

Department of Precision and Microsystems Engineering

A single input contact-based compliant switching mechanism

Martin Frederik Hendrik de Jong

Report no : 2023.029
Coach : Dr. ir. D.(Davood) Farhadi, ir. M. (Malte) ten Wolde
Professor : Dr. ir. D.(Davood) Farhadi
Specialisation : Mechatronic System Design (MSD)
Type of report : Thesis
Date : 25-5-2023

A single input contact-based compliant switching mechanism

by

M.F.H. de Jong

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday May 25, 2023 at 16:00.

Student number: 4493370
Project duration: September, 2021– May, 2023
Thesis committee: Prof. dr. ir. J. L. Herder, TU Delft, chair
Dr. ir. D. Farhadi, TU Delft, supervisor
Ir. M. ten Wolde, TU Delft, daily supervisor

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Contents

1	Abstract	1
2	Introduction	3
3	Methods	5
3.1	Analytical derivation	6
3.2	Design embodiment	9
3.3	Finite element analysis	9
3.4	Fabrication and experimental setup	10
4	Results	13
5	Discussion	17
6	Conclusion	19
	Bibliography	21
A	Analytical Derivation	23
A.1	Bottom beam	23
A.2	Rigid Link	25
A.3	Top beam	26
B	MATLAB scripts	29
B.1	Parameters	29
B.2	Top beam actuation.	30
B.3	Bottom beam actuation.	34
B.4	Functions	38
B.4.1	Constraints	39
B.4.2	Vector loop functions	39
B.4.3	Equation setup.	41
C	Design and Fabrication	45
D	Finite Element Analysis	47
E	Literature Paper	59
F	Design variations	69

1

Abstract

In the field of mechanical metamaterials, unconventional physical properties are realized by changing the geometric structure of a unit cell. A metamaterial is composed of numerous unit cells. Researchers have taken it a step further by making a reprogrammable unit cell. A reprogrammable metamaterial has additional elements or mechanisms in the unit cells that allow their properties to be modified. The state of the unit cell corresponds to a physical property. Switching this state requires an external stimulus. Each unit cell in the metamaterial requires an external stimulus to realise a distinctive state. To decrease the number of external stimuli for a tessellated structure, a unit cell is required that switches state depending on stored information. The focus of this thesis is to design a state switching mechanism for a single unit cell. Recently developed state dependent switching mechanisms consist of parallel distributed compliant beams connected in the centre. An off-centre actuation of a single beam requires less input force than a beam actuated in the centre. There is no model that can be used to develop a state switching mechanism with the ability to change the position of the connecting element. The connecting element must be flexible to allow for rotation. Therefore in this thesis a Pseudo-Rigid-Body Model (PRBM) of a single input switching mechanism is developed that includes the off-centre connection to perform an analysis of the key parameters. The model consists of lumped beams to have a clear deformation path. The model is developed in MATLAB and validated with a finite element model (FEM). Additionally, a 3D printed prototype is made and experimentally validated to compare with the PRBM and FEM simulation. This model enables the ability to understand the effect of the flexible connecting segment and the decrease in force magnitude to actuate the system. The geometrical advantage can be tuned through preload and the ratio between the rigid beam segments. The developed model is a powerful tool that can be used to validate the functionality of any set of parameters of a coupled beam contact-based state switching mechanism.

2

Introduction

Metamaterials have unconventional properties that differ or even surpass the properties of the constituent material [1]. The property is realized by designing a geometric structure, other than solid, that is used as a building block, or unit cell. These unit cells are tessellated and form the metamaterial. These metamaterials allow for tuning of a desired physical property, such as the Poisson ratio [2, 3], shape transformation [4–8], energy absorption capacity [9–11], or stiffness [12]. All of these metamaterials are tuned to have a desired property before production. Researchers have taken it a step further by making a reprogrammable unit cell. A reprogrammable metamaterial has additional elements or mechanisms in the unit cells that allow their properties to be modified post fabrication. The state of the unit cell corresponds to a physical property. For example, a metamaterial can be reconfigured into a different geometry by applying a magnetic field [13], heat [14], or an electric current [15]; or the stiffness of a material can be tuned by changing the pressure [16, 17].

The reprogrammable behaviour of a metamaterial enables the incorporation of additional properties into materials, such as mechanical computing [18], memory [19], and learning [20, 21]. In one example described in [19], a bistable unit cell is utilized as a memory element. The state of this element, which corresponds to a specific stiffness, can be controlled by an external electromagnetic field acting as a stimulus. To achieve a distinct state in the tessellated structure, each unit cell requires control through an external stimulus. This can be achieved by either switching the state of a single unit cell or by applying a large electromagnetic field that switches all unit cells to the same state. Since the electromagnetic field is applied in two opposite directions, it is considered as two separate inputs. To minimize the number of inputs needed, the unit cell should switch its state based on the stored information, i.e., the current state of the unit cell.

To achieve a unit cell that changes state depending on stored information, the unit cell itself requires a state-switching mechanism. Similar to the bistable unit cell in the example, the mechanism should not require energy to remain in the current state. State of the art state switching mechanisms are based on latch-lock mechanisms [22–24] or on the bi-stability of a buckling beam [25–29]. In the latch-lock mechanism, a spring element locks onto a latch when actuated and unlocks when actuated again, returning to its initial state. In [28] an array of bistable beams interact with one another. These beams sequentially snap after each actuation and thereby count the number of inputs. The limitation in this design is that it has to be reset manually. In [29] a single input switching mechanism is developed that utilizes the second buckling mode of a single beam. The monolithic design has a small surface area, but it does not have clearly distinct stable states. In [27] a single input switching mechanism with a small surface area is developed. This mechanism consists of double clamped curved beams that are connected in the centre.

It is expected that a connection position other than the centre results in a lower actuation force, because the off-centre actuation of a buckling beam results in a lower actuation force [30–33]. The models used for the development of these mechanisms lack the ability to change the position of the connecting element. Due to the complex deformation of a distributed compliant beam it can't be used for a beam with an off-centre connection. The position and orientation of the link between the beams are influenced by various modes during actuation, making it challenging to incorporate them into a model. The link must be flexible to account for the different modes of the beams. Additionally, the pre-curved beams result in asymmetric

actuation, requiring different actuation displacements depending on the state.

The primary objective of this thesis is to develop a model for a double clamped switching mechanism. The model will utilize lumped beams, which contain distinct deformation points where potential energy is stored. The model takes into account the variation of the connection position and the actuation position. Through this model, an analysis is conducted to enhance the understanding of the double clamped switching mechanism. Moreover, the model will enable the fine-tuning of the design for specific applications by selecting appropriate values, such as the preload displacement, to achieve symmetric actuation and leverage the geometrical advantage.

The development of this model will involve several steps. First, the working principle of an existing switching mechanisms described. The changes made to this design in the developed model are motivated. A pseudo-rigid-body model (PRBM) is developed and this model is solved for the actuation force using the Lagrangian. The insights acquired from this model are used to design and fabricate a decimetre scale prototype. This prototype is evaluated in a finite element model (FEM) and experimentally validated to compare the force displacement characteristic resulting from the analytical model. From these results, the limitations of the model are addressed.

This thesis is structured as follows. First, the analytic derivation and the embodiment of the design are explained in Chapter 3. Then in Chapter 4, the analytical performance is compared with FEM simulations and experimentally validated. In Chapter 5, the results will be discussed. Finally, Chapter 6 concludes the results.

3

Methods

The model development starts by considering a state of the art system, as depicted in Figure 3.1 [27], which consists of a single input micro electro-mechanical system (MEMS) device with two stable states. This particular system is chosen due to its low surface area. It comprises a double distributed compliant curved-beam structure and a V-beam actuator. Depending on the state of the curved-beam structure, the actuator makes contact with either a forward push rod or a backward push rod. As a result, a moment is generated on the actuated beam, causing it to snap through when sufficient force is applied. This moment-driven input reduces the overall size of the mechanism compared to an orthogonal orientation. To suppress deformation in asymmetric modes and maintain a linear path of the centre shuttle, the beams are connected at the centre. Additionally, the beams are fabricated as curved, resulting in different potential energy levels for the stable states. This indicates asymmetric state switching of the mechanism. By adjusting the curvature of the beams, the energy potential levels can be modified.

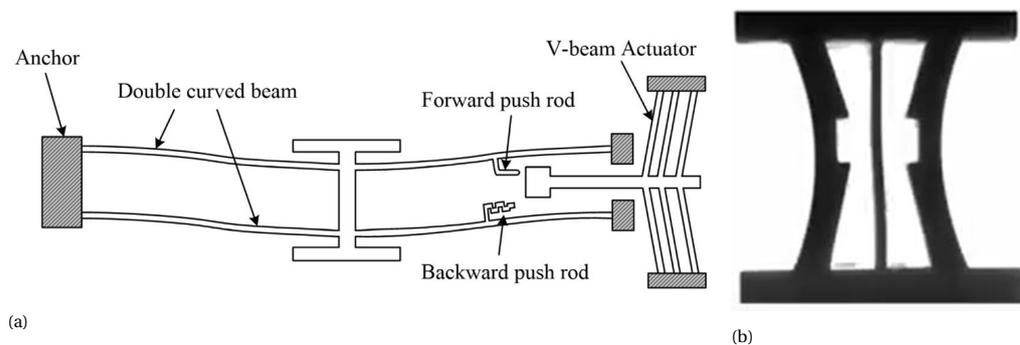


Figure 3.1: State of the art mechanisms a) A single input MEMS device with a V-beam combined with a moment driven input with courtesy of [27] b) A single input monolithic bistable mechanism with courtesy of [29]

In Figure 3.1b a monolithic single input bistable mechanism is presented [29]. This mechanism utilises the second buckling mode of a single beam. The state of the mechanism is denoted by the slender beam in the middle of the mechanism. This beam can either be curved to the left side or to the right side. By compression of this mechanism in vertical direction the slender beam bends towards the side of the curvature. In the presented state it would deflect to the left side. The slender beam makes contact with the left supporting structure at the lower tip. This contact induces the buckling of the bottom part of the slender beam to the right. When the vertical compression is released the slender beam is curved towards the right and will remain in that state. If the mechanism is compressed again it will switch to the initial state. As mentioned in the introduction the limitation of this design is that the stable state of the mechanism is not distinct.

Symmetric actuation can be realised in MEMS device with a moment driven input by fabricating it with initially straight beams and applying a preload displacement to one of the anchors. That results in the same potential energy level for the two stable states. The linear path of the centre shuttle can be desired for certain applications but the off centre connection presumably decreases the actuation force. As mentioned in

the introduction the existing model can't be used for a beam with an off-centre connection due to the complex deformation of a distributed compliant beam. The position and the orientation of the link between the two beams are affected by several modes during the actuation. These modes are difficult to incorporate in a model. Therefore a model with lumped beams is proposed that captures the behaviour well and has clear deformation points where the potential energy is stored. A lumped beam has clear deformation points because it consists of rigid (lumped) elements connected with nodes. A lumped four-bar-mechanism with torsion springs in each node is shown in Figure 3.2. The lumped mechanism is presented in an unstable equilibrium, two stable equilibria and the position without preload displacement.

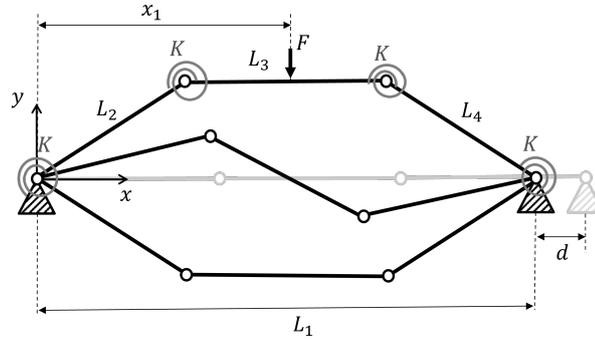


Figure 3.2: Analytic model of a lumped beam with a preload displacement d with one degree of freedom. The beam is depicted in three positions under a load F in y -direction. The beam has a clear buckling path depending on the geometric parameters L_2, L_3, L_4, d and spring stiffness K

The actuation force of the lumped beam is lower when the actuation position x_1 in Figure 3.2 is off-centre. However, it does result in a longer actuation displacement [30–33]. By combining the known deformation of the lumped beam model and the benefits of the off-centre connection an improved state-switching mechanism can be designed. This mechanism can be designed with a preload to ensure symmetric actuation for the state switching and tunability of potential energy levels in stable positions. The moment-driven input will be incorporated to decrease area size. The steps taken to arrive at a final design will be explained in the next section of the analytic derivation.

3.1. Analytical derivation

In this section the analytical derivation of the design is explained. First, a PRBM will be developed that will enable the derivation of the kinetics. Then the potential energy equations are set up and solved with the Lagrange multiplier method in order to find a minimum of potential energy for every configuration. An expression for the force as function of the input displacement is obtained from the Lagrange multiplier. This expression can be compared to the FEM simulations and the experimental validation.

In the development of a model, the PRBM illustrated in Figure 3.3a was developed in order to investigate the desired geometry of the mechanism, see Figure 3.3a. This model consists of two lumped beams connected off-centre by a fixed-fixed beam that has two nodes between the fixed ends that allow for rotation of the connection. Using a spring as a connection would also allow this rotation, but would only account for the stiffness in the longitudinal direction of the spring. The kinematics of the design are derived by setting up three vector loops as given in Figure 3.3b. The first vector loop describes the relations between the angles of the bottom beam $\theta_2, \theta_3, \theta_4$. The second vector loop describes the relation between the angles of the top beam $\gamma_2, \gamma_3, \gamma_4$. The third vector loop describes the relations between the angles of the link $\beta_2, \beta_3, \beta_4$. There is a constant angle present between β_2 and θ_4 as a result of the chosen geometry. The same holds for γ_4 and β_4 . The top and bottom beams are considered to have the same geometry. Both beams are initially straight and the preload d imposed on both beams is the same and similar to the defined preload of the single beam in Figure 3.2. The range of motion of the mechanism is geometry dependent and is evaluated by Grashof's law. A rocker-rocker type four-bar mechanism has a limited range of motion and no fully rotating links. Such a four-bar mechanism must meet the requirement that the sum of lengths of the ground link (L_1) and floating link (L_3) is larger than the sum of lengths of the input link (L_2) and the output link (L_4). Since $L_1 = L_2 + L_3 + L_4 - d$ the requirement can be written as $L_2 + 2L_3 + L_4 - d > L_2 + L_4$ and simplified to $2L_3 > d$ to have a rocker-rocker type four-bar mechanism.

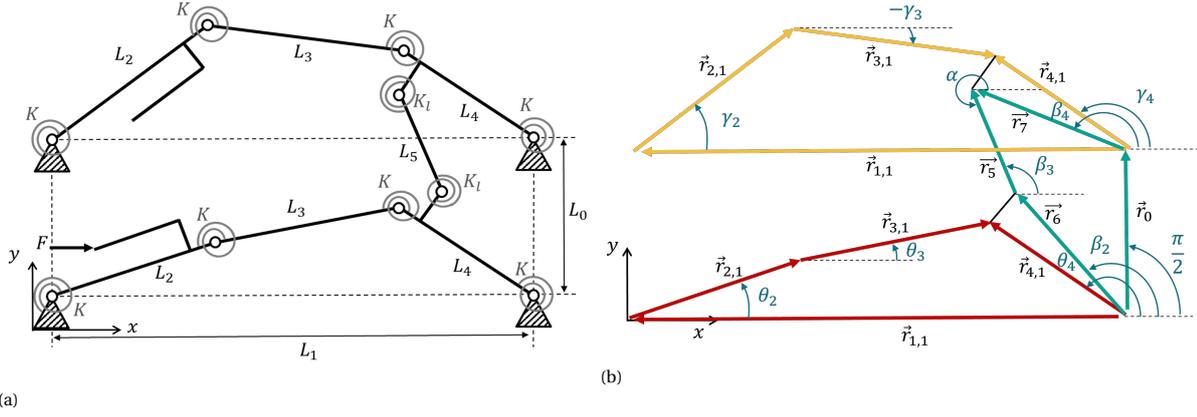


Figure 3.3: a) The degrees of freedom and potential energy are derived with this pseudo rigid body model. b) The three vector loops that are used to derive the kinetics. The red vector loop is considered as the bottom beam. The yellow vector loop as the top beam and the blue vector loop presents the link. The red and yellow vector loop have a similar geometry

In each vector loop, the sum of the vectors are set to zero, then each vector is expressed in its components. By setting up the vector loops correctly the result is two equations with a total of two unknown variables. A complete derivation of the vector loops and the resulting relations between the angles is provided in Appendix A. The vector loops are used to derive the kinematics for a full actuation cycle. In the Lagrange multiplier method the angles in each configuration are used as an initial guess to decrease the computational cost.

