have to be the case. When different loads are considered the structure can have multiple functionalities, next to the extending and retracting functionality.

The rigid members in the system configuration are taken to be two meters long. This has influence on the results because an inaccuracy of the flexure tip angle, gets exaggerated by the connected rigid member, since the position of the end of the rigid member is determined by the angle multiplied by the member length. When the rigid members are short the model is more forgiving, since inaccuracies are not enlarged. This dependency of the model accuracy should be considered when evaluating a configuration, especially when the structure is to be used in high accuracy applications.

The optimization considered in this research can only evaluate the optimal flexure thickness. It would be ideal if flexure length could be evaluated as well. However, this is extremely difficult with a target shape optimization because the target coordinates are dependant on the length of the flexure. When the length would be integrated, the optimization would become too complex. When it would be possible to integrate the length in the optimization, it is possible to design flexure mechanisms only with a given desired deflection, making the optimization more flexible and versatile.

For the connected flexures optimization only the case of the SMA is considered. Materials with a constants Young's modulus behave similarly in the optimization, except for the outlier values seen at the SMA results. The results for aluminium show a more compact cloud, while having the same trend as the SMA results.

Finally it should be noted that this PRBM only assumes beam like, straight flexures, with a constant cross-section. In reality the sharp edges at the start and end point of the flexure, can introduce high stress concentrations. For real world applications, corner-filleted flexures should be considered. Corner-filleted flexures are more bending compliant and induce lower stresses, however they are harder to model and are less accurate in rotation [14].

6

Conclusions

This research optimizes the design for foldable structures by introducing a design constant which relates flexure material's yield strength, Young's modulus, flexure length and thickness with the desired deflection, equation 6.1. Thereby making the design of flexures for certain use case more intuitive. With the design constant it is immediately visible what feasible flexure designs are, and how parameter changes influence the other flexure parameters. Moreover, the design constant accounts for the flexure material's yield strength, which only leads to feasible and safe flexure designs. This offers an ideal basis for efficient flexure design and can aid researchers in early design stages when trialing multiple starting configurations when designing foldable compliant structures.

$$\frac{F * l}{t * \Theta} = 0.438 \tag{6.1}$$

Where $F = \frac{\sigma_y}{E}$, the measure for the flexibility of the material, l is the flexure length, t is the flexure thickness and Θ is the desired/feasible deflection. Here σ_y and E stand for the flexure material's yield strength and Young's modulus respectively.

The results from the model and optimization cover a large range of application possibilities because an extreme target shape is evaluated and materials with superelastic properties are used. It is shown how these superelastic properties can be exploited to facilitate large deflections while stresses are low. Moreover, an Ashby plot is presented, by which suitable flexure materials can quickly be identified. The measure for material flexibility is presented, which plays a big role in flexure design. The design constant can be used to directly identify the influence of the material choice to the flexure behaviour.

Furthermore, with an optimization for a foldable structure design, it is shown how an optimization can be used to utilize certain flexure material properties and flexure thickness to facilitate a desired configuration. The trade-off between safe stresses and large deflections is discussed for the optimization, where the optimal solution depends on the needs of the user. With a foldable structure configuration it is shown how an optimization can be used for the design of flexures for foldable structures, while taking into account the accuracy of the compliant mechanism's motion.

For future research it is interesting to visualize the behaviour of a flexure with a different cross-section than a rectangular beam-like cross-section, for example a rod configuration. For a rod the moment of inertia is $I = \frac{\pi d^4}{64}$, which influences the deflection and the stresses. Furthermore, more research is required into the practicality of flexures for large scale applications. When not designed properly, the out of plane stiffness of large scale flexures can become too low, risking the planar motion of flexures. The unloading of SMAs is not considered in this research while it could have a significant effect on flexure behaviour in real world applications, therefore, more research is required into the unloading effects. Future research should be focused on the influence of micro slip for large scale flexure applications as well.

References

[1] Ferdinando Auricchio, Robert L Taylor, and Jacob Lubliner. Shape-memory alloys: macromodelling and numerical simulations of the superelastic behavior. *Computer methods in applied mechanics and engineering*, 146(3-4):281–312, 1997.

[2] Yves Bellouard and Reymond Clavel. Shape memory alloy flexures. *Materials Science and Engineering: A*, 378(1-2):210–215, 2004.

[3] Freek GJ Broeren, Volkert van der Wijk, and Just L Herder. Spatial pseudo-rigid body model for the analysis of a tubular mechanical metamaterial. *Mathematics and Mechanics of Solids*, 25(2): 305–316, 2020.