The total potential energy is described as the sum of each individual component. In this design, all potential energy comes from torsion springs. The potential energy in a single torsion spring is $\frac{1}{2}K\Phi^2$, where K is the torsional stiffness of the spring and Φ is the relative rotation of the spring. A quasi static study is performed to omit the kinetic energy effects. With the chosen vectors in the vector loops, it is not possible to define the position of the mechanism with only one variable. For each angle given, there are two possible solutions for the other two angles. Besides that, for the vector loop of the top and bottom beam, the solution will contain an angle between π and $-\pi$ while the angle of θ_4 and γ_4 is modelled as an angle between 0 and 2π . This is not a problem for the position analysis. But it gives faulty results for the potential energy calculation. The problem of negative angles is solved by adding 2π to each position angle of θ_4 and γ_4 that becomes negative. The total potential energy $E_{tot} = P_{bot} + P_{top} + P_{link}$. The fabricated position is denoted by the angles $\theta_{2,0}$, $\theta_{3,0}$, $\theta_{4,0}$, $\gamma_{2,0}$, $\gamma_{3,0}$, $\gamma_{4,0}$, $\beta_{2,0}$, $\beta_{3,0}$, α_0 . The potential energy in the bottom beam

$$P_{bot} = \frac{1}{2}K(\Phi_1^2 + \Phi_2^2 + \Phi_3^2 + \Phi_4^2). \quad (3.1)$$

With $\Phi_1 = \theta_2 - \theta_{2,0}$, $\Phi_2 = (\theta_3 - \theta_{3,0}) - (\theta_2 - \theta_{2,0})$, $\Phi_3 = (\theta_3 - \theta_{3,0}) + (\theta_4 - \theta_{4,0})$ and $\Phi_4 = \theta_4 - \theta_{4,0}$. For the top beam the potential energy

$$P_{top} = \frac{1}{2}K(\Phi_5^2 + \Phi_6^2 + \Phi_7^2 + \Phi_8^2). \quad (3.2)$$

With $\Phi_5 = \gamma_2 - \gamma_{2,0}$, $\Phi_6 = (\gamma_3 - \gamma_{3,0}) - (\gamma_2 - \gamma_{2,0})$, $\Phi_7 = (\gamma_3 - \gamma_{3,0}) + (\gamma_4 - \gamma_{4,0})$ and $\Phi_8 = \gamma_4 - \gamma_{4,0}$.

The potential energy stored in the link

$$P_{link} = \frac{1}{2}K_l(\Phi_9^2 + \Phi_{10}^2). \quad (3.3)$$

With $\Phi_9 = \beta_3 - \beta_{3,0}$ and $\Phi_{10} = \alpha - \alpha_0$.

The resulting kinematics and potential energy relations are used in the Lagrange multiplier method. Which is a method to find local extremes subject to certain equality constraints [34]. An imposed constraint results in a Lagrange multiplier that is equivalent to the moment required to force this constraint. This moment can be used to derive the required force in a certain direction. In general, for a function with n amount of variables

and m amount of constraints the Lagrangian function

$$\Lambda(x_1, \dots, x_n, \lambda_1, \dots, \lambda_m) = f(x_1, \dots, x_n) - \sum_{k=1}^M \lambda_k g_k(x_1, \dots, x_n). \quad (3.4)$$

This means that for multiple constraints the equations to solve for are

$$\nabla \Lambda = 0 \iff \begin{cases} \nabla f(\mathbf{x}) - \sum_{k=1}^M \lambda_k \nabla g_k(\mathbf{x}) = 0 \\ g_{1:M}(\mathbf{x}) = 0 \end{cases} \quad (3.5)$$

The total amount of equations to be solved is $n + M$ with $n + M$ unknowns. The proposed lumped compliant design can be solved with this approach. For reaction forces of interest, the energy function and constraints for each link can be stated. Not all reaction moments that result from the Lagrange multiplier method are of interest. However they are necessary to properly constrain the system.

In the total potential energy equation there are eight angles $\theta_2, \theta_3, \theta_4, \gamma_1, \gamma_2, \gamma_3, \gamma_4, \beta_3, \alpha$ needed to calculate the potential energy. These angles are presented by x_n in 3.6. For each vector loop there are two constraints needed. Besides that, there is one constraint needed to impose a virtual angle η of the input. This means that the total amount for unknowns to be solved for is 15. All the equations are set up according to 3.5. This results in the following systems of equations to be solved for when the bottom beam is actuated.

$$eq_n = \frac{\partial E_{tot}}{\partial x_n} - \left(\lambda_1 \frac{\partial g_1}{\partial x_n} + \lambda_2 \frac{\partial g_2}{\partial x_n} + \lambda_3 \frac{\partial g_3}{\partial x_n} + \lambda_4 \frac{\partial g_4}{\partial x_n} + \lambda_5 \frac{\partial g_5}{\partial x_n} + \lambda_6 \frac{\partial g_6}{\partial x_n} + \lambda_7 \frac{\partial g_7}{\partial x_n} \right) = 0 \quad (3.6)$$

$$g_1 = 0 \Rightarrow \theta_2 - \eta = 0 \quad (3.7)$$

$$g_2 = 0 \Rightarrow L_2 \sin \theta_2 + L_3 \sin \theta_3 - L_4 \sin \theta_4 = 0 \quad (3.8)$$

$$g_3 = 0 \Rightarrow L_2 \sin \gamma_2 + L_3 \sin \gamma_3 - L_4 \sin \gamma_4 = 0 \quad (3.9)$$

$$g_4 = 0 \Rightarrow L_2 \cos \theta_2 + L_3 \cos \theta_3 + L_4 \cos \theta_4 - (L_2 + L_3 + L_4 - d) = 0 \quad (3.10)$$

$$g_5 = 0 \Rightarrow L_2 \cos \gamma_2 + L_3 \cos \gamma_3 + L_4 \cos \gamma_4 - (L_2 + L_3 + L_4 - d) = 0 \quad (3.11)$$

$$g_6 = L_6 \sin \beta_2 + L_5 \cos \beta_3 - L_7 \cos(\beta_4) = 0 \quad (3.12)$$

$$g_7 = L_6 \cos \beta_2 + L_5 \sin \beta_3 - L_7 \sin(\beta_4) - L_0 = 0 \quad (3.13)$$

These equations are implemented in a MATLAB script which can be found in Appendix B. The force displacement curve of the design for different preloads is shown in Figure 3.4c. The top and bottom beam are considered to have the same dimensions. Due to this symmetry, the actuation of the top and bottom beam results in the same force displacement relation. By increasing the preload, the input displacement and the actuation force increases. The preload must be as low as possible to have a low actuation force and input displacement. On the other hand, it must be high enough to have distinct equilibria. Moreover, as stated in the previous section the output displacement increases with an increase in preload. This means there is a trade-off between the actuation force and the desired output displacement.

The input mechanism is designed as two parallel guided beam flexures [35]. The PRBM of the input mechanism is given in Figure 3.5. The parasitic motion in y-direction is two tenths of a mm during the range of motion for snap through as can be seen in Figure 3.4b.

The equation used to calculate the force displacement relation is given in 3.14 with E being the Young's modulus of the material and I the second moment of inertia of the beam flexures. L_{in} is the total length of the beam flexures and L_f denotes the rigid part of the beams. For the small displacements that are required for the actuation, there is a linear relation between the force and the displacement:

$$F = \frac{8K_\theta EI\theta}{L_{in}^2 \cos(\theta)}. \quad (3.14)$$

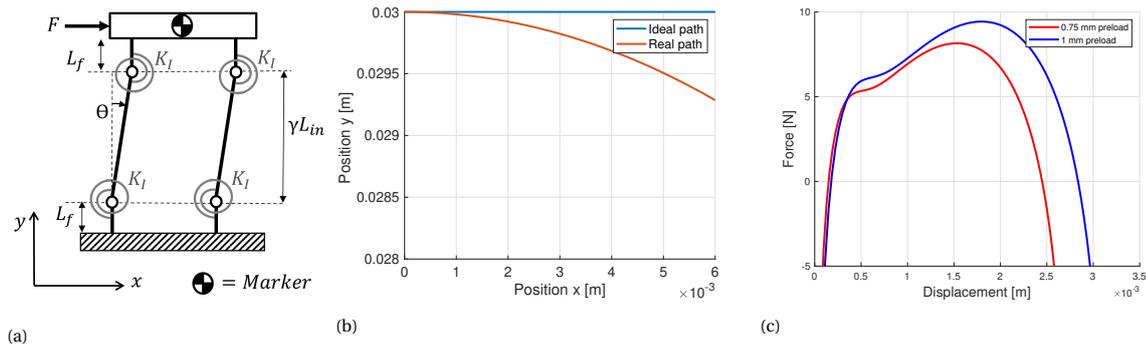


Figure 3.4: a) Pseudo rigid body model of the input mechanism presented as a fixed guided beam. The spring stiffness K_I is the same for each torsion spring b) Ideal and real path of the marker on the input mechanism. In the real path there is parasitic motion for the range of motion c) Top and bottom beam actuation for various preloads. This shows the effect of the preload on the force displacement curve.

3.2. Design embodiment

In Figure 3.5a an embodiment of the analytical model is shown. This embodiment is fabricated as a proof of concept. All parameters are implemented in MATLAB and the scripts can be found in Appendix B. A maximum size is chosen to be on a decimetre scale. The smallest dimension with the chosen 3D printing production process is 0.4 mm, with a 0.4 mm nozzle of a 3D printer. Since a single line is not reliable in 3D printing at least two lines must be printed next to another. Therefore the flexures should be at least 0.8 mm wide. During the fabrication it was observed that 0.9 mm thickness results in a reliable flexure.

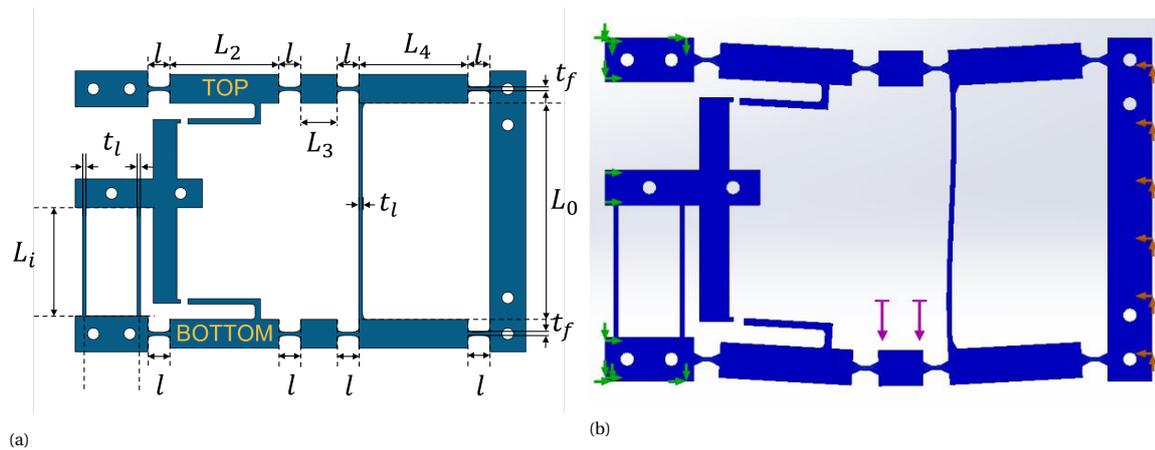


Figure 3.5: a) Top view model with important parameters: flexure length $l = 6$ mm, input mechanism beam length $L_i = 30$ mm, flexure width $t_f = 0.9$ mm, flexure width $t_l = 1$ mm, $L_0 = 60$ mm, $L_2 = 30$ mm, $L_3 = 10$ mm, $L_4 = 30$ mm b) Constraints placed in SOLIDWORKS 2022 simulation environment. The green arrows on the left on the top and bottom beam are the fixed anchors. The purple arrows in the middle represents a force of 0.1 N in the first step to ensure a stable state after preload. The orange arrows on the right are the preload displacement

3.3. Finite element analysis

The Solidworks simulation environment is used for the finite element analysis of the designed model, because this easily allows for interaction between contact surfaces. It is performed as a nonlinear study with a 2D approximation of the design to decrease the computation time. The setup with constraints is shown in Figure 3.5b. The material selected is the existing Polyethylene terephthalate glycol (PETG) material in the library with a minor adjustment of the Young's modulus to 2.1 GPa to resemble the PETG raw filament used for production. Other material parameters: Poisson's ratio of 0.37, mass density of 1270 kg/m^3 and a tensile strength of 57.3 MPa. A contact pair is created between the input mechanism and the pushing rods of the top and bottom beam. A fine mesh is created of the entire geometry with a mesh refinement for the leaf flexure connections between the rigid links to approximate the curvature of the fillets and have a better stress distribution in these flexible elements. The mechanism is evaluated in five steps. In the first step the left anchor

beams are fixed. Next to that the preload displacement is added on the right side of the mechanism. To make sure the system goes to one stable state a force of 0.1 N is added on the bottom beam. In the second time step the preload displacement is also fixed and the force on the bottom beam is removed. In the same time step the input mechanism is displaced for five mm in positive x-direction. During this time step the input mechanism makes contact with the top beam and switches the state of the coupled beams. In the next time step the input is moved back to the initial position. These two steps are repeated to switch the mechanism back to the initial state.

In Figure 3.6 the two stable states are given. The preload used in this study is one mm. It is observed that the simulation does not work for lower preloads because the input mechanism does not make contact with the pushing rods on the right location. On the right side of this figure the maximum Von Mises stresses are given during the entire simulation. The peak stresses occur at the two middle leaf flexures on the top and the bottom beam. This is where the largest rotation of the flexures takes place. The peak stress given here is 130 MPa, this is higher than the yield stress of 50 MPa of the selected material PETG. In Appendix D the setup and results from a simplified ANSYS model are given. In this model the input mechanism is omitted because the peak stresses were of interest and they do not occur in the input mechanism. The peak stresses are around 80 MPa. The exact yield strength of the material is not given in the specifications but assuming a regular PETG material it should be between 50 and 100 MPa. In both FEM models it indicates that the peak stresses result in plastic deformation. From experience it is observed that the peak stresses resulting from a FEM are higher than the actual stresses. Therefore, a prototype is made and experimentally validated.

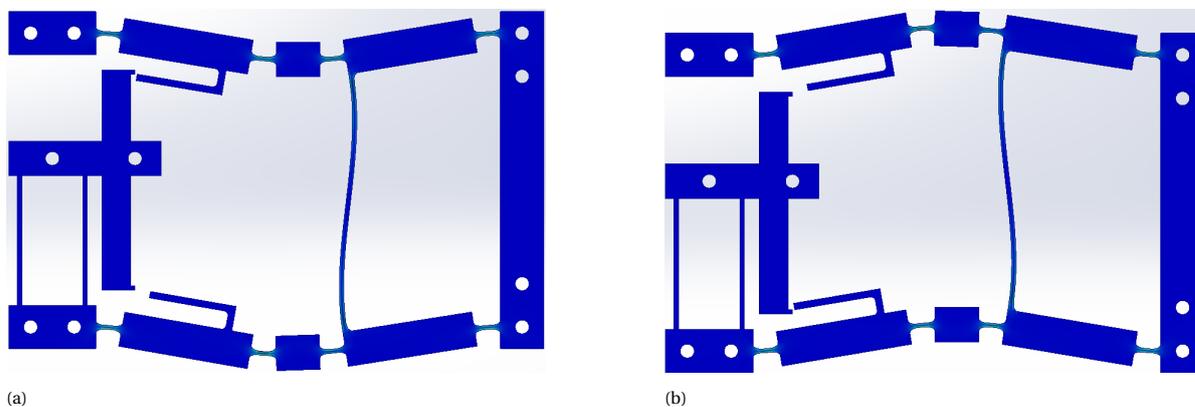


Figure 3.6: Two stable states of the switching mechanism in FEM simulation. The maximum stress is at the middle two flexures of the top and bottom beam and has a value of 130 MPa

3.4. Fabrication and experimental setup

Most parts of the design have been 3D printed with a PRUSA MK3 printer. Technical drawings with all the measurements and sizes can be found in Appendix C. A figure of the prototype is given in Figure 3.7. The prototype consists of four parts, a bottom plate, the functional switching mechanism, a clamp and the preload bar. The bottom plate is made from polylactic acid (PLA) and has grooves that prevent movement of the prototype in the y-direction. Next to that the bottom plate has mounting pads on the sides to easily mount it with clamps to the Thorlabs frame. The functional unit cell is made from PETG. The Prototype is fixed on one side of the prototype to the bottom plate with two three mm thick bolts with m3 threaded nuts that prevent the out of plane motion in the z-direction. The clamp is also made from PLA and is made to allow a maximum preload displacement of 1 mm. The preload bar is also made from PLA and fixated between the clamp and knobs on the bottom plates. This preload bar can have two different orientations that result in a preload of 0.75 mm or 1 mm.