[4] BYU CMR. *Compliant Mechanisms Explained*. `https://www.compliantmechanisms.byu.edu/about-compliant-mechanisms`, 2021. Accessed: 2021-10-22.

[5] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[6] Delft University of Technology. *Compliant Mechanisms, Functional flexibility*. `https://www.tudelft.nl/3me/over/afdelingen/precision-and-microsystems-engineering-pme/research/structural-optimization-and-mechanics-som/som-research/compliant-mechanisms`, 2021. Accessed: 2021-10-22.

[7] The Efficient Engineer. *Understanding the Finite Element Method* [Video]. Youtube. `https://www.youtube.com/watch?v=GHjopp47vvQ&ab_channel=TheEfficientEngineer`, 2021.

[8] Granta. *EduPack (2021 R2)*. [Computer software].

[9] Jonathan B Hopkins. *Compliant Mechanisms Lecture 1 Part 4* [Video]. Youtube. `https://www.youtube.com/watch?v=FJd3LLe3NE0&ab_channel=TheFACTsofMechanicalDesign`, 2021.

[10] Larry L Howell, Ashok Midha, and Tony W Norton. *Evaluation of equivalent spring stiffness for use in a pseudo-rigid-body model of large-deflection compliant mechanisms*. 1996.

[11] Larry L Howell, Spencer P Magleby, and Brian M Olsen. *Handbook of Compliant Mechanisms*. John Wiley & Sons, 2013.

[12] Sangyoon Lee and Ilkyeong Moon. Robust empty container repositioning considering foldable containers. *European Journal of Operational Research*, 280(3):909–925, 2020.

[13] Nicolae Lobontiu. *Compliant mechanisms: design of flexure hinges*. CRC press, 2002.

[14] Nicolae Lobontiu, Jeffrey SN Paine, Ephrahim Garcia, and Michael Goldfarb. Corner-filleted flexure hinges. *J. Mech. Des.*, 123(3):346–352, 2001.

[15] Guilherme Machado, Hervé Louche, Thierry Alonso, and Denis Favier. Superelastic cellular niti tube-based materials: Fabrication, experiments and modeling. *Materials & Design (1980-2015)*, 65:212–220, 2015.

[16] Jun Peng and G Jeffrey Snyder. A figure of merit for flexibility. *Science*, 366(6466):690–691, 2019.

[17] Kazuya Saito, Akira Tsukahara, and Yoji Okabe. New deployable structures based on an elastic origami model. *Journal of mechanical design*, 137(2):021402, 2015.

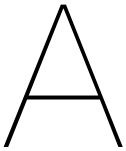
[18] Venkatasubramanian Kalpathy Venkiteswaran and Hai-Jun Su. Effect of beam geometry on the accuracy of pseudo-rigid-body models. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 57120, page V05AT08A020. American Society of Mechanical Engineers, 2015.

[19] Venkatasubramanian Kalpathy Venkiteswaran and Hai-Jun Su. A parameter optimization framework for determining the pseudo-rigid-body model of cantilever-beams. *Precision Engineering*, 40:46–54, 2015.

[20] Venkatasubramanian Kalpathy Venkiteswaran and Hai-Jun Su. Extension effects in compliant joints and pseudo-rigid-body models. *Journal of Mechanical Design*, 138(9):092302, 2016.

[21] Yue-Qing Yu, Zhong-Lei Feng, and Qi-Ping Xu. A pseudo-rigid-body 2r model of flexural beam in compliant mechanisms. *Mechanism and Machine Theory*, 55:18–33, 2012.

[22] Yue-Qing Yu, Qian Li, and Qi-Ping Xu. Pseudo-rigid-body dynamic modeling and analysis of compliant mechanisms. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 232(9):1665–1678, 2018.



Single flexure 2R PRBM Python code

```python
"""
@author: Jurian van Dijk
"""
# 2R Pseudo-Rigid-Body Model
from math import cos, sin, radians
import numpy as np
import matplotlib.pyplot as plt

# Model parameters for specific load case (Moment load)
gamma1 = 0.66       # characteristic radius factor
gamma2 = 0.18
gamma0 = 0.16

ktheta1 = 2.0571    # spring stiffness coefficient, nondimensionalized torsion spring constant
ktheta2 = 1.9175

# Specify external moment load on tip
m0 = 100     # [Nm]

# Specify flexure properties
E = 70e9     # young's modulus [Pa]
l = 1         # length [m]
w = 0.2       # width [m]
h = 0.001     # height [m]

# Create empty lists for storage of data
angles = []
tipcoordinates = []

# Calculate tip coordinates for each value of the variable
for m0 in np.arange(0, 260, 10):   # h:(0.00074, 0.00152, 0.00002), m0:(0, 260, 10),
                                   # E:(25e9, 205e9, 5e9)
    # Moment of inertia of beam with rectangular cross section
    I = (w * h ** 3) / 12
    sigma = ((m0 * (h / 2)) / I) / 1e6 # max bending stress [MPa]

# ============================================================================
#     if  500e6 < (m0 * (h / 2)) / I < 580e6:
#         E = 25e9
#     else:
#         E = 60e9
# ============================================================================