To validate the performed analytic studies and FEM simulations the manufactured prototype was tested. In the experiments a linear input motion was applied to the prototype while the input force and the output displacement were measured. The experimental setup can be found in Figure 3.8a. The linear motion stage was controlled using a PI mercury motion controller. The linear input was ensured by aligning the prototype with the PI M505.4DG low-profile translation stage. This translation stage has a design resolution of 0.017 μm and zero-backlash due to the recirculating ball screw drive. A FUTTEK LSB200 FSH03875 load cell was attached

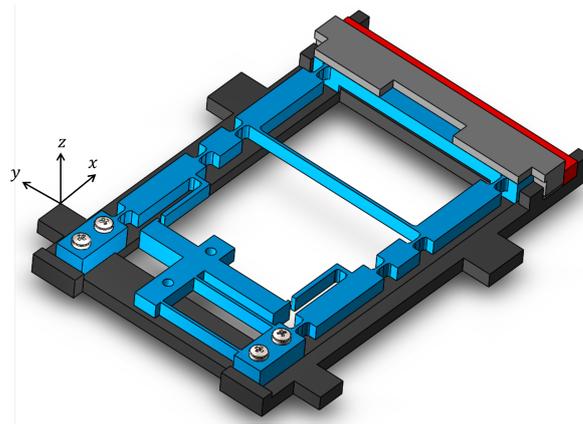


Figure 3.7: CAD model of the fabricated design, The blue structure is the function switching mechanism. The black structure is the bottom plate. The grey structure is the clamp and the red element is the preload bar.

between the mechanism and the stage. This load cell can handle forces up to 45N and has a maximum inaccuracy of $\pm 0.245\%$ of the rated output, due to some non-linearity, hysteresis and non-repeatability. The linear stage itself is mounted to a frame that is standing with rubber feet on a table where the prototype is fixed on. Between 35N and 40N this frame starts to move because the threshold of static friction is surpassed. This frame is not fixed due to safety considerations. The prototype is fixated with clamps on a Thorlabs frame. Several markers were added on the prototype to track the motion of the switching mechanism during the experiments. A phone camera with a resolution of 48 MP and linear field of view was used to film the movement of the mechanism.

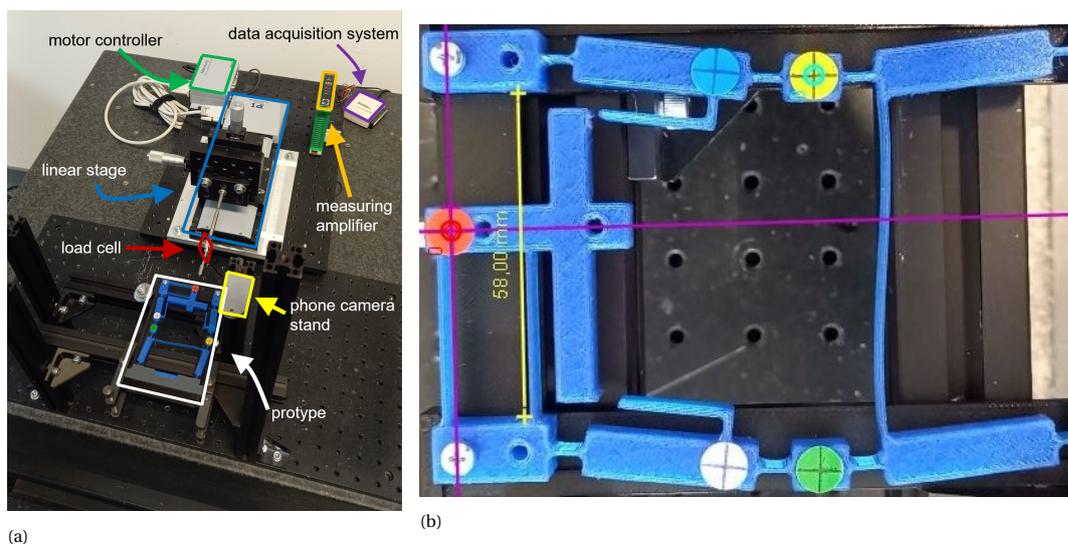


Figure 3.8: Experimental setup and tracker layout a) Experimental setup with prototype b) Setup of marker tracking.

In the post processing of the experiments an open source software program called tracker was used to retrieve the relation between the input and the output displacement. A picture of the layout is provided in Figure 3.8b. A red marker is placed on the input displacement position. A yellow marker is placed on the middle rigid beam of the top beam to measure the midpoint output displacement. A green marker is placed on the middle rigid beam of the bottom beam, to measure the midpoint output displacement. The program is used in the following manner, a size calibration bar of 58mm is added as the known distance between two anchor beams on the side of the input mechanism. The purple axis is used as reference for all measurements. The origin's centred on the red marker in the first frame. The auto tracking of the markers is set up the same for each measurement. In the first frame of the video a key frame is made that creates a template for the next frames to compare to. The position is recorded for each consecutive frame. If the resemblance with the key frame is too low, a new key frame is manually selected. This happens when the displacement of the tracking object is

relatively large or when the frame itself is blurry. For each measurement video, the tracking is performed for the red, yellow and green position markers.

4

Results

Both the analytical model and the FEM are verified by comparing them to experiments. The results of both methods and the experimental results are presented in Figure 4.1. A single measurement is shown in the figures. The peak force resulting from the experiments is about 6N, while the force resulting from the analytic model is around 12N. The FEM simulation gives an even higher peak force of 13N. The maximum force is reached just before the mechanism snaps through to the next stable state. For the first millimetre of input displacement, the reaction force comes solely from the input mechanism. The three separate graphs result in a similar slope which indicates that the approximation of the input mechanism stiffness in the analytic model is good. The top and bottom actuation of the beams in analytic model is symmetric, in the FEM it is similar, but in the experiments it has a different force-displacement curvature. It can be seen that in the experiments the input displacement required for state switching is larger for the top beam than for the bottom beam. In Figure 4.1a it can be seen that for all three graphs snap through happens between 4 and 5 mm of input displacement. Each graph has a sudden change in slope in this region when the switching mechanism snaps. In Figure 4.1b it can be seen that the snap-through for the top beam happens at a larger input displacement than the analytic model and the FEM simulation. By comparing these two figures it can be seen that for all graphs in gradient at snap through in Figure 4.1b is smaller than the gradient of the graphs in Figure 4.1a.

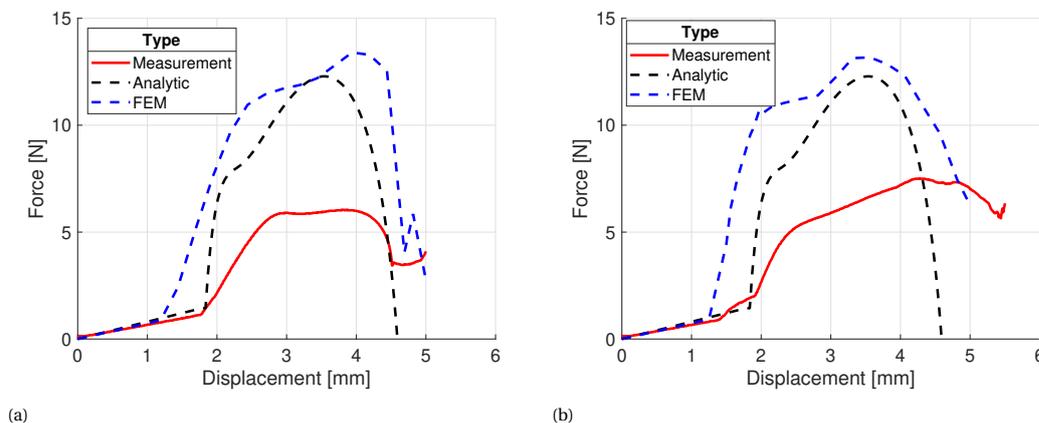


Figure 4.1: Force versus displacement curves for one mm of preload displacement a) Bottom beam actuation b) Top beam actuation

The differences between the top and bottom beam actuation in the experiments can be seen more clearly by taking a look at the position of the mechanism at a certain input displacement. Figure 4.2 displays figures created with the tracker software. The presented input displacement is retrieved from the video analysis. In Figure 4.3 a vertical dotted line is displayed at the corresponding input displacements. The figures correspond to the top and bottom beam actuation with a preload of 0.75 mm.

At 0 mm the mechanism is in a stable state. Between 1.5 mm and 2 mm the force in the top beam measure-

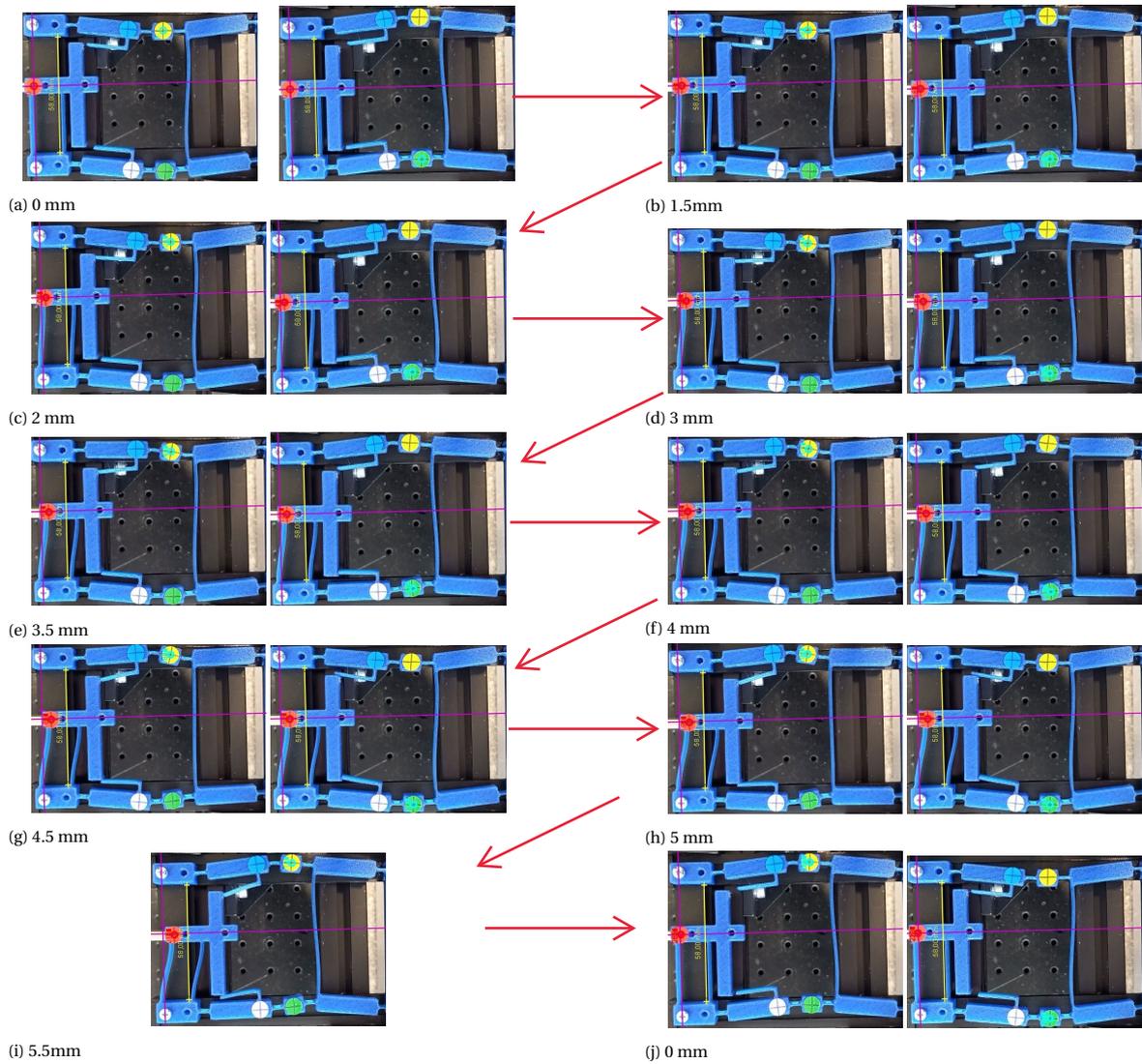


Figure 4.2: Position tracking for various input displacement steps with 0.75 mm preload displacement. In each sub figure, the left figure is top actuation and the right figure is bottom actuation. a) The force is solely dependent on the input mechanism b) Tip contact with the switching mechanism for top beam actuation c) Full contact and clear stiffening effect d) Force gradient decreases slightly e) Force gradient increases slightly f) First peak force for the top beam actuation. Snap through has occurred for bottom beam actuation g) Top beam in unstable equilibrium h) Top beam and bottom beam are buckled i) Full actuation, no figure for the bottom beam actuation because it was actuated until 5 mm e) Configuration when input displacement is released

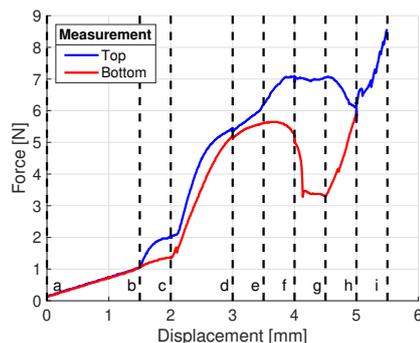


Figure 4.3: Force displacement graph for a preload of 0.75 mm for top (blue) and bottom (red) beam actuation, labels a-i in the figure correspond to the sub figures in Figure 4.2

ment is higher than the bottom beam measurement. This difference is caused by contact of the tip of the input mechanism with the pushing rod. At 3 mm input displacement there is a decrease in the stiffness due to the position of the top beam. The left most rigid beam of the actuated beam is parallel with the input displacement. This decrease in stiffness is present for both measurements. At 3.5 mm during the actuation of the top beam there is a slight contact between the side of the input mechanism and the pushing rod of the bottom beam. This contact increases the actuation force. At this displacement there is no contact between the input mechanism and the top beam for the bottom beam measurement. At 4.5 mm there is a second peak in the force displacement graph for the top beam. At this point the top beam is almost at an unstable equilibrium position and wants to snap through to the other stable equilibrium. However the bottom beam is not yet over this threshold and prevents this snap through. At 4.5 mm input displacement the mechanism has switched state for the bottom beam actuation. For the top beam actuation the mechanism snaps through at 5 mm input displacement. After the state switching the input displacement can be increased without switching the state back to the initial state. By removing the input displacement the state of the mechanism has been switched to a new equilibrium. By actuating the system again the input mechanism will be in contact with the other beam first and follow that force displacement graph during actuation.

In Figure 4.4 the input vs output relation is given. These displacements correspond with the force displacement graph for a preload of 0.75 mm presented in Figure 4.3. The two stable states can clearly be seen at -4 mm and 4 mm output displacement. The top beam is actuated until 5.5 mm and the bottom beam until 5 mm. The figure shows that a larger input displacement is required to switch state for the actuation of the top beam (5 mm) than the actuation of the bottom beam (4 mm). The actuation of the bottom beam has a large gradient at snap through. For the top beam actuation it can be seen that just after snap through the bottom beam is not at the stable equilibrium yet. This is caused by contact between the input mechanism and the bottom beam pushing rod. By removing the input displacement it slowly settles. This contact is less present during the actuation of the bottom beam because the required state switching input displacement is lower.

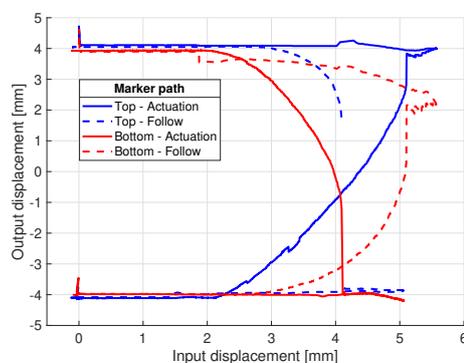


Figure 4.4: Input vs output displacement. The top beam (blue lines) corresponds to the yellow marker and the bottom beam (red lines) to the green marker in Figure 4.2. The solid lines present the actuated side and the dashed lines the non-actuated side.

In Figure 4.5a the results of the measurements with 0.75 mm preload are shown. There is a difference in the actuation of the top and bottom beam. The consecutive measurements of the top beam are decreasing in force up to a certain extent. Although they are similar in shape this indicates some sort of settling of the mechanism. The measurements are plotted as a scatter. In the figure it can clearly be seen that there is a stiffening effect occurring when the input mechanism comes in touch with the switching mechanism. For a preload of 0.75 mm this backlash is around 1.5 mm for the top beam and 2 mm for the bottom beam. The top beam switches state around 7 N of force and 5 mm of input displacement, while the bottom beam switches state at 6 N of force and 4.2 mm of input displacement. In Figure 4.5b the results of the measurements with 1 mm preload are shown. Similar to the experiments with 0.75 mm preload there is a difference in the actuation of the top beam and the bottom beam. The measurements do not deviate much from one another. The stiffening effect of the contact between the input mechanism and the main mechanism occurs earlier than in the 0.75 mm preload experiments. For the top beam actuation this is around 1.8 mm and for the bottom beam this is 1.4 mm. This means that with a larger preload displacement the actuation force increases and the input displacement also increases. For the top beam the peak actuation force is 7.8 N and for the bottom beam this is 6.2 N. The total input displacement for the system to switch to the next state has also increased to 4.5 mm for the bottom beam actuation and for the top beam to 5.4 mm of input displacement.

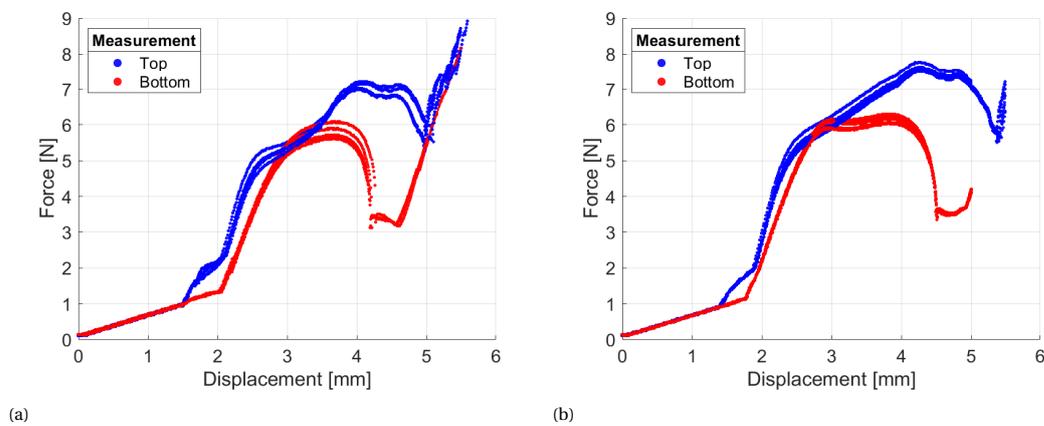


Figure 4.5: Performed measurements for two separate preloads for the top and bottom beam a) 0.75 mm preload b) 1 mm preload

5

Discussion

The results from the analytical model, finite element analysis, and experiments show that the actuation of the top and bottom beams is not symmetric. The top beam snaps later than the bottom beam. One of the reasons can be that the input mechanism is only fixed to the anchors of the bottom beam. This asymmetry of the input mechanism causes it to slightly tilt towards the bottom beam during the actuation of the bottom beam. This direction change leads to a lower actuation force. Another reason for the asymmetric actuation can be caused by the boundary conditions in the experiments, such as the preload displacement, linear actuation of the input mechanism, the fixed bottom plate or a misplaced screw due to the tolerances of the 3D printing process. The 3D printing process used is not very accurate. By combining several 3D printed parts the resulting preload on the beams may be different than the intended preload displacement. A 3D printed structure consists of layers. When printing a hole these layers can be misaligned in z-direction and cause a position shift in x or y-direction when a screw is forced through this hole, see Figure 3.7 for the directions. Besides that, the printing accuracy of a 3D printer can be up to 0.3 mm for all axis if the printer is not properly calibrated. The tolerances of a 3D printer are 0.1 mm. Besides that, a 3D printed structure shrinks when it cools down. All of these fabrication inaccuracies can cause an unequal preload for the top and bottom beam. The effect of a unequal preload is further explained in Appendix F. A higher preload results in a higher actuation force and displacement. Since the actuation force and the input displacement for snap through for the top beam are higher than the bottom beam it appears that the preload displacement on the top beam is higher than the bottom beam.

The overall decrease in magnitude could be the cause of the preload displacement being less than the intended one mm. As mentioned in the previous section this decrease in preload displacement could be caused by the tolerances of the 3D printing process. In Figure 3.4c the effect of the preload on the force displacement curve is presented. In this figure the backlash is not yet taken into account. The preload needs to match the distance between the input mechanism and the pushing rods. For a lower preload, the distance must be smaller to have less backlash and the input mechanism must be wider to ensure contact at the right place. By decreasing the preload the total tip displacement in y-direction of the pushing rods decreases, which means that there is less room for error in the fabrication process. In the FEM, the simulation did not function properly when the preload was less than 1 mm. The input mechanism did not touch the pushing rods at the right place and thereby did not make the mechanism switch state. During the experiments, it was observed that the hooks on the input mechanism are slightly rounded towards the middle. Instead of missing the pushing rod or have direct contact, it guides the pushing rod toward the middle of the input mechanism. A close-up figure is given in Appendix F for better representation.

The high stresses observed in the finite element analysis dictate plastic deformation in the design during actuation. This would cause plastic deformation at peak stress locations. These locations are at the two flexures in the middle of the top and bottom beams. Under plastic deformation, these flexures would change in geometry. This change in geometry could decrease the peak stresses and thereby the actuation force required to switch states. In the experiments, it was observed that the actuation force of the mechanism decreased with each actuation to a certain extent. This result supports the claim that plastic deformation occurred. On the other hand, the prototype did not yield. It could also be a coincidence that in consecutive measurements the

actuation force decreased. The peak stresses resulting from the finite element analysis could be higher than the actual stresses due to the approximation of the geometry by creating a mesh.

The model could be improved by designing an equivalent distributed compliant mechanism. This would reduce the stress concentrations in the design. However a distributed compliant flexure deforms under multiple modes. The position and orientation of each segment of a flexure needs to be derived in order to capture the behaviour of the entire switching mechanism. The mathematical model required to include these positions would be complex.

A limitation of the design is caused by the contact surfaces used for the actuation of the mechanism. These contact surfaces are necessary for a functional design, but introduce friction and wear. This friction and wear decreases the functionality of the design over time. The scalability is also limited by these contact surfaces. If the device is scaled down to nanometre scale the surface contact locations are sensitive to micro stitching. To prevent the Van der Waals force to have an effect a design restriction can be made that the mechanism is not made at nanometre level. If a mechanism at this level is desired a recommendation would be to replace the contact surfaces with non-linear springs. The non-linear springs would be located between the input mechanism and the pushing rods. This addition would increase the complexity of the model, but decrease the sensitivity to micro stitching.

6

Conclusion

This thesis has introduced a pseudo rigid body model of a single input contact based switching mechanism. The model developed includes the variation of the connection position and the actuation position, which effectively reduces the actuation force of the switching mechanism. Lumped beams are utilized in the model, that contain distinct deformation points where potential energy is stored. The functionality of the model is validated through a prototype and FEM. The force displacement curves obtained from experiments exhibit lower magnitudes of force compared to the analytic derivation and FEM simulation, although their shapes remain similar. The force displacement curves obtained from FEM simulations closely resemble the curves derived analytically in terms of shape and force magnitude. The input force required at a preload of 0.75 mm is 6 N and 7 N for the bottom and top beam respectively. The input force required at a preload of 1 mm is 6.2 N and 7.8 N for the bottom and top beam respectively. The geometrical advantage is related to the applied preload and the ratio between the rigid beam segments. The middle rigid beam segment for the top and bottom beam must be smaller than the other two rigid beam segments, but larger than two times the applied preload to ensure a limited range of motion. The developed model serves as a powerful tool for validating the functionality of different parameter sets of a coupled beam contact-based state switching mechanism. The model developed has significant applications in the design of programmable mechanical metamaterials. It can be used to implement a state switching mechanism in the geometry of a unit cell.

For future research, a distributed compliant model should be looked into that reduces the peak stresses and includes the same benefits for the off-centre actuation. Furthermore, the load capacity of the design is not experimentally validated in this research. Connecting several unit cells and observing their interactions would be important to research to validate the application in a metamaterial. In addition, the developed model has its limitations with respect to scalability due to the contact surfaces in the design. A contactless mechanism could be designed by implementing a nonlinear spring between the input mechanism and the pushing rods. This would decrease the sensitivity of micro stitching when scaling down the device.

Bibliography

- [1] Katia Bertoldi, Vincenzo Vitelli, Johan Christensen, and Martin Van Hecke. Flexible mechanical metamaterials. *Nature Reviews Materials*, 2, 10 2017.
- [2] Nicholas Pagliocca, Kazi Zahir Uddin, Ibnaj Anamika Anni, Chen Shen, George Youssef, and Behrad Koohbor. Flexible planar metamaterials with tunable poisson's ratios. *Materials Design*, 215:110446, 3 2022.
- [3] Amin Farzaneh, Nikhil Pawar, Carlos M. Portela, and Jonathan B. Hopkins. Sequential metamaterials with alternating poisson's ratios. *Nature Communications*, 13, 12 2022.
- [4] Corentin Coulais, Eial Teomy, Koen de Reus, Yair Shokef, and Martin van Hecke. Combinatorial design of textured mechanical metamaterials. *Nature*, 535:529–532, 7 2016.
- [5] Ruslan Guseinov, Connor McMahan, Jesús Pérez, Chiara Daraio, and Bernd Bickel. Programming temporal morphing of self-actuated shells. *Nature Communications*, 11, 12 2020.
- [6] Mina Konaković-Luković, Julian Panetta, Keenan Crane, and Mark Pauly. Rapid deployment of curved surfaces via programmable auxetics. *ACM Transactions on Graphics*, 37, 2018.
- [7] Babak Haghpanah, Ladan Salari-Sharif, Peyman Pourrajab, Jonathan Hopkins, and Lorenzo Valdevit. Multistable shape-reconfigurable architected materials. *Advanced Materials*, 28:7915–7920, 9 2016.
- [8] E. Hawkes, B. An, N. M. Benbernou, H. Tanaka, S. Kim, E. D. Demaine, D. Rus, and R. J. Wood. Programmable matter by folding. *Proceedings of the National Academy of Sciences of the United States of America*, 107:12441–12445, 7 2010.
- [9] Haigui Fan, Yuchen Tian, Lijuan Yang, Dapeng Hu, and Peiqi Liu. Multistable mechanical metamaterials with highly tunable strength and energy absorption performance. *Mechanics of Advanced Materials and Structures*, 2020.
- [10] Sicong Shan, Sung H. Kang, Jordan R. Raney, Pai Wang, Lichen Fang, Francisco Candido, Jennifer A. Lewis, and Katia Bertoldi. Multistable architected materials for trapping elastic strain energy. *Advanced Materials*, 27:4296–4301, 8 2015.
- [11] Zhiqiang Meng, Mingchao Liu, Yafei Zhang, and Chang Qing Chen. Multi-step deformation mechanical metamaterials. *Journal of the Mechanics and Physics of Solids*, 144, 11 2020.
- [12] Xianglong Yu, Ji Zhou, Haiyi Liang, Zhengyi Jiang, and Lingling Wu. Mechanical metamaterials associated with stiffness, rigidity and compressibility: A brief review. *Progress in Materials Science*, 94:114–173, 5 2018.
- [13] S Macrae Montgomery, Shuai Wu, Xiao Kuang, Connor D Armstrong, Cole Zemelka, Qiji Ze, Rundong Zhang, Ruike Zhao, H Jerry Qi, S M Montgomery, X Kuang, C D Armstrong, H J Qi, S Wu, C Zemelka, Q Ze, R Zhang, and R Zhao. Magneto-mechanical metamaterials with widely tunable mechanical properties and acoustic bandgaps. *Advanced Functional Materials*, 31:2005319, 1 2021.
- [14] Lucia M. Korpas, Rui Yin, Hiromi Yasuda, and Jordan R. Raney. Temperature-responsive multistable metamaterials. *ACS Applied Materials and Interfaces*, 13:31163–31170, 7 2021.
- [15] Charles El Helou, Philip R. Buskohl, Christopher E. Tabor, and Ryan L. Harne. Digital logic gates in soft, conductive mechanical metamaterials. *Nature Communications*, 12, 12 2021.
- [16] Xiaojun Tan, Shuai Chen, Bing Wang, Jie Tang, Lianchao Wang, Shaowei Zhu, Kaili Yao, and Peifei Xu. Real-time tunable negative stiffness mechanical metamaterial. *Extreme Mechanics Letters*, 41:100990, 11 2020.

- [17] Yifan Wang, Liuchi Li, Douglas Hofmann, José E. Andrade, and Chiara Daraio. Structured fabrics with tunable mechanical properties. *Nature* 2021 596:7871, 596:238–243, 8 2021.
- [18] Hiromi Yasuda, Philip R. Buskohl, Andrew Gillman, Todd D. Murphey, Susan Stepney, Richard A. Vaia, and Jordan R. Raney. Mechanical computing. *Nature*, 598:39–48, 10 2021.
- [19] Tian Chen, Mark Pauly, and Pedro M. Reis. A reprogrammable mechanical metamaterial with stable memory. *Nature*, 589:386–390, 1 2021.
- [20] Menachem Stern, Daniel Hexner, Jason W. Rocks, and Andrea J. Liu. Supervised learning in physical networks: From machine learning to learning machines. *Physical Review X*, 11:021045, 6 2021.
- [21] Zhiqiang Meng, Hujie Yan, Mingchao Liu, Wenkai Qin, Guy M. Genin, and Chang Qing Chen. Encoding and storage of information in mechanical metamaterials. *Advanced Science*, 4 2023.
- [22] Zili Tang, Qi Tao, Zehua Lan, Zong Liu, Toshiyuki Tsuchiya, Xiaohong Wang, and Man Wong. High durability latch-lock bi-stable switch on y-shaped cantilever. In *2021 21st International Conference on Solid-State Sensors, Actuators and Microsystems (Transducers)*, pages 675–678, 2021.
- [23] Yang Gao, Toshiki Ema, Zeyuan Cao, Sheng Ni, Elena Y.L. Chan, Osamu Tabata, Toshiyuki Tsuchiya, Xiaohong Wang, and Man Wong. A planar single-actuator bi-stable switch based on latch-lock mechanism. *2019 20th International Conference on Solid-State Sensors, Actuators and Microsystems and Eurosensors XXXIII, TRANSDUCERS 2019 and EUROSENSORS XXXIII*, pages 705–708, 6 2019.
- [24] Zhenci Sun, Wei Bian, and Jiahao Zhao. A zero static power consumption bi-stable rf mems switch based on inertial generated timing sequence method. *Microsystem Technologies*, 28:973–984, 4 2022.
- [25] Hen-Wei Huang and Yao-Joe Yang. A mems bistable device with push-on–push-off capability. *Journal of Microelectromechanical Systems*, 22(1):7–9, 2013.
- [26] Hen-Wei Huang, Fu-Wei Lee, and Yao-Joe Joseph Yang. Design criteria for a push-on push-off mems bistable device. *Journal of Microelectromechanical Systems*, 25(5):900–908, 2016.
- [27] Shin Wei Huang, Fan Chi Lin, and Yao Joe Yang. A novel single-actuator bistable microdevice with a moment-driven mechanism. *Sensors and Actuators A: Physical*, 310:111934, 8 2020.
- [28] Lennard J. Kwakernaak and Martin van Hecke. Counting and sequential information processing in mechanical metamaterials. <http://arxiv.org/abs/2302.06947>, 2 2023.
- [29] Martin van Hecke. Talk about metamaterial research at leiden university. <https://www.youtube.com/watch?v=0W2JZhZJzAc&t=3171s>, 2021.
- [30] Hussein Hussein, Patrice Moal, Gilles Bourbon, Yassine Haddab, Philippe Lutz, Philippe Lutz Modeling, and Patrice Le Moal. Analysis of a pre-shaped curved beam : Influence of high modes of buckling, 2015.
- [31] A. Amor, A. Fernandes, and J. Pouget. Numerical and experimental investigations of bistable beam snapping using distributed laplace force. *Meccanica*, 57:109–119, 1 2022.
- [32] B. Camescasse, A. Fernandes, and J. Pouget. Bistable buckled beam: Elastica modeling and analysis of static actuation. *International Journal of Solids and Structures*, 50:2881–2893, 9 2013.
- [33] Wenzhong Yan, Yunchen Yu, and A. Mehta. Analytical modeling for rapid design of bistable buckled beams. *Theoretical and Applied Mechanics Letters*, 9:264–272, 7 2019.
- [34] *Lagrange Multiplier*, pages 292–294. Springer New York, New York, NY, 2008.
- [35] Craig Lusk. *Using Pseudo-Rigid Body Models*, chapter 5, pages 55–76. John Wiley Sons, Ltd, 2013.

A

Analytical Derivation

In this section the relations between the angles of the vector loops are derived. First the bottom beam relations are derived, followed by the link and ending with the top beam.