```

```
    # Determine spring constants
    k1 = gamma1 * ktheta1 * ((E * I) / l)
    k2 = gamma2 * ktheta2 * ((E * I) / l)

    # Determine resulting angles with torque
    theta1 = m0 * (1 - gamma0) / k1
    theta2 = m0 * gamma2 / k2
    theta = theta1 + theta2

    # Resulting tip coordinates
    p3x =  l * (gamma0 + gamma1 * cos(radians(theta1)) + gamma2 * cos(radians(theta)))
    p3y =  l * (gamma1 * sin(radians(theta1)) + gamma2 * sin(radians(theta)))

    # Create lists to store data
    angles.append(theta1)
    tipcoordinates.append((p3x, p3y))
    print('t:', theta, 't1:', theta1, 't2:', theta2, 'x:', p3x, 'y:', p3y, 'sigma:', sigma)

# Determine orientation last iteration rigid flexure link 1
p2x = l * (gamma0 + (cos(radians(theta1)) * gamma1))
p2y = l * (sin(radians(theta1)) * gamma1)

# Determine orientation first iteration rigid flexure link 1
p21x = l * (gamma0 + (cos(radians(angles[0])) * gamma1))
p21y = l * (sin(radians(angles[0])) * gamma1)

# Determine orientation middle iteration rigid flexure link 1
c = 10 # Number of iteration
p22x = l * (gamma0 + (cos(radians(angles[c])) * gamma1))
p22y = l * (sin(radians(angles[c])) * gamma1)

# Make arrays to plot x and y coordinates of the tip
x = np.array([x[0] for x in tipcoordinates])
y = np.array([x[1] for x in tipcoordinates])

# Plot tip coordinates
plt.plot(x, y, '.-')

# Plot flexure orientation last iteration
plt.plot([0, l * gamma0],[0,0], '-r')
plt.plot([l * gamma0, p2x],[0, p2y], 'o-r')
plt.plot([p2x, p3x],[p2y, p3y], 'o-r')

# Plot flexure orientation first iteration
plt.plot([l * gamma0, p21x],[0, p21y], 'o-r')
plt.plot([p21x, tipcoordinates[0][0]],[p21y, tipcoordinates[0][1]], 'o-r')

# Plot flexure orientation middle iteration
plt.plot([l * gamma0, p22x],[0, p22y], 'o-r')
plt.plot([p22x, tipcoordinates[c][0]],[p22y, tipcoordinates[c][1]], 'o-r')

plt.axis([-0.4, 1, 0, 0.85])
plt.title('Flexure tip coordinates')
plt.xlabel('p3x')
plt.ylabel('p3y')
plt.show()
```

B

Connected flexure 2R PRBM Python code

```
"""
@author: Jurian van Dijk
"""

# Pseudo-Rigid-Body Model with two rotary joints
from math import  cos, sin, radians
#import numpy as np
import matplotlib.pyplot as plt

# Model parameters for specific load case (Moment load), same moment at every hinge
gamma1 = 0.66       # characteristic radius factor
gamma2 = 0.18
gamma0 = 0.16

ktheta1 = 2.0571    # spring stiffness coefficient, nondimensionalized torsion spring constant
ktheta2 = 1.9175

# Specify external moment load on tip
m0 = 40.93     # [Nm]

# Specify flexure properties
E = 70e9     # young's modulus [Pa]
l1 = 2.25      # length flexure1[m]
l3 = l1*2          # length other flexures
w = 0.2       # width [m]
h = 0.001       # height [m]

# Rigid beam properties
l2 = 2      # length [m]

# Calculate FLEXURE 1 tip coordinates for each value of the variable

# Moment of inertia of beam with rectangular cross section
I = (w * h ** 3) / 12

# Determine spring constant flexure1
k1 = gamma1 * ktheta1 * ((E * I) / l1)
k2 = gamma2 * ktheta2 * ((E * I) / l1)

# Determine spring constant other flexures
k3 = gamma1 * ktheta1 * ((E * I) / l3)
```

```
k4 = gamma2 * ktheta2 * ((E * I) / l3)

# Determine resulting angles with torque
theta1 = (m0 * (1 - gamma0) / k1)
theta2 = (m0 * gamma2 / k2)
theta = theta1 + theta2

theta0 = (m0 * l3) / (E * I)

# Resulting tip coordinates flexure1
p3x = l1*(gamma0 + gamma1 * cos(radians(theta1)) + gamma2 * cos(radians(theta)))
p3y = l1*(gamma1 * sin(radians(theta1)) + gamma2 * sin(radians(theta)))

print('t:', theta, 't0:', theta0, 't1:', theta1, 't2:', theta2, 'x/L:', p3x, 'y/L:', p3y)

# Coordinate start point flexure 2
sx = p3x + l2 * cos(radians(theta))
sy = p3y + l2 * sin(radians(theta))

# Calculate FLEXURE2 tip coordinates
# Determine resulting LOCAL angles
theta4 = -m0 * (1 - gamma0) / k3     # theta1 2nd flexure
theta5 = -m0 * gamma2 / k4           # theta2 2nd flexure
theta3 = theta4 + theta5

# Determine GLOBAL angles
THETA1 = theta4 + theta
THETA2 = theta5 + theta
THETA = theta3 + theta

# Resulting global tip coordinates
s3x = sx + l3*(gamma0*cos(radians(theta)) + gamma1*cos(radians(THETA1)) + gamma2*cos(radians(THETA)))
s3y = sy + l3*(gamma0*sin(radians(theta)) + gamma1*sin(radians(THETA1)) + gamma2*sin(radians(THETA)))

# Determine start point flexure 3
tx = s3x + l2 * cos(radians(THETA))
ty = s3y + l2 * sin(radians(THETA))

# Calculate FLEXURE3 tip coordinates
# Global angles flexure 3
THETA13 = -theta4 + THETA
THETA23 = -theta5 + THETA
THETA03 = -theta3 + THETA

# Resulting global tip coordinates
t3x = tx + l3*(gamma0*cos(radians(THETA)) + gamma1*cos(radians(THETA13)) + gamma2*cos(radians(THETA03)))
t3y = ty + l3*(gamma0*sin(radians(THETA)) + gamma1*sin(radians(THETA13)) + gamma2*sin(radians(THETA03)))

# Determine start point flexure 4
fx = t3x + l2 * cos(radians(theta))
fy = t3y + l2 * sin(radians(theta))

# Calculate FLEXURE4 tip coordinates
# Resulting global tip coordinates
f3x = fx + l3*(gamma0*cos(radians(theta)) + gamma1*cos(radians(THETA1)) + gamma2*cos(radians(THETA)))
f3y = fy + l3*(gamma0*sin(radians(theta)) + gamma1*sin(radians(THETA1)) + gamma2*sin(radians(THETA)))

# Determine end point
```

```
ex = f3x + l2 * cos(radians(THETA))
ey = f3y + l2 * sin(radians(THETA))


print('sx',sx,'sy',sy,'s3x',s3x,'s3y',s3y,'tx',tx,'ty',ty,'t3x',t3x,'t3y',t3y
        ,'fx',fx,'fy',fy,'f3x',f3x,'f3y',f3y,'ex',ex,'ey',ey)
# Plot flexure1 last iteration
p2x = l1*(gamma0 + (cos(radians(theta1)) * gamma1))
p2y = l1*(sin(radians(theta1)) * gamma1)
plt.plot([0,l1*gamma0],[0,0], '-r')
plt.plot([l1*gamma0, p2x],[0, p2y], 'o-r')
plt.plot([p2x, p3x],[p2y, p3y], 'o-r')

# Plot rigid beam
plt.plot([p3x, sx],[p3y, sy], '-k')
plt.plot([s3x,tx],[s3y,ty], '-k')
plt.plot([t3x,fx],[t3y,fy], '-k')
plt.plot([f3x,ex],[f3y,ey], '-k')

# Plot flexure2 last iteration
s0x = sx + l3*(gamma0 * cos(radians(theta)))
s0y = sy + l3*(gamma0 * sin(radians(theta)))
s21x = s0x + l3*(gamma1 * cos(radians(THETA1)))
s21y = s0y + l3*(gamma1 * sin(radians(THETA1)))
plt.plot([sx,s0x],[sy,s0y], '-r')
plt.plot([s0x, s21x],[s0y, s21y], 'o-r')
plt.plot([s21x, s3x],[s21y, s3y], 'o-r')

# Plot flexure3 last iteration
t0x = tx + l3*(gamma0 * cos(radians(THETA)))
t0y = ty + l3*(gamma0 * sin(radians(THETA)))
t21x = t0x + l3*(gamma1 * cos(radians(THETA13)))
t21y = t0y + l3*(gamma1 * sin(radians(THETA13)))
plt.plot([tx,t0x],[ty,t0y], '-r')
plt.plot([t0x, t21x],[t0y, t21y], 'o-r')
plt.plot([t21x, t3x],[t21y, t3y], 'o-r')

# Plot flexure4 last iteration
f0x = fx + l3*(gamma0 * cos(radians(theta)))
f0y = fy + l3*(gamma0 * sin(radians(theta)))
f21x = f0x + l3*(gamma1 * cos(radians(THETA1)))
f21y = f0y + l3*(gamma1 * sin(radians(THETA1)))
plt.plot([fx,f0x],[fy,f0y], '-r')
plt.plot([f0x, f21x],[f0y, f21y], 'o-r')
plt.plot([f21x, f3x],[f21y, f3y], 'o-r')

plt.axis('square')
plt.title('System orientation for length %1.1f m' %l3)
plt.xlabel('x[m]')
plt.ylabel('y[m]')
plt.show()
```