A.1. Bottom beam

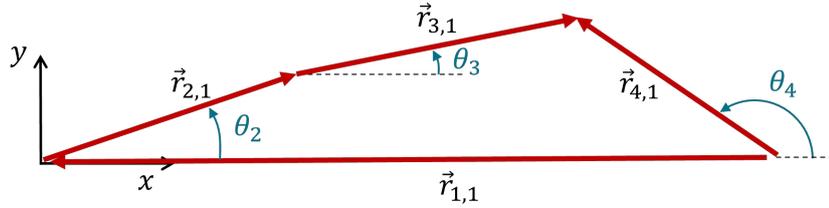


Figure A.1: Vector loop diagram

The sum of the vectors in the loop is set equal to zero

$$\vec{r}_{2,1} + \vec{r}_{3,1} - \vec{r}_{4,1} + \vec{r}_{1,1} = 0 \quad (\text{A.1})$$

Then each vector in the loop is expressed in its components and we can split these into two equations

$$L_2 \cos \theta_2 + L_3 \cos \theta_3 - L_4 \cos \theta_4 + L_1 \cos \theta_1 = 0 \quad (\text{A.2})$$

$$L_2 \sin \theta_2 + L_3 \sin \theta_3 - L_4 \sin \theta_4 + L_1 \sin \theta_1 = 0 \quad (\text{A.3})$$

In the given Figure A.1 θ_1 is equal to π . This value is filled in and the equation is rewritten in to isolate θ_3

$$L_2 \cos \theta_2 + L_3 \cos \theta_3 - L_4 \cos \theta_4 + L_1 \cos \pi = 0 \quad (\text{A.4})$$

$$L_2 \sin \theta_2 + L_3 \sin \theta_3 - L_4 \sin \theta_4 + L_1 \sin \pi = 0 \quad (\text{A.5})$$

$$L_3 \cos \theta_3 = L_4 \cos \theta_4 - L_2 \cos \theta_2 + L_1 \quad (\text{A.6})$$

$$L_3 \sin \theta_3 = L_4 \sin \theta_4 - L_2 \sin \theta_2 \quad (\text{A.7})$$

In the next step both equations are squared and added together

$$L_3^2 = (L_4 \cos \theta_4 - L_2 \cos \theta_2 + L_1)^2 + (L_4 \sin \theta_4 - L_2 \sin \theta_2)^2 \quad (\text{A.8})$$

In the next steps the equation will be expanded.

$$\begin{aligned} L_3^2 &= L_4^2 \cos^2 \theta_4 + L_2^2 \cos^2 \theta_2 + L_1^2 + L_4^2 \sin^2 \theta_4 + L_2^2 \sin^2 \theta_2 - 2L_2L_4 (\cos \theta_2 \cos \theta_4 + \sin \theta_2 \sin \theta_4) - 2L_1L_2 \cos \theta_2 + 2L_1L_4 \cos \theta_4 \\ &= L_4^2 + L_2^2 + L_1^2 - 2L_1L_2 \cos \theta_2 + 2L_1L_4 \cos \theta_4 - 2L_2L_4 (\cos \theta_2 \cos \theta_4 + \sin \theta_2 \sin \theta_4) \\ 0 &= \frac{L_4^2 + L_2^2 + L_1^2 - L_3^2}{2L_2L_4} - \frac{L_1}{L_2} \cos \theta_4 + \frac{L_1}{L_4} \cos \theta_2 - (\cos \theta_2 \cos \theta_4 + \sin \theta_2 \sin \theta_4) \end{aligned}$$

Constants K_1, K_2 and K_3 are introduced to simplify the equation

$$0 = K_1 + K_2 \cos \theta_4 - K_3 \cos \theta_2 - (\cos \theta_2 \cos \theta_4 + \sin \theta_2 \sin \theta_4) \quad (\text{A.9})$$

$$K_1 = \frac{L_4^2 + L_2^2 + L_1^2 - L_3^2}{2L_2L_4} \quad K_2 = \frac{L_1}{L_2} \quad K_3 = \frac{L_1}{L_4} \quad (\text{A.10})$$

In the next step we use the tangent of a half angle to rewrite the equation into a solvable second degree polynomial equation.

$$t = \tan \frac{\theta_4}{2}, \quad \sin \theta_4 = \frac{2t}{1+t^2}, \quad \cos \theta_4 = \frac{1-t^2}{1+t^2} \quad (\text{A.11})$$

$$t^2 (k_1 - k_2 - k_3 \cos \theta_2 + \cos \theta_2) + 2t (-\sin \theta_2) + (k_1 + k_2 - k_3 \cos \theta_2 - \cos \theta_2) \quad (\text{A.12})$$

$$At^2 + 2Bt + C = 0 \quad (\text{A.13})$$

This results in the following solution for θ_4 if θ_2

$$t = \frac{-B \pm \sqrt{B^2 - AC}}{A}, \quad \text{with} \quad \begin{cases} A = k_1 - k_2 - k_3 \cos \theta_2 + \cos \theta_2 \\ B = -\sin \theta_2 \\ C = k_1 + k_2 - k_3 \cos \theta_2 - \cos \theta_2 \end{cases} \quad (\text{A.14})$$

The same process is done again by starting at Equation A.6 and reordering the equations to isolate θ_4 .

$$L_4 \cos \theta_4 = L_3 \cos \theta_3 + L_2 \cos \theta_2 - L_1 \quad (\text{A.15})$$

$$L_4 \sin \theta_4 = L_3 \sin \theta_3 + L_2 \sin \theta_2 \quad (\text{A.16})$$

In the next step both equations are squared and added together

$$L_4^2 = (L_3 \cos \theta_3 + L_2 \cos \theta_2 - L_1)^2 + (L_3 \sin \theta_3 + L_2 \sin \theta_2)^2 \quad (\text{A.17})$$

In the next steps the equation will be expanded.

$$\begin{aligned} L_4^2 &= L_3^2 \cos^2 \theta_3 + L_2^2 \cos^2 \theta_2 + L_1^2 + L_3^2 \sin^2 \theta_3 + L_2^2 \sin^2 \theta_2 + 2L_2L_3 (\cos \theta_2 \cos \theta_3 + \sin \theta_2 \sin \theta_3) - 2L_1L_3 \cos \theta_3 - 2L_1L_2 \cos \theta_2 \\ &= L_3^2 + L_2^2 + L_1^2 - 2L_1L_2 \cos \theta_2 - 2L_1L_3 \cos \theta_3 + 2L_2L_3 (\cos \theta_2 \cos \theta_3 + \sin \theta_2 \sin \theta_3) \end{aligned}$$

$$0 = \frac{L_4^2 + L_2^2 + L_1^2 - L_3^2}{2L_2L_3} - \frac{L_1}{L_2} \cos \theta_3 - \frac{L_1}{L_3} \cos \theta_2 + (\cos \theta_2 \cos \theta_4 + \sin \theta_2 \sin \theta_4)$$

$$0 = K_4 - K_2 \cos \theta_3 - K_5 \cos \theta_2 + (\cos \theta_2 \cos \theta_3 + \sin \theta_2 \sin \theta_3) \quad (\text{A.18})$$

$$K_2 = \frac{L_1}{L_2} \quad K_4 = \frac{L_3^2 + L_2^2 + L_1^2 - L_4^2}{2L_2L_3} \quad K_5 = \frac{L_1}{L_3} \quad (\text{A.19})$$

$$t = \tan \frac{\theta_3}{2}, \quad \sin \theta_3 = \frac{2t}{1+t^2}, \quad \cos \theta_3 = \frac{1-t^2}{1+t^2} \quad (\text{A.20})$$

$$t^2 (k_4 + k_2 - k_5 \cos \theta_2 - \cos \theta_2) + 2t (\sin \theta_2) + (k_4 - k_2 - k_5 \cos \theta_2 + \cos \theta_2) \quad (\text{A.21})$$

$$At^2 + 2Bt + C = 0 \quad (\text{A.22})$$

This results in the following solution for θ_3 if θ_2 is known.

$$t = \frac{-B \pm \sqrt{B^2 - AC}}{A}, \quad \text{with} \quad \begin{cases} A = k_4 + k_2 - k_5 \cos \theta_2 - \cos \theta_2 \\ B = \sin \theta_2 \\ C = k_4 - k_2 - k_5 \cos \theta_2 + \cos \theta_2 \end{cases} \quad (\text{A.23})$$

A.2. Rigid Link

In figure Figure A.2 the vector loop of the rigid link is given.

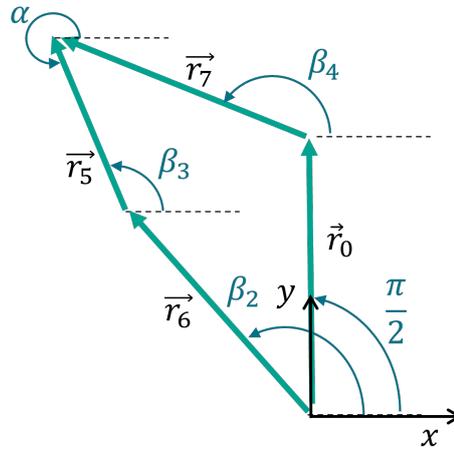


Figure A.2: Vector loop diagram

The sum of the vectors in the loop is set equal to zero

$$\vec{r}_6 + \vec{r}_5 - \vec{r}_7 - \vec{r}_0 = 0 \quad (\text{A.24})$$

Then each vector in the loop is expressed in its components and we can split these into two equations

$$L_6 \cos \beta_2 + L_5 \cos \beta_3 - L_7 \cos \beta_4 - L_0 \cos \frac{\pi}{2} = 0 \quad (\text{A.25})$$

$$L_6 \sin \beta_2 + L_5 \sin \beta_3 - L_7 \sin \beta_4 - L_0 \sin \frac{\pi}{2} = 0 \quad (\text{A.26})$$

Rewrite the equation to isolate β_3

$$L_5 \cos \beta_3 = L_7 \cos \beta_4 - L_6 \cos \beta_2 \quad (\text{A.27})$$

$$L_5 \sin \beta_3 = L_7 \sin \beta_4 - L_6 \sin \beta_2 + L_0 \quad (\text{A.28})$$

In the next step both equations are squared and added together

$$L_5^2 = (L_7 \cos \beta_4 - L_6 \cos \beta_2)^2 + (L_7 \sin \beta_4 - L_6 \sin \beta_2 + L_0)^2 \quad (\text{A.29})$$

In the next steps the equation will be expanded.

$$\begin{aligned} L_5^2 &= L_7^2 \cos^2 \beta_4 + L_6^2 \cos^2 \beta_2 + L_7^2 \sin^2 \beta_4 + L_6^2 \sin^2 \beta_2 + L_0^2 - 2L_6L_7 (\cos \beta_2 \cos \beta_4 + \sin \beta_2 \sin \beta_4) + 2L_0L_7 \sin \beta_4 - 2L_0L_6 \sin \beta_2 \\ &= L_7^2 + L_6^2 + L_0^2 - 2L_6L_7 (\cos \beta_2 \cos \beta_4 + \sin \beta_2 \sin \beta_4) + 2L_0L_7 \sin \beta_4 - 2L_0L_6 \sin \beta_2 \\ 0 &= \frac{L_7^2 + L_6^2 + L_0^2 - L_5^2}{2L_6L_7} - \frac{L_0}{L_6} \sin \beta_4 - \frac{L_0}{L_7} \sin \beta_2 + (\cos \beta_2 \cos \beta_4 + \sin \beta_2 \sin \beta_4) \end{aligned}$$

$$0 = K_9 - K_8 \sin \beta_2 + K_7 \sin \beta_4 - \cos \beta_2 \cos \beta_4 + \sin \beta_2 \sin \beta_4 \quad (\text{A.30})$$

$$t = \tan \frac{\beta_4}{2}, \quad \sin \beta_4 = \frac{2t}{1+t^2}, \quad \cos \beta_4 = \frac{1-t^2}{1+t^2} \quad (\text{A.31})$$

$$0 = K_9 - K_8 \sin \beta_2 + K_7 \frac{2t}{1+t^2} - \cos \beta_2 \frac{1-t^2}{1+t^2} + \sin \beta_2 \frac{2t}{1+t^2} \quad (\text{A.32})$$

$$t^2 (K_9 - K_8 \sin \beta_2 + \cos \beta_2) + 2t (K_7 - \sin \beta_2) + (K_9 - K_8 \sin \beta_2 - \cos \beta_2) \quad (\text{A.33})$$

$$At^2 + 2Bt + C = 0 \quad (\text{A.34})$$

This results in the following solution for β_4 if β_2 is known.

$$t = \frac{-B \pm \sqrt{B^2 - AC}}{A}, \text{ with } \begin{cases} A = K_9 - K_8 \sin \beta_2 + \cos \beta_2 \\ B = K_7 - \sin \beta_2 \\ C = K_9 - K_8 \sin \beta_2 - \cos \beta_2 \end{cases} \quad (\text{A.35})$$

Now starting with the same vector loop and isolating β_4 to retrieve a relation for β_4 in terms of β_2

$$L_7^2 = (L_5 \cos \beta_3 + L_6 \cos \beta_2)^2 + (L_5 \sin \beta_3 + L_6 \sin \beta_2 - L_0)^2 \quad (\text{A.36})$$

$$0 = \frac{L_5^2 + L_6^2 + L_0^2 - L_7^2}{2L_5L_6} - \frac{L_0}{L_5} \sin \beta_2 - \frac{L_0}{L_6} \sin \beta_3 + (\cos \beta_2 \cos \beta_3 + \sin \beta_2 \sin \beta_3) \quad (\text{A.37})$$

$$0 = K_{11} - K_{10} \sin \beta_2 - K_7 \sin \beta_3 + \cos \beta_2 \cos \beta_3 + \sin \beta_2 \sin \beta_3 \quad (\text{A.38})$$

$$t^2 (K_{11} - K_{10} \sin \beta_2 - \cos \beta_2) + 2t (K_7 - \sin \beta_2) + (K_{11} - K_{10} \sin \beta_2 - \cos \beta_2) \quad (\text{A.39})$$

$$At^2 + 2Bt + C = 0 \quad (\text{A.40})$$

This results in the following solution for β_4 if β_2 is known.

$$t = \frac{-B \pm \sqrt{B^2 - AC}}{A}, \text{ with } \begin{cases} A = K_{11} - K_{10} \sin \beta_2 - \cos \beta_2 \\ B = -K_7 + \sin \beta_2 \\ C = K_{11} - K_{10} \sin \beta_2 - \cos \beta_2 \end{cases} \quad (\text{A.41})$$

A.3. Top beam

In figure Figure A.3 the vector loop of the top beam is given.

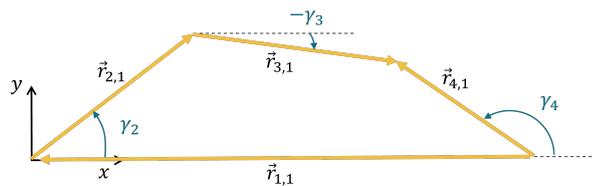


Figure A.3: Vector loop diagram

The sum of the vectors in the loop is set equal to zero

$$\vec{r}_{2,2} + \vec{r}_{3,2} - \vec{r}_{4,2} + \vec{r}_{1,2} = 0 \quad (\text{A.42})$$

$$L_2 \cos \gamma_2 + L_3 \cos \gamma_3 - L_4 \cos \gamma_4 + L_1 \cos \gamma_1 = 0 \quad (\text{A.43})$$

$$L_2 \sin \gamma_2 + L_3 \sin \gamma_3 - L_4 \sin \gamma_4 + L_1 \sin \gamma_1 = 0 \quad (\text{A.44})$$

$$L_2 \cos \gamma_2 + L_3 \cos \gamma_3 - L_4 \cos \gamma_4 + L_1 \cos \pi = 0 \quad (\text{A.45})$$

$$L_2 \sin \gamma_2 + L_3 \sin \gamma_3 - L_4 \sin \gamma_4 + L_1 \sin \pi = 0 \quad (\text{A.46})$$

$$L_3 \cos \gamma_3 = L_4 \cos \gamma_4 - L_2 \cos \gamma_2 + L_1 \quad (\text{A.47})$$

$$L_3 \sin \gamma_3 = L_4 \sin \gamma_4 - L_2 \sin \gamma_2 \quad (\text{A.48})$$

$$L_3^2 = (L_4 \cos \gamma_4 - L_2 \cos \gamma_2 + L_1)^2 + (L_4 \sin \gamma_4 - L_2 \sin \gamma_2)^2 \quad (\text{A.49})$$

$$0 = K_1 + K_2 \cos \gamma_4 - K_3 \cos \gamma_2 - (\cos \gamma_2 \cos \gamma_4 + \sin \gamma_2 \sin \gamma_4) \quad (\text{A.50})$$

$$K_1 = \frac{L_4^2 + L_2^2 + L_1^2 - L_3^2}{2L_2L_4} \quad K_2 = \frac{L_1}{L_2} \quad K_3 = \frac{L_1}{L_4} \quad (\text{A.51})$$

$$t = \tan \frac{\gamma_2}{2}, \quad \sin \gamma_2 = \frac{2t}{1+t^2}, \quad \cos \gamma_2 = \frac{1-t^2}{1+t^2} \quad (\text{A.52})$$

$$t^2 (k_1 + k_2 \cos \gamma_4 + k_3 + \cos \gamma_4) + 2t (-\sin \gamma_2) + (k_1 + k_2 \cos \gamma_4 - k_3 - \cos \gamma_4) \quad (\text{A.53})$$