C

Single flexure thickness NSGAII Optimization Python code

```
"""
@author: Jurian van Dijk
"""

# Pseudo-Rigid-Body Model with two rotary joints
from math import cos, sin, radians, dist
import numpy as np
#import matplotlib.pyplot as plt

from pymoo.optimize import minimize
from pymoo.model.problem import Problem
from pymoo.algorithms.nsga2 import NSGA2
from pymoo.visualization.scatter import Scatter

# Model parameters for moment load case
gamma1 = 0.66       # characteristic radius factor
gamma2 = 0.18
gamma0 = 0.16

ktheta1 = 2.0571    # spring stiffness coefficient, nondimensionalized torsion spring constant
ktheta2 = 1.9175

# Specify external moment load on tip
m0 = 38    # [Nm]

# Specify flexure properties
l = 4.5        # length [m]
w = 0.2       # width [m]

E = 70e9     # young's modulus [Pa]

# Target tip coordinates for 180 degree rotation of model
P3x =  -0.4871090400272526 # Target x-coordinate
P3y =  2.943004286350274 # Target y-coordinate

# Optimization, elementwise
class MyProblem(Problem):

    def __init__(self):
        super().__init__(n_var=1,              # h is variable
                         n_obj=3,              # location, volume, stress
```

```
                        n_constr=0,             # Stress limit
                        xl=np.array([0.001]),   # lower boundary of variable
                        xu=np.array([0.01]),    # upper boundary of variable
                        elementwise_evaluation=True)

    def _evaluate(self, h, out, *args, **kwargs):
        I = (w * h ** 3) / 12

        if  500e6 < (m0 * (h / 2)) / I < 580e6:
            E = 25e9
        else:
            E = 60e9

        k1 = gamma1 * ktheta1 * ((E * I) / l)
        k2 = gamma2 * ktheta2 * ((E * I) / l)
        theta1 = m0 * (1 - gamma0) / k1
        theta2 = m0 * gamma2 / k2
        theta = theta1 + theta2
        p3x = l * (gamma0 + gamma1 * cos(radians(theta1)) + gamma2 * cos(radians(theta)))
        p3y = l * (gamma1 * sin(radians(theta1)) + gamma2 * sin(radians(theta)))

        f1 = dist([P3x, P3y], [p3x, p3y])
        f2 = w * h * l
        f3 = (m0 * (h/ 2)) / I

        g1 = ((m0 * (h/ 2)) / I) - 580e6

        out["F"] = np.column_stack([f1, f2, f3])
        out["G"] = np.array(g1)

problem = MyProblem()

algorithm = NSGA2(pop_size=100)

stop_criteria = ('n_gen', 100)

results = minimize(
    problem = problem,
    algorithm = algorithm,
    termination = stop_criteria,
    seed=1,
    save_history=True,
    verbose=True)       # False/True to hide/show algorithm results

# Plotting
# Objective space
plot = Scatter(title = "Objective space")
plot.add(results.F)
#plot.add(results.F[:,[0,2]])     # 0=location, 1=volume, 2=stress
plot.add(results.F[55,:], s=100, color='m')
plot.add(results.F[92,:], s=100, color='c')
plot.add(results.F[1,:], s=100, color='y')
plot.add(results.F[0,:], s=100, color='k')
plot.add(results.F[19,:], s=100, color='r')

plot.show()
```

D

Connected flexure thickness NSGAII Optimization Python code

```
"""
@author: Jurian van Dijk
"""

# Pseudo-Rigid-Body Model with two rotary joints
from math import cos, sin, radians, dist
import numpy as np
#import matplotlib.pyplot as plt

from pymoo.optimize import minimize
from pymoo.model.problem import Problem
from pymoo.algorithms.nsga2 import NSGA2
from pymoo.visualization.scatter import Scatter

# Model parameters for moment load case
gamma1 = 0.66       # characteristic radius factor
gamma2 = 0.18
gamma0 = 0.16

ktheta1 = 2.0571    # spring stiffness coefficient, nondimensionalized torsion spring constant
ktheta2 = 1.9175

# Specify external moment load on tip
m0 = 28     # [Nm]

# Specify flexure properties
l1 = 2.25        # length first flexure [m]
l2 = 2           # length rigid links
l3 = l1*2        # length other flexures [m]
w = 0.2       # width [m]

#E = 70e9     # young's modulus [Pa]

# Target tip coordinates for 90 degree rotation of first flexure
P3x = 1.3373833630856669    # Target x-coordinate
P3y = 1.522983763681005     # Target y-coordinate

# Target coordinates for 180 degree flexure rotation
Sx = 1.337234155736915  # Start second flexure
Sy =  3.5229837581152967