$$At^2 + 2Bt + C = 0 \quad (\text{A.54})$$

$$t = \frac{-B \pm \sqrt{B^2 - AC}}{A}, \quad \text{with} \quad \begin{cases} A = k_1 + k_2 \cos \gamma_4 + k_3 + \cos \gamma_4 \\ B = -\sin \gamma_2 \\ C = k_1 + k_2 \cos \gamma_4 - k_3 - \cos \gamma_4 \end{cases} \quad (\text{A.55})$$

$$L_2 \cos \gamma_2 = L_4 \cos \gamma_4 - L_3 \cos \gamma_3 + L_1 \quad (\text{A.56})$$

$$L_2 \sin \gamma_2 = L_4 \sin \gamma_4 - L_3 \sin \gamma_3 \quad (\text{A.57})$$

$$L_3^2 = (L_4 \cos \gamma_4 - L_3 \cos \gamma_3 + L_1)^2 + (L_4 \sin \gamma_4 - L_3 \sin \gamma_3)^2 \quad (\text{A.58})$$

$$L_3^2 = L_4^2 \cos^2 \gamma_4 + L_3^2 \cos^2 \gamma_3 + L_1^2 + L_4^2 \sin^2 \gamma_4 + L_3^2 \sin^2 \gamma_3 - 2L_3L_4 (\cos \gamma_3 \cos \gamma_4 + \sin \gamma_3 \sin \gamma_4) - 2L_1L_3 \cos \gamma_3 + 2L_1L_4 \cos \gamma_4$$

$$= L_4^2 + L_3^2 + L_1^2 - 2L_3L_4 (\cos \gamma_3 \cos \gamma_4 + \sin \gamma_3 \sin \gamma_4) - 2L_1L_3 \cos \gamma_3 + 2L_1L_4 \cos \gamma_4$$

$$0 = \frac{L_4^2 + L_3^2 + L_1^2 - L_2^2}{2L_3L_4} - \frac{L_1}{L_4} \cos \gamma_3 + \frac{L_1}{L_3} \cos \gamma_4 - (\cos \gamma_3 \cos \gamma_4 + \sin \gamma_3 \sin \gamma_4)$$

$$0 = K_6 - K_3 \cos \gamma_3 + K_5 \cos \gamma_4 - (\cos \gamma_3 \cos \gamma_4 + \sin \gamma_3 \sin \gamma_4) \quad (\text{A.59})$$

$$K_3 = \frac{L_1}{L_4} \quad K_5 = \frac{L_1}{L_3} \quad K_6 = \frac{L_4^2 + L_3^2 + L_1^2 - L_2^2}{2L_3L_4} \quad (\text{A.60})$$

$$t = \tan \frac{\gamma_3}{2}, \quad \sin \gamma_3 = \frac{2t}{1+t^2}, \quad \cos \gamma_3 = \frac{1-t^2}{1+t^2} \quad (\text{A.61})$$

$$t^2 (k_6 + k_3 - k_5 \cos \gamma_4 + \cos \gamma_4) + 2t (-\sin \gamma_4) + (k_6 - k_3 + k_5 \cos \gamma_4 - \cos \gamma_4) \quad (\text{A.62})$$

$$At^2 + 2Bt + C = 0 \quad (\text{A.63})$$

$$t = \frac{-B \pm \sqrt{B^2 - AC}}{A}, \quad \text{with} \quad \begin{cases} A = k_6 + k_3 - k_5 \cos \gamma_4 + \cos \gamma_4 \\ B = -\sin \gamma_4 \\ C = k_6 - k_3 + k_5 \cos \gamma_4 - \cos \gamma_4 \end{cases} \quad (\text{A.64})$$

B

MATLAB scripts

In this Appendix the MATLAB code used for the analytic derivation is presented. This includes the actuation of the top beam and the bottom beam. This is performed in separate scripts due to the fact that both actuation paths require a different set of functions. To run the script for the top beam actuation the files and functions required are

B.1. Parameters

```
1 %% Parameters
2 preload = 1e-3; % preload displacement of the mechanism
3 gamma=0.8517;
4 Ktheta=2.65;
5 l = 6e-3; % length of flexure
6 w=8e-3; % width of in plane rigid elements
7 Link=60e-3+w; % Distance between the two beams
8 L2 = 20e-3; % length beam 1
9 L3 = 10e-3; % length beam 2
10 L4 = 40e-3; % length beam 3
11 L1 = L2+L3+L4+3*l-preload; % Effective length after preload
12
13 thickness = 0.9e-3; % thickness of flexure
14 thickness2 = 1e-3; % thickness of Link
15 width = 5e-3; % width of flexure
16 width_b = 5e-3; % Out of plane width beam
17 Ia = (1/12)*width*thickness^3; % Inertia of flexure
18 Ib = (1/12)*width*thickness2^3; % Inertia of Link
19 E=2.1e9; %Young's modulus PETG
20
21 % Notch hinge stiffnesses
22 constant=1; %for mode coupling constant=3 when all beams are equal length
23 Ka = E*Ia/l;
24 Kb = E*Ia/l;
25 Kc = E*Ia/l;
26 Kd = E*Ia/l;
27 Klink=2*gamma*Ktheta*E*Ib/Link;
28
29 % Lengths of rigid beams including the flexure lengths
30 r1=L1;
31 r2=L2+l;
32 r3=L3+l;
33 r4=L4+l;
34
35 con_r4=r4-l/2;%-thickness2/2; % Connection of Link
36 % Bottom and top beam
37 k1=(r1^2+r2^2-r3^2+r4^2)/(2*r2*r4);%k3
38 k2=r1/r2;%k1
39 k3=r1/r4;%k2
40 k4=(r1^2+r2^2+r3^2-r4^2)/(2*r2*r3);%k5
41 k5=r1/r3;%k4
```

```

42 k6=(r1^2-r2^2+r3^2+r4^2)/(2*r3*r4);
43
44 % Rigid link
45 r1_prime=Link;
46 r2_prime=sqrt(con_r4^2+((1-gamma)*(Link/2))^2);
47 r3_prime=gamma*Link;
48 r4_prime=sqrt(con_r4^2+((1-gamma)*(Link/2))^2);
49
50 k7=r1_prime/r2_prime;
51 k8=r1_prime/r4_prime;
52 k9=(r1_prime^2+r2_prime^2-r3_prime^2+r4_prime^2)/(2*r2_prime*r4_prime);
53 k10=r1_prime/r3_prime;
54 k11=(r1_prime^2+r2_prime^2+r3_prime^2-r4_prime^2)/(2*r2_prime*r3_prime);
55 k12=(r1_prime^2-r2_prime^2+r3_prime^2+r4_prime^2)/(2*r2_prime*r3_prime);
56
57 ta=atan((sqrt(-k3^2 + 2*k3*k4 + k5^2 - k4^2 - 2*k5 + 1)/(-1 + k5))/( -(k3 - k4)/(-1 +
58 k5)));% is this the equilibrium when r2 and r4 are equal?
59 tb=atan((-sqrt(-k3^2 + 2*k3*k4 + k5^2 - k4^2 - 2*k5 + 1)/(-1 + k5))/( -(k3 - k4)/(-1 +
60 k5)));
61 theta2_max=acos((2*r2^2+2*r2*r3+2*r2*r4-2*r2*preload-2*r3*preload-2*r4*preload+preload
62 ^2)/(2*r2*(r2+r3+r4-preload)))-0.001;% maximum geometric angle
63 g2max=theta2_max; % maximum geometric angle
64
65 %% Input hook
66
67 r2c=r2-8.75e-3;%8.75e-3
68 arm_a=5e-3+w/2; %5e-3
69 arm_b=19.25e-3;%19.25e-3
70 rb=sqrt((r2c-arm_b)^2+arm_a^2);
71 r2c_prime=sqrt(r2c^2+arm_a^2);
72 eps1=asin(arm_a/r2c_prime);
73 eps2=asin(arm_a/rb);

```

B.2. Top beam actuation

```

1 %% This file uses freudenstein to solve the geometry
2 % Then a multivariable problem is setup and the angles derived from the
3 % geometry are used in the total energy potential. V3 also uses the
4 % constraint equations
5
6 clear variables; close all; clc;
7 tic
8 %% Parameters
9 Parameters_three_loops;
10 addpath Geometric_relations;
11 addpath Other_functions;
12 syms theta2 gamma2 gamma4
13 n=100;
14 % start_stable_angle=-0.16335;%-ta
15 % ;%-0.1713095931494799209359676738055;%-0.1606458081052162256696647091303;
16 % angle_in=linspace(start_stable_angle,theta2_max,n); % Start near top equilibrium
17 % angle_in=linspace(-theta2_max,theta2_max,n); % For full range
18 dist=-1*(r2*sin(angle_in)-r2*sin(angle_in(1,1)));
19 x0=[0,0,pi,0,0,pi,pi/2,3*pi/2]; %Fabricated position
20 %x0=[ta,0,ta,ta,0,ta];
21
22 %% Derive geometry
23 Angle_C=atan((1-gamma)*(Link/2)/r4_prime);
24 %% Top beam
25 [gamma3,gamma4] = Top_beam_gamma2(gamma2,k1,k2,k3,k4,k5);
26
27 ga2(1:n,1)=angle_in; % Start in top equilibrium, zero force
28
29 for i=1:n
30 ga3(i,1)=eval(subs(gamma3(1,2),gamma2,ga2(i,1)));
31 ga4(i,1)=eval(subs(gamma4(1,2),gamma2,ga2(i,1)));
32 if ga4(i,1)<0
33 ga4(i,1)=2*pi+ga4(i,1);
34 end
35 end

```

```

35
36 %% Rigid link
37 clear gamma4
38 syms gamma4
39 beta4=gamma4+Angle_C; % In what configuration is the beam?
40 [beta2,beta3] = Rigid_Link_gamma2(beta4,k7,k8,k9,k10,k12);
41 for i=1:n
42 b2(i,1)=eval(subs(beta2(1,1),gamma4,ga4(i,1)));
43 b3(i,1)=eval(subs(beta3(1,1),gamma4,ga4(i,1)));
44 b4(i,1)=eval(subs(beta4(1,1),gamma4,ga4(i,1)));
45     if b2(i,1)<0
46         b2(i,1)=2*pi+b2(i,1);
47     end
48 end
49 al2=b3+pi; %needed for the energy solution
50 %% Bot beam
51 clear beta2
52 syms beta2
53 theta4=beta2+Angle_C;
54 [theta2,theta3] = Bottom_beam_gamma2(theta4,k1,k2,k3,k5,k6);
55 for i=1:n
56     t2g(i,1)=subs(theta2(1,2),beta2,b2(i,1));
57     t3g(i,1)=subs(theta3(1,2),beta2,b2(i,1));
58     t4g(i,1)=subs(theta4,beta2,b2(i,1));
59 end
60
61 %% Solve for minimum energy.
62 % Now we are interested in the force needed for this movement
63 % With the given initial guess we solve for the system
64 clear theta2 theta3 theta4 gamma2 gamma3 gamma4 beta3 alpha2 lambda1 lambda2 lambda3
65     lambda4 lambda5 lambda6 lambda7 angle
66 syms theta2 theta3 theta4 gamma2 gamma3 gamma4 beta3 alpha2 lambda1 lambda2 lambda3
67     lambda4 lambda5 lambda6 lambda7 angle
68 [Etot,Etot,Ebot,Elink] = Potential_energy(theta2,theta3,theta4, gamma2, gamma3, gamma4,
69     beta3, alpha2,x0,Ka,Kb,Kc,Kd,Klink);
70 [constraint1,constraint2,constraint3,constraint4,constraint5,constraint6,constraint7] =
71     Constraints_top_beam(r2,r3,r4,r4_prime,r3_prime,Link,theta2,theta3,theta4,gamma2,
72     gamma3,gamma4,Angle_C,beta3,preload, angle);
73 %% Initial configuration
74
75 const1=subs(constraint1, angle, angle_in(1,1));
76 const2=constraint2;
77 const3=constraint3;
78 const4=constraint4;
79 const5=constraint5;
80 const6=constraint6;
81 const7=constraint7;
82 Etotal=Etot(1,1);
83 [Equation_array] = Equations_top_beam(Etotal,theta2,theta3,theta4,gamma2,gamma3,gamma4,
84     beta3,alpha2,lambda1,lambda2,lambda3,lambda4,lambda5,lambda6,lambda7,const1,const2,
85     const3,const4,const5,const6,const7);
86
87 S=vpasolve(Equation_array,[theta2,theta3,theta4,gamma2,gamma3,gamma4,beta3,alpha2,
88     lambda1,lambda2,lambda3,lambda4,lambda5,lambda6,lambda7],[t2g(1,1),t3g(1,1),t4g
89     (1,1),ga2(1,1),ga3(1,1),ga4(1,1),b3(1,1),al2(1,1),0,0,0,0,0,0]);
90 t2(1,1)=S.theta2;
91 t3(1,1)=S.theta3;
92 t4(1,1)=S.theta4;
93 g2(1,1)=S.gamma2;
94 g3(1,1)=S.gamma3;
95 g4(1,1)=S.gamma4;
96 be3(1,1)=S.beta3;
97 alp2(1,1)=S.alpha2;
98 lamb1(1,1)=S.lambda1;
99 lamb2(1,1)=S.lambda2;
100 lamb3(1,1)=S.lambda3;
101 lamb4(1,1)=S.lambda4;
102 lamb5(1,1)=S.lambda5;
103 lamb6(1,1)=S.lambda6;
104 lamb7(1,1)=S.lambda7;

```

```

97 Et(1,1)=subs(Etotal,[theta2;theta3;theta4;gamma2;gamma3;gamma4;beta3;alpha2],[t2(1,1);
t3(1,1);t4(1,1);g2(1,1);g3(1,1);g4(1,1);be3(1,1);alp2(1,1)]);
98 Ebt(1,1)=subs(Ebot,[theta2;theta3;theta4;gamma2;gamma3;gamma4;beta3;alpha2],[t2(1,1);t3
(1,1);t4(1,1);g2(1,1);g3(1,1);g4(1,1);be3(1,1);alp2(1,1)]);
99 Etp(1,1)=subs(Etop,[theta2;theta3;theta4;gamma2;gamma3;gamma4;beta3;alpha2],[t2(1,1);t3
(1,1);t4(1,1);g2(1,1);g3(1,1);g4(1,1);be3(1,1);alp2(1,1)]);
100 El(1,1)=subs(Elink,[theta2;theta3;theta4;gamma2;gamma3;gamma4;beta3;alpha2],[t2(1,1);t3
(1,1);t4(1,1);g2(1,1);g3(1,1);g4(1,1);be3(1,1);alp2(1,1)]);
101
102 %% Sweep for the rest of the angles
103 % With the known single DOF movement the potential energy is solved for the entire
range of motion
104 for i=2:n-1
105     const1=subs(constraint1,angle,angle_in(i));
106
107     [Equation_array] = Equations_top_beam(Etotal,theta2,theta3,theta4,gamma2,gamma3,
gamma4,beta3,alpha2,lambda1,lambda2,lambda3,lambda4,lambda5,lambda6,lambda7,const1,
const2,const3,const4,const5,const6,const7);
108
109     S=vpasolve(Equation_array,[theta2,theta3,theta4,gamma2,gamma3,gamma4,beta3,alpha2,
lambda1,lambda2,lambda3,lambda4,lambda5,lambda6,lambda7],[t2g(i,1),t3g(i,1),t4g(i
,1),ga2(i,1),ga3(i,1),ga4(i,1),b3(i,1),a12(i,1),lamb1(i-1,1),lamb2(i-1,1),lamb3(i
-1,1),lamb4(i-1,1),lamb5(i-1,1),lamb6(i-1,1),lamb7(i-1,1)]);
110     t2(i,1)=S.theta2;
111     t3(i,1)=S.theta3;
112     t4(i,1)=S.theta4;
113     g2(i,1)=S.gamma2;
114     g3(i,1)=S.gamma3;
115     g4(i,1)=S.gamma4;
116     be3(i,1)=S.beta3;
117     alp2(i,1)=S.alpha2;
118     lamb1(i,1)=S.lambda1;
119     lamb2(i,1)=S.lambda2;
120     lamb3(i,1)=S.lambda3;
121     lamb4(i,1)=S.lambda4;
122     lamb5(i,1)=S.lambda5;
123     lamb6(i,1)=S.lambda6;
124     lamb7(i,1)=S.lambda7;
125     Et(i,1)=subs(Etotal,[theta2;theta3;theta4;gamma2;gamma3;gamma4;beta3;alpha2],[t2(i
,1);t3(i,1);t4(i,1);g2(i,1);g3(i,1);g4(i,1);be3(i,1);alp2(i,1)]);
126     Ebt(i,1)=subs(Ebot,[theta2;theta3;theta4;gamma2;gamma3;gamma4;beta3;alpha2],[t2(i
,1);t3(i,1);t4(i,1);g2(i,1);g3(i,1);g4(i,1);be3(i,1);alp2(i,1)]);
127     Etp(i,1)=subs(Etop,[theta2;theta3;theta4;gamma2;gamma3;gamma4;beta3;alpha2],[t2(i
,1);t3(i,1);t4(i,1);g2(i,1);g3(i,1);g4(i,1);be3(i,1);alp2(i,1)]);
128     El(i,1)=subs(Elink,[theta2;theta3;theta4;gamma2;gamma3;gamma4;beta3;alpha2],[t2(i
,1);t3(i,1);t4(i,1);g2(i,1);g3(i,1);g4(i,1);be3(i,1);alp2(i,1)]);
129
130     D_Etop(i,1)=Etp(i,1)-Etp(i-1,1);
131     D_Ebot(i,1)=Ebt(i,1)-Ebt(i-1,1);
132     D_Etot(i,1)= Et(i,1)- Et(i-1,1);
133
134     F_in(i,1)=-lamb1(i-1,1)/((r2-1/2)*cos(g2(i-1,1))); % minus sign because this gives
the reaction force.
135     D_in(i,1)=(r2-1/2)*sin(g2(i-1,1));
136     F_in2(i,1)=-lamb1(i-1,1)/(rb*sin(g2(i-1,1)-eps2)); % Force with input hooks
137     D_in2(i,1)=rb*cos(g2(i-1,1)-eps2); % Displacement with input hooks
138
139 end
140 toc
141 %% Last step
142 tic
143 for i=n
144     [Equation_array] = Equations_top_beam_v2(Etotal,theta2,theta3,theta4,gamma2,gamma3,
gamma4,beta3,alpha2,lambda2,lambda3,lambda4,lambda5,lambda6,lambda7,const2,const3,
const4,const5,const6,const7);
145     range=[theta2_max/2, theta2_max];
146     GUESS=[range;[theta2_max,-theta2_max];[pi-theta2_max,pi];range;[theta2_max,-
theta2_max];[pi-theta2_max,pi];[-pi,pi];[pi,2*pi];[NaN,NaN];[NaN,NaN];[NaN,NaN]
];[NaN,NaN];[NaN,NaN];[NaN,NaN]];
147
148     if D_Etot(i-1)<0 && D_Etop(i-1)<0 && D_Ebot(i-1)<0