```

```
S3x =  4.280536834904316  # Tip second flexure
S3y = 3.0365024759304324

Tx = 4.280387627555564 # Start third flexure
Ty = 1.0365024814961408

T3x = 7.223690306722965 # Tip third flexure
T3y = 1.5229837636810053

Fx = 7.223541099374213 # Start fourth flexure
Fy = 3.5229837581152967

F3x =  10.166843778541613 # Tip fourth
F3y =  3.0365024759304324

Ex =  10.166694571192862  # End point
Ey =   1.0365024814961408

# Optimization, elementwise
class MyProblem(Problem):

    def __init__(self):
        super().__init__(n_var=4,                # h is variable
                         n_obj=3,                # location, volume, stress
                         n_constr=0,             # Stress limit
                         xl=np.array([0.001, 0.001, 0.001, 0.001]),  # lower boundary of variable
                         xu=np.array([0.01, 0.01, 0.01, 0.01]),    # upper boundary of variable
                         elementwise_evaluation=True)

    def _evaluate(self, h, out, *args, **kwargs):
        # Values for first flexure
        I = (w * h[0] ** 3) / 12

        if  500e6 < (m0 * (h[0] / 2)) / I < 580e6:
            E = 25e9
        else:
            E = 60e9

        k1 = gamma1 * ktheta1 * ((E * I) / l1)
        k2 = gamma2 * ktheta2 * ((E * I) / l1)
        theta1 = m0 * (1 - gamma0) / k1
        theta2 = m0 * gamma2 / k2
        theta = theta1 + theta2

        # Actual first flexure tip
        p3x = l1 * (gamma0 + gamma1 * cos(radians(theta1)) + gamma2 * cos(radians(theta)))
        p3y = l1 * (gamma1 * sin(radians(theta1)) + gamma2 * sin(radians(theta)))

        # Coordinate start point flexure 2
        sx = p3x + l2 * cos(radians(theta))
        sy = p3y + l2 * sin(radians(theta))

        # Calculate FLEXURE2 tip coordinates
        # Determine spring constant other flexures
        I2 = (w * h[1] ** 3) / 12

        if  500e6 < (m0 * (h[1] / 2)) / I2 < 580e6:
            E = 25e9
```

```
        else:
            E = 60e9

        k3 = gamma1 * ktheta1 * ((E * I2) / l3)
        k4 = gamma2 * ktheta2 * ((E * I2) / l3)
        # Determine resulting LOCAL angles
        theta4 = -m0 * (1 - gamma0) / k3     # theta1 2nd flexure
        theta5 = -m0 * gamma2 / k4           # theta2 2nd flexure
        theta3 = theta4 + theta5

        # Determine GLOBAL angles flexure 2
        THETA1 = theta4 + theta
        THETA = theta3 + theta

        # Resulting global tip coordinates flexure2
        s3x = sx + l3*(gamma0*cos(radians(theta)) + gamma1*cos(radians(THETA1)) + gamma2*cos(radians(THETA)))
        s3y = sy + l3*(gamma0*sin(radians(theta)) + gamma1*sin(radians(THETA1)) + gamma2*sin(radians(THETA)))

        # Determine start point flexure 3
        tx = s3x + l2 * cos(radians(THETA))
        ty = s3y + l2 * sin(radians(THETA))

        # Calculate FLEXURE3 tip coordinates
        # Determine spring constant other flexures
        I3 = (w * h[2] ** 3) / 12

        if  500e6 < (m0 * (h[2] / 2)) / I3 < 580e6:
            E = 25e9
        else:
            E = 60e9

        k5 = gamma1 * ktheta1 * ((E * I3) / l3)
        k6 = gamma2 * ktheta2 * ((E * I3) / l3)
        # Determine resulting LOCAL angles
        theta7 = m0 * (1 - gamma0) / k5     # theta1 3rd flexure
        theta8 = m0 * gamma2 / k6           # theta2 3rd flexure
        theta6 = theta7 + theta8

        # Global angles flexure 3
        THETA13 = theta7 + THETA
        THETA03 = theta6 + THETA

        # Resulting global tip coordinates flexure3
        t3x = tx + l3*(gamma0*cos(radians(THETA)) + gamma1*cos(radians(THETA13)) + gamma2*cos(radians(THETA03)))
        t3y = ty + l3*(gamma0*sin(radians(THETA)) + gamma1*sin(radians(THETA13)) + gamma2*sin(radians(THETA03)))

        # Determine start point flexure 4
        fx = t3x + l2 * cos(radians(THETA03))
        fy = t3y + l2 * sin(radians(THETA03))

        # Calculate FLEXURE4 tip coordinates
        # Determine spring constant other flexures
        I4 = (w * h[3] ** 3) / 12

        if  500e6 < (m0 * (h[3] / 2)) / I4 < 580e6:
            E = 25e9
        else:
            E = 60e9
```

```
158
159         k7 = gamma1 * ktheta1 * ((E * I4) / l3)
160         k8 = gamma2 * ktheta2 * ((E * I4) / l3)
161         # Determine resulting LOCAL angles
162         theta10 = -m0 * (1 - gamma0) / k7     # theta1 4th flexure
163         theta11 = -m0 * gamma2 / k8           # theta2 4th flexure
164         theta9 = theta10 + theta11
165
166         # Global angles flexure 4
167         THETA14 = theta10 + THETA03
168         THETA04 = theta9 + THETA03
169
170         # Resulting global tip coordinates flexure4
171         f3x = fx+l3*(gamma0*cos(radians(THETA03))+gamma1*cos(radians(THETA14))+gamma2*cos(radians(THETA04
172         f3y = fy+l3*(gamma0*sin(radians(THETA03))+gamma1*sin(radians(THETA14))+gamma2*sin(radians(THETA04
173
174         # Determine end point
175         ex = f3x + l2 * cos(radians(THETA04))
176         ey = f3y + l2 * sin(radians(THETA04))
177
178
179         # Objectives
180         # Minimize sum of distances between target coordinates and model coordinates
181         f1 = dist([P3x, P3y], [p3x, p3y]) + dist([Sx, Sy], [sx, sy]) + dist([S3x, S3y], [s3x, s3y])
182         + dist([Tx, Ty], [tx, ty]) + dist([T3x, T3y], [t3x, t3y]) + dist([Fx, Fy], [fx, fy])
183         + dist([F3x, F3y], [f3x, f3y]) + dist([Ex, Ey], [ex, ey])
184
185         # Minimize sum of volumes of flexures
186         f2 = (w * h[0] * l1) + (w * h[1] * l3) + (w * h[2] * l3) + (w * h[3] * l3)
187
188         # Minimize sum of maximum stresses in flexures
189         f3= ((m0*(h[0]/ 2)) / I) + ((m0*(h[1]/ 2)) / I2) + ((m0*(h[2]/ 2)) / I3) + ((m0*(h[3]/ 2)) / I4)
190
191         # Stress in each segment
192         g1 = ((m0 * (h[0] / 2)) / I)
193         g2 = ((m0 * (h[1] / 2)) / I2)
194         g3 = ((m0 * (h[2] / 2)) / I3)
195         g4 = ((m0 * (h[3] / 2)) / I4)
196
197
198         out[”F”] = np.column_stack([f1, f2, f3])
199         out[”G”] = np.column_stack([g1, g2, g3, g4])
200
201 problem = MyProblem()
202
203 algorithm = NSGA2(pop_size=100)
204 stop_criteria = ('n_gen', 100)
205
206 results = minimize(
207     problem = problem,
208     algorithm = algorithm,
209     termination = stop_criteria,
210     seed=1,
211     save_history=True,
212     verbose=True)       # False/True to hide/show algorithm results
213
214
215 # Plotting
216 # Objective space
```

```
plot = Scatter(title = ”Objective space”)
plot.add(results.F)
#plot.add(results.F[:,[0,1]])
plot.add(results.F[1,:], s=100, color='k')
plot.add(results.F[2,:], s=100, color='k')
plot.add(results.F[13,:], s=100, color='y')
plot.add(results.F[3,:], s=100, color='m')
plot.add(results.F[82,:], s=100, color='c')
plot.add(results.F[4,:], s=100, color='m')
plot.add(results.F[42,:], s=100, color='r')

plot.show()
```