```

```

149 fprintf('SNAP\n')
150 S=vpasolve(Equation_array,[theta2,theta3,theta4,gamma2,gamma3,gamma4,beta3,alpha2,
lambda2,lambda3,lambda4,lambda5,lambda6,lambda7],GUESS);
151 else
152 fprintf('NO SNAP\n')
153 S=vpasolve(Equation_array,[theta2,theta3,theta4,gamma2,gamma3,gamma4,beta3,alpha2,
lambda2,lambda3,lambda4,lambda5,lambda6,lambda7],[t2g(i,1),t3g(i,1),t4g(i,1),ga2(i,
1),ga3(i,1),ga4(i,1),b3(i,1),al2(i,2),lamb2(i-1,1),lamb3(i-1,1),lamb4(i-1,1),lamb5
(i-1,1),lamb6(i-1,1),lamb7(i-1,1)]);
154 end
155
156 t2(i,1)=S.theta2;
157 t3(i,1)=S.theta3;
158 t4(i,1)=S.theta4;
159 g2(i,1)=S.gamma2;
160 g3(i,1)=S.gamma3;
161 g4(i,1)=S.gamma4;
162 be3(i,1)=S.beta3;
163 alp2(i,1)=S.alpha2;
164 lamb2(i,1)=S.lambda2;
165 lamb3(i,1)=S.lambda3;
166 lamb4(i,1)=S.lambda4;
167 lamb5(i,1)=S.lambda5;
168 lamb6(i,1)=S.lambda6;
169 lamb7(i,1)=S.lambda7;
170 Et(i,1)=subs(Etotal,[theta2;theta3;theta4;gamma2;gamma3;gamma4;beta3;alpha2],[t2(i,
1);t3(i,1);t4(i,1);g2(i,1);g3(i,1);g4(i,1);be3(i,1);alp2(i,1)]);
171 Ebt(i,1)=subs(Ebot,[theta2;theta3;theta4;gamma2;gamma3;gamma4;beta3;alpha2],[t2(i,
1);t3(i,1);t4(i,1);g2(i,1);g3(i,1);g4(i,1);be3(i,1);alp2(i,1)]);
172 Etp(i,1)=subs(Etop,[theta2;theta3;theta4;gamma2;gamma3;gamma4;beta3;alpha2],[t2(i,
1);t3(i,1);t4(i,1);g2(i,1);g3(i,1);g4(i,1);be3(i,1);alp2(i,1)]);
173 El(i,1)=subs(Elink,[theta2;theta3;theta4;gamma2;gamma3;gamma4;beta3;alpha2],[t2(i,
1);t3(i,1);t4(i,1);g2(i,1);g3(i,1);g4(i,1);be3(i,1);alp2(i,1)]);
174 end
175
176 toc
177
178 %% Derive data points for the position
179
180 for k=1:n
181 %Bottom beam
182 a(k,1:2)=[0;0];
183 b(k,1:2)=r2*[cos(t2(k,1));sin(t2(k,1))];
184 c1(k,1:2)=[b(k,1)+r3*cos(t3(k,1));b(k,2)+r3*sin(t3(k,1))];
185 d1(k,1:2)=[c1(k,1)-r4*cos(t4(k,1));c1(k,2)-r4*sin(t4(k,1))];
186
187 %Rigid Link
188 s1(k,1:2)=[r1+con_r4*cos(t4(k,1));con_r4*sin(t4(k,1))];
189 s2(k,1:2)=[r1+r2_prime*cos(t4(k,1)-Angle_C);r2_prime*sin(t4(k,1)-Angle_C)];
190 s3(k,1:2)=[s2(k,1)+gamma*Link*cos(be3(k,1));s2(k,2)+gamma*Link*sin(be3(k,1))];
191 s4(k,1:2)=[r1+con_r4*cos(g4(k,1));Link+con_r4*sin(g4(k,1))];
192
193 % Input Hook Bottom
194 sin1(k,1:2)=[0;0];
195 sin2(k,1:2)=[r2c*cos(t2(k,1));r2c*sin(t2(k,1))];
196 sin3(k,1:2)=[r2c_prime*cos(t2(k,1)+eps1);r2c_prime*sin(t2(k,1)+eps1)];
197 sin4(k,1:2)=[rb*cos(t2(k,1)+eps2);rb*sin(t2(k,1)+eps2)];
198
199 % Input Hook Top
200 sin5(k,1:2)=[0;Link];
201 sin6(k,1:2)=[r2c*cos(g2(k,1));Link+r2c*sin(g2(k,1))];
202 sin7(k,1:2)=[r2c_prime*cos(g2(k,1)-eps1);Link+r2c_prime*sin(g2(k,1)-eps1)];
203 sin8(k,1:2)=[rb*cos(g2(k,1)-eps2);Link+rb*sin(g2(k,1)-eps2)];
204
205 %Top beam
206 e(k,1:2)=[0;Link];
207 f(k,1:2)=[r2*cos(g2(k,1));Link+r2*sin(g2(k,1))];
208 g1(k,1:2)=[f(k,1)+r3*cos(g3(k,1));f(k,2)+r3*sin(g3(k,1))];
209 h1(k,1:2)=[g1(k,1)-r4*cos(g4(k,1));g1(k,2)-r4*sin(g4(k,1))];
210
211 %Output relations

```

```

212 Output_top(k,1:2)=[f(k,1)+0.5*r3*cos(g3(k,1));f(k,2)+0.5*r3*sin(g3(k,1))];
213 Output_bot(k,1:2)=[b(k,1)+0.5*r3*cos(t3(k,1));b(k,2)+0.5*r3*sin(t3(k,1))];
214 end
215 %% Plotting!
216 load('Top_full_1mm')
217 F_T_1mm=F_in2;
218 D_T_1mm=D_in2;
219 load('Top_full_0_75mm')
220 F_T_0_75mm=F_in2;
221 D_T_0_75mm=D_in2;
222
223 figure(1) % Plot the F-D curve
224 hold on
225 plot(D_T_0_75mm(2:n-1)-D_T_0_75mm(2),F_T_0_75mm(2:n-1),'r-','Linewidth',2')
226 plot(D_T_1mm(2:n-1)-D_T_1mm(2),F_T_1mm(2:n-1),'b-','Linewidth',2')
227 axis([0 3.5e-3 -5 10])
228 grid on
229 xlabel('Displacement [m]')
230 ylabel('Force [N]')
231 title('F-d curve top beam actuation')
232 legend('0.75 mm preload','1 mm preload')
233
234 figure(2) % Plot the Potential energy
235 plot(D_in2(2:end)-D_in2(2),Et(2:length(D_in2)),D_in2(2:end)-D_in2(2),Ebt(2:length(D_in2
    )),D_in2(2:end)-D_in2(2),Etp(2:length(D_in2)),D_in2(2:end)-D_in2(2),El(2:length(
    D_in2)),'Linewidth',2)
236 legend('Etotal','Ebottom','Etop','Elink')
237 ylabel('E [J]')
238 xlabel('D_{in}')
239 title('Potential energy top beam actuation')
240 grid on
241
242 figure(3)
243 plot(D_in2(2:end,1)-D_in2(2),Output_top(2:n-1,2)-Output_top(2,2),'Linewidth',2')
244 grid on
245 ylabel('Output displacement [m]')
246 xlabel('Input displacement [m]')
247 title('Top beam actuation ')
248
249 %save("Top_full_0_75mm")
250 %save("Top_equilibrium_1mm")
251 %save("Top_full_1mm")

```

B.3. Bottom beam actuation

```

1 %% This file uses freudenstein to solve the geometry
2 % Then a multivariable problem is setup and the angles derived from the
3 % geometry are used in the total energy potential. This potential energy
4 % equation is solved with the Lagrange multiplier.
5
6 clear variables; close all; clc;
7 tic
8 %% Parameters
9 Parameters_three_loops;
10 addpath Geometric_relations;
11 addpath Other_functions;
12 syms theta2 gamma2 gamma4
13 n=100;
14 % start_stable_angle=0.16335;
15 % angle_in=linspace(start_stable_angle,theta2_max,n); % Start near top equilibrium
16 angle_in=linspace(theta2_max,-theta2_max,n); % For full range
17 dist=-1*(r2*sin(angle_in)-r2*sin(angle_in(1,1)));
18 x0=[0,0,pi,0,0,pi,pi/2,3*pi/2]; %Fabricated position
19 %x0=[ta,0,ta,ta,0,ta];
20
21 %% Derive geometry
22 % This is the angle between the vector loop of the rigid link and the top and bottom
    beam, can be positive or negative
23 Angle_C=atan((1-gamma)*(Link/2)/r4_prime);
24 %% Bot beam

```

```

25 [theta3,theta4] = Bottom_beam_theta2(theta2,k1,k2,k3,k4,k5);
26 t2g(1:n,1)=angle_in;
27 for i=1:n
28     t3g(i,1)=eval(subs(theta3(1,1),theta2,t2g(i,1)));
29     t4g(i,1)=eval(subs(theta4(1,1),theta2,t2g(i,1)));
30     if t4g(i,1)<0
31         t4g(i,1)=2*pi+t4g(i,1);
32     end
33 end
34
35 %% Rigid link
36 clear theta4
37 syms theta4
38 beta2=theta4-Angle_C; % In what configuration is the beam?
39 [beta3,beta4] = Rigid_Link_theta2(beta2,k7,k8,k9,k10,k11);
40 for i=1:n
41     b2(i,1)=eval(subs(beta2(1,1),theta4,t4g(i,1)));
42     b3(i,1)=eval(subs(beta3(1,1),theta4,t4g(i,1)));
43     b4(i,1)=eval(subs(beta4(1,1),theta4,t4g(i,1)));
44     if b4(i,1)<0
45         b4(i,1)=2*pi+b4(i,1);
46     end
47 end
48 a12=b3+pi; %needed for the energy solution
49
50 %% Top beam
51 clear beta4
52 syms beta4
53 gamma4=beta4-Angle_C;
54 [gamma2,gamma3] = Top_beam_theta2(gamma4,k1,k2,k3,k5,k6);
55
56 for i=1:n
57     ga2(i,1)=eval(subs(gamma2(1,1),beta4,b4(i,1)));
58     ga3(i,1)=eval(subs(gamma3(1,1),beta4,b4(i,1)));
59     ga4(i,1)=eval(subs(gamma4(1,1),beta4,b4(i,1)));
60 end
61
62
63 %% Solve for minimum energy.
64 % Now we are interested in the force needed for this movement
65 % With the given initial guess we solve for the system
66 clear theta2 theta3 theta4 gamma2 gamma3 gamma4 beta3 alpha2 lambda1 lambda2 lambda3
67     lambda4 lambda5 lambda6 lambda7 angle
68 syms theta2 theta3 theta4 gamma2 gamma3 gamma4 beta3 alpha2 lambda1 lambda2 lambda3
69     lambda4 lambda5 lambda6 lambda7 angle
70
71 [Etot,ETop,EBot,Elink] = Potential_energy(theta2,theta3,theta4, gamma2, gamma3, gamma4,
72     beta3,alpha2,x0,Ka,Kb,Kc,Kd,Klink);
73 [constraint1,constraint2,constraint3,constraint4,constraint5,constraint6,constraint7] =
74     Constraints_bot_beam(r2,r3,r4,r4_prime,r3_prime,Link,theta2,theta3,theta4,gamma2,
75     gamma3,gamma4,Angle_C,beta3,preload,angle);
76 %% Initial configuration
77
78 const1=subs(constraint1,angle,angle_in(1,1));
79 const2=constraint2;
80 const3=constraint3;
81 const4=constraint4;
82 const5=constraint5;
83 const6=constraint6;
84 const7=constraint7;
85 Etotal=Etot(1,1);
86 [Equation_array] = Equations_top_beam(Etotal,theta2,theta3,theta4,gamma2,gamma3,gamma4,
87     beta3,alpha2,lambda1,lambda2,lambda3,lambda4,lambda5,lambda6,lambda7,const1,const2,
88     const3,const4,const5,const6,const7);
89
90 S=vpasolve(Equation_array,[theta2,theta3,theta4,gamma2,gamma3,gamma4,beta3,alpha2,
91     lambda1,lambda2,lambda3,lambda4,lambda5,lambda6,lambda7],[t2g(1,1),t3g(1,1),t4g
92     (1,1),ga2(1,1),ga3(1,1),ga4(1,1),b3(1,1),a12(1,1),0,0,0,0,0,0]);
93 t2(1,1)=S.theta2;
94 t3(1,1)=S.theta3;
95 t4(1,1)=S.theta4;