E

Ashby plot

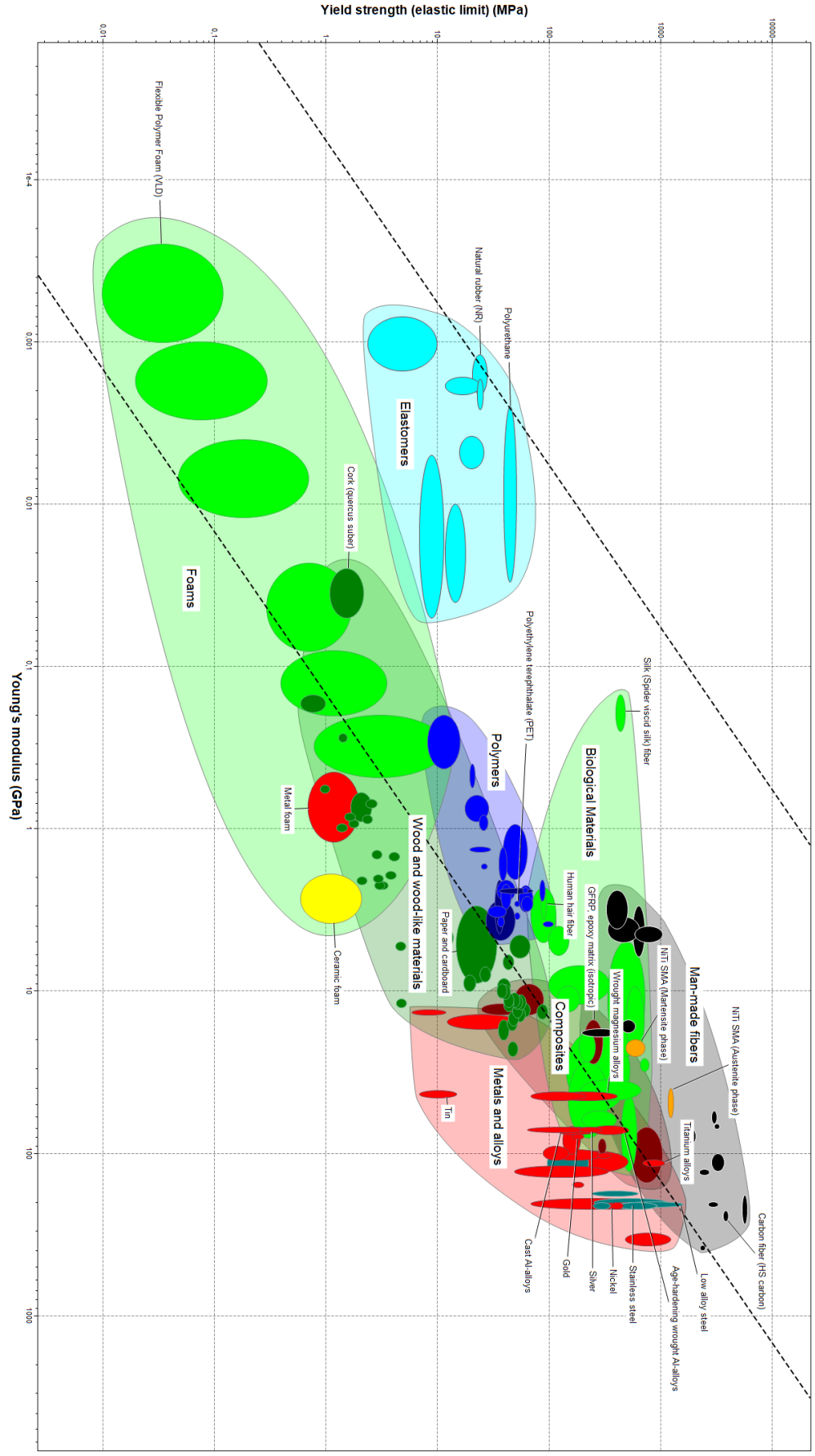


Figure E.1: Ashby plot ratio yield strength to Young's modulus