```

```

87 g2(1,1)=S.gamma2;
88 g3(1,1)=S.gamma3;
89 g4(1,1)=S.gamma4;
90 be3(1,1)=S.beta3;
91 alp2(1,1)=S.alpha2;
92 lamb1(1,1)=S.lambda1;
93 lamb2(1,1)=S.lambda2;
94 lamb3(1,1)=S.lambda3;
95 lamb4(1,1)=S.lambda4;
96 lamb5(1,1)=S.lambda5;
97 lamb6(1,1)=S.lambda6;
98 lamb7(1,1)=S.lambda7;
99 Et(1,1)=subs(Etotal,[theta2;theta3;theta4;gamma2;gamma3;gamma4;beta3;alpha2],[t2(1,1);
    t3(1,1);t4(1,1);g2(1,1);g3(1,1);g4(1,1);be3(1,1);alp2(1,1)]);
100 Ebt(1,1)=subs(Ebot,[theta2;theta3;theta4;gamma2;gamma3;gamma4;beta3;alpha2],[t2(1,1);t3
    (1,1);t4(1,1);g2(1,1);g3(1,1);g4(1,1);be3(1,1);alp2(1,1)]);
101 Etp(1,1)=subs(Etop,[theta2;theta3;theta4;gamma2;gamma3;gamma4;beta3;alpha2],[t2(1,1);t3
    (1,1);t4(1,1);g2(1,1);g3(1,1);g4(1,1);be3(1,1);alp2(1,1)]);
102 El(1,1)=subs(Elink,[theta2;theta3;theta4;gamma2;gamma3;gamma4;beta3;alpha2],[t2(1,1);t3
    (1,1);t4(1,1);g2(1,1);g3(1,1);g4(1,1);be3(1,1);alp2(1,1)]);
103
104 %% Sweep for the rest of the angles
105 % With the known single DOF movement the potential energy is solved for the entire
    range of motion
106 for i=2:n-1
107     const1=subs(constraint1,angle,angle_in(i));
108
109     [Equation_array] = Equations_top_beam(Etotal,theta2,theta3,theta4,gamma2,gamma3,
    gamma4,beta3,alpha2,lambda1,lambda2,lambda3,lambda4,lambda5,lambda6,lambda7,const1,
    const2,const3,const4,const5,const6,const7);
110
111     S=vpasolve(Equation_array,[theta2,theta3,theta4,gamma2,gamma3,gamma4,beta3,alpha2,
    lambda1,lambda2,lambda3,lambda4,lambda5,lambda6,lambda7],[t2g(i,1),t3g(i,1),t4g(i
    ,1),ga2(i,1),ga3(i,1),ga4(i,1),b3(i,1),a12(i,1),lamb1(i-1,1),lamb2(i-1,1),lamb3(i
    -1,1),lamb4(i-1,1),lamb5(i-1,1),lamb6(i-1,1),lamb7(i-1,1)]);
112     t2(i,1)=S.theta2;
113     t3(i,1)=S.theta3;
114     t4(i,1)=S.theta4;
115     g2(i,1)=S.gamma2;
116     g3(i,1)=S.gamma3;
117     g4(i,1)=S.gamma4;
118     be3(i,1)=S.beta3;
119     alp2(i,1)=S.alpha2;
120     lamb1(i,1)=S.lambda1;
121     lamb2(i,1)=S.lambda2;
122     lamb3(i,1)=S.lambda3;
123     lamb4(i,1)=S.lambda4;
124     lamb5(i,1)=S.lambda5;
125     lamb6(i,1)=S.lambda6;
126     lamb7(i,1)=S.lambda7;
127     Et(i,1)=subs(Etotal,[theta2;theta3;theta4;gamma2;gamma3;gamma4;beta3;alpha2],[t2(i
    ,1);t3(i,1);t4(i,1);g2(i,1);g3(i,1);g4(i,1);be3(i,1);alp2(i,1)]);
128     Ebt(i,1)=subs(Ebot,[theta2;theta3;theta4;gamma2;gamma3;gamma4;beta3;alpha2],[t2(i
    ,1);t3(i,1);t4(i,1);g2(i,1);g3(i,1);g4(i,1);be3(i,1);alp2(i,1)]);
129     Etp(i,1)=subs(Etop,[theta2;theta3;theta4;gamma2;gamma3;gamma4;beta3;alpha2],[t2(i
    ,1);t3(i,1);t4(i,1);g2(i,1);g3(i,1);g4(i,1);be3(i,1);alp2(i,1)]);
130     El(i,1)=subs(Elink,[theta2;theta3;theta4;gamma2;gamma3;gamma4;beta3;alpha2],[t2(i
    ,1);t3(i,1);t4(i,1);g2(i,1);g3(i,1);g4(i,1);be3(i,1);alp2(i,1)]);
131
132     D_Etop(i,1)=Etp(i,1)-Etp(i-1,1);
133     D_Ebot(i,1)=Ebt(i,1)-Ebt(i-1,1);
134     D_Etot(i,1)= Et(i,1)- Et(i-1,1);
135
136     F_in(i,1)=-lamb1(i-1,1)/((r2-1/2)*cos(t2(i-1,1))); % minus sign because this gives
    the reaction force.
137     D_in(i,1)=(r2-1/2)*sin(t2(i-1,1));
138     F_in2(i,1)=-lamb1(i-1,1)/(rb*sin(t2(i-1,1)+eps2));
139     D_in2(i,1)=rb*cos(t2(i-1,1)+eps2); % Displacement with input hooks
140
141 end
142 toc

```

```

143 %% Last step
144 tic
145 for i=n
146     [Equation_array] = Equations_top_beam_v2(Etotal, theta2, theta3, theta4, gamma2, gamma3,
147     gamma4, beta3, alpha2, lambda2, lambda3, lambda4, lambda5, lambda6, lambda7, const2, const3,
148     const4, const5, const6, const7);
149     range=[-theta2_max/2, -theta2_max];
150     GUESS=[range; [theta2_max, -theta2_max]; [pi+theta2_max, pi]; range; [theta2_max, -
151     theta2_max]; [pi+theta2_max, pi]; [-pi, pi]; [pi, 2*pi]; [NaN, NaN]; [NaN, NaN]; [NaN, NaN
152     ]; [NaN, NaN]; [NaN, NaN]; [NaN, NaN]];
153     if D_Etot(i-1)<0 && D_Etop(i-1)<0 && D_Ebot(i-1)<0
154     S=vpasolve(Equation_array, [theta2, theta3, theta4, gamma2, gamma3, gamma4, beta3, alpha2,
155     lambda2, lambda3, lambda4, lambda5, lambda6, lambda7], GUESS);
156     fprintf('SNAP\n')
157     else
158     S=vpasolve(Equation_array, [theta2, theta3, theta4, gamma2, gamma3, gamma4, beta3, alpha2,
159     lambda2, lambda3, lambda4, lambda5, lambda6, lambda7], [t2g(i,1), t3g(i,1), t4g(i,1), ga2(i
160     ,1), ga3(i,1), ga4(i,1), b3(i,1), al2(i,1), lamb2(i-1,1), lamb3(i-1,1), lamb4(i-1,1), lamb5
161     (i-1,1), lamb6(i-1,1), lamb7(i-1,1)]);
162     fprintf('NO SNAP\n')
163     end
164     t2(i,1)=S.theta2;
165     t3(i,1)=S.theta3;
166     t4(i,1)=S.theta4;
167     g2(i,1)=S.gamma2;
168     g3(i,1)=S.gamma3;
169     g4(i,1)=S.gamma4;
170     be3(i,1)=S.beta3;
171     alp2(i,1)=S.alpha2;
172     lamb2(i,1)=S.lambda2;
173     lamb3(i,1)=S.lambda3;
174     lamb4(i,1)=S.lambda4;
175     lamb5(i,1)=S.lambda5;
176     lamb6(i,1)=S.lambda6;
177     lamb7(i,1)=S.lambda7;
178     Et(i,1)=subs(Etotal, [theta2; theta3; theta4; gamma2; gamma3; gamma4; beta3; alpha2], [t2(i
179     ,1); t3(i,1); t4(i,1); g2(i,1); g3(i,1); g4(i,1); be3(i,1); alp2(i,1)]);
180     Ebt(i,1)=subs(Ebot, [theta2; theta3; theta4; gamma2; gamma3; gamma4; beta3; alpha2], [t2(i
181     ,1); t3(i,1); t4(i,1); g2(i,1); g3(i,1); g4(i,1); be3(i,1); alp2(i,1)]);
182     Etp(i,1)=subs(Etop, [theta2; theta3; theta4; gamma2; gamma3; gamma4; beta3; alpha2], [t2(i
183     ,1); t3(i,1); t4(i,1); g2(i,1); g3(i,1); g4(i,1); be3(i,1); alp2(i,1)]);
184     El(i,1)=subs(Elink, [theta2; theta3; theta4; gamma2; gamma3; gamma4; beta3; alpha2], [t2(i
185     ,1); t3(i,1); t4(i,1); g2(i,1); g3(i,1); g4(i,1); be3(i,1); alp2(i,1)]);
186 end
187 toc
188 %% Derive data points for the position
189 for k=1:n
190     %%Bottom beam
191     a(k,1:2)=[0;0];
192     b(k,1:2)=r2*[cos(t2(k,1)); sin(t2(k,1))];
193     c1(k,1:2)=[b(k,1)+r3*cos(t3(k,1)); b(k,2)+r3*sin(t3(k,1))];
194     d1(k,1:2)=[c1(k,1)-r4*cos(t4(k,1)); c1(k,2)-r4*sin(t4(k,1))];
195     %%Rigid Link
196     s1(k,1:2)=[r1+con_r4*cos(t4(k,1)); con_r4*sin(t4(k,1))];
197     s2(k,1:2)=[r1+r2_prime*cos(t4(k,1)-Angle_C); r2_prime*sin(t4(k,1)-Angle_C)];
198     s3(k,1:2)=[s2(k,1)+gamma*Link*cos(be3(k,1)); s2(k,2)+gamma*Link*sin(be3(k,1))];
199     s4(k,1:2)=[r1+con_r4*cos(g4(k,1)); Link+con_r4*sin(g4(k,1))];
200     %% Input Hook Bottom
201     sin1(k,1:2)=[0;0];
202     sin2(k,1:2)=[r2c*cos(t2(k,1)); r2c*sin(t2(k,1))];
203     sin3(k,1:2)=[r2c_prime*cos(t2(k,1)+eps1); r2c_prime*sin(t2(k,1)+eps1)];
204     sin4(k,1:2)=[rb*cos(t2(k,1)+eps2); rb*sin(t2(k,1)+eps2)];
205     %% Input Hook Top

```

```

202 sin5(k,1:2)=[0;Link];
203 sin6(k,1:2)=[r2c*cos(g2(k,1));Link+r2c*sin(g2(k,1))];
204 sin7(k,1:2)=[r2c_prime*cos(g2(k,1)-eps1);Link+r2c_prime*sin(g2(k,1)-eps1)];
205 sin8(k,1:2)=[rb*cos(g2(k,1)-eps2);Link+rb*sin(g2(k,1)-eps2)];
206
207 %Top beam
208 e(k,1:2)=[0;Link];
209 f(k,1:2)=[r2*cos(g2(k,1));Link+r2*sin(g2(k,1))];
210 g1(k,1:2)=[f(k,1)+r3*cos(g3(k,1));f(k,2)+r3*sin(g3(k,1))];
211 h1(k,1:2)=[g1(k,1)-r4*cos(g4(k,1));g1(k,2)-r4*sin(g4(k,1))];
212
213 %Output relations
214 Output_top(k,1:2)=[f(k,1)+0.5*r3*cos(g3(k,1));f(k,2)+0.5*r3*sin(g3(k,1))];
215 Output_bot(k,1:2)=[b(k,1)+0.5*r3*cos(t3(k,1));b(k,2)+0.5*r3*sin(t3(k,1))];
216 end
217 %% Plotting!
218
219 figure(1) % Plot the F-D curve
220 plot(D_in2(2:n-1)-D_in2(2),F_in2(2:n-1),'r-','Linewidth',2)
221 axis([0 3.5e-3 -5 10])
222 grid on
223 xlabel('Displacement [m]')
224 ylabel('Force [N]')
225 title('F-d curve bottom beam actuation')
226
227 figure(2) % Plot the Potential energy
228 % plot(D_in2(2:end),Et(2:length(D_in2)),D_in2(2:end),Ebt(2:length(D_in2)),D_in2(2:end),
229 % Etp(2:length(D_in2)),D_in2(2:end),El(2:length(D_in2)),'Linewidth',2)
229 plot(D_in2(2:end)-D_in2(2),Et(2:length(D_in2)),D_in2(2:end)-D_in2(2),Ebt(2:length(D_in2)
230 ),D_in2(2:end)-D_in2(2),Etp(2:length(D_in2)),D_in2(2:end)-D_in2(2),El(2:length(
231 D_in2)), 'Linewidth',2)
230 legend('Etotal','Ebottom','Etop','Elink')
231 ylabel('E [J]')
232 xlabel('Displacement [m]')
233 title('Potential energy bottom beam actuation')
234 grid on
235
236 %save("Bottom_equilibrium_1mm")
237 %save("Bottom_full_1mm")
238 %save("Bottom_full_0_75mm")

```

B.4. Functions

```

1 function [Etot,Etop,Ebot,Elink] = Potential_energy(theta2,theta3,theta4, gamma2, gamma3
, gamma4,beta3,alpha2,x0,Ka,Kb,Kc,Kd,Klink)
2 %ENERGY_SINGLE_BEAM Summary of this function goes here
3 % Detailed explanation goes here
4
5 %Bottom beam
6 PHI1=theta2-x0(1);
7 PHI2=(theta3-x0(2))-(theta2-x0(1));
8 PHI3=(theta3-x0(2))-(theta4-x0(3));
9 PHI4=(theta4-x0(3));
10
11 %Top beam
12 PHI5=gamma2-x0(4);
13 PHI6=(gamma3-x0(5))-(gamma2-x0(4));
14 PHI7=(gamma3-x0(5))-(gamma4-x0(6));
15 PHI8=(gamma4-x0(6));
16
17 %Rigid Link
18 PHI9=beta3-x0(7);
19 PHI10=alpha2-x0(8);
20
21 %Energy
22 Ebot=0.5*Ka*PHI1^2 + 0.5*Kb*PHI2^2+ 0.5*Kc*PHI3^2+0.5*Kd*PHI4^2;
23 Etop=0.5*Ka*PHI5^2 + 0.5*Kb*PHI6^2+ 0.5*Kc*PHI7^2+0.5*Kd*PHI8^2;
24 Elink=0.5*Klink*PHI9^2+0.5*Klink*PHI10^2;
25 Etot=Ebot+Etop+Elink;
26

```

```
27 end
```

B.4.1. Constraints

```
1 function [constraint1,constraint2,constraint3,constraint4,constraint5,constraint6,
   constraint7] = Constraints_bot_beam(r2,r3,r4,r4_prime,r3_prime,Link,theta2,theta3,
   theta4,gamma2,gamma3,gamma4,Angle_C,beta3,preload,angle)
2 %CONSTRAINTS_ALL_VARIABLES Summary of this function goes here
3 % Detailed explanation goes here
4
5 constraint1=theta2-angle;
6 constraint2=r2*sin(theta2)+r3*sin(theta3)-r4*sin(theta4);
7 constraint3=r2*cos(theta2)+r3*cos(theta3)-r4*cos(theta4)-(r2+r3+r4-preload);
8 constraint4=r2*sin(gamma2)+r3*sin(gamma3)-r4*sin(gamma4);
9 constraint5=r2*cos(gamma2)+r3*cos(gamma3)-r4*cos(gamma4)-(r2+r3+r4-preload);
10 constraint6=r4_prime*sin(theta4-Angle_C)+r3_prime*sin(beta3)-r4_prime*sin(gamma4+
   Angle_C)-Link;
11 constraint7=r4_prime*cos(theta4-Angle_C)+r3_prime*cos(beta3)-r4_prime*cos(gamma4+
   Angle_C);
12 end
```

```
1 function [constraint1,constraint2,constraint3,constraint4,constraint5,constraint6,
   constraint7] = Constraints_top_beam(r2,r3,r4,r4_prime,r3_prime,Link,theta2,theta3,
   theta4,gamma2,gamma3,gamma4,Angle_C,beta3,preload,angle)
2 %CONSTRAINTS_ALL_VARIABLES Summary of this function goes here
3 % Detailed explanation goes here
4
5 constraint1=gamma2-angle;
6 constraint2=r2*sin(theta2)+r3*sin(theta3)-r4*sin(theta4);
7 constraint3=r2*cos(theta2)+r3*cos(theta3)-r4*cos(theta4)-(r2+r3+r4-preload);
8 constraint4=r2*sin(gamma2)+r3*sin(gamma3)-r4*sin(gamma4);
9 constraint5=r2*cos(gamma2)+r3*cos(gamma3)-r4*cos(gamma4)-(r2+r3+r4-preload);
10 constraint6=r4_prime*sin(theta4-Angle_C)+r3_prime*sin(beta3)-r4_prime*sin(gamma4+
   Angle_C)-Link;
11 constraint7=r4_prime*cos(theta4-Angle_C)+r3_prime*cos(beta3)-r4_prime*cos(gamma4+
   Angle_C);
12 end
```

B.4.2. Vector loop functions

```
1 function [gamma3,gamma4] = Top_beam_gamma2(gamma2,k1,k2,k3,k4,k5)
2 %BOTTOM_BEAM_THETA2 This function is used when solving for theta2
3 % For a given theta2 there are two possibilities for theta3 and theta4.
4 % Only in a singularity state there both solutions are possible and can
5 % change.
6 A=k1-k2+(1-k3)*cos(gamma2);
7 B=-sin(gamma2);
8 C=k1+k2-(1+k3)*cos(gamma2);
9
10 D=k4+k2-(1+k5)*cos(gamma2);
11 E=sin(gamma2);
12 F=k4-k2+(1-k5)*cos(gamma2);
13
14 gamma3(1,1:2)=[2*atan((-E+sqrt(E^2-D*F))/(D));2*atan((-E-sqrt(E^2-D*F))/(D))];
15 gamma4(1,1:2)=[2*atan((-B-sqrt(B^2-A*C))/(A));2*atan((-B+sqrt(B^2-A*C))/(A))];
16 end
```

```
1 function [gamma2,gamma3] = Top_beam_theta2(gamma4,k1,k2,k3,k5,k6)
2 %BOTTOM_BEAM_THETA4 This function is used when solving for theta2 with
3 % gamma4
4 % For a given theta4 there are two possibilities for theta2 and theta3.
5 % Only in a singularity state both solutions are possible and can
6 % change.
7 A=k6+k3+(1+k5)*cos(gamma4);
8 B=-sin(gamma4);
9 C=k6-k3+(k5-1)*cos(gamma4);
10
11 D=k1+k3+(1+k2)*cos(gamma4);
12 E=-sin(gamma4);
13 F=k1-k3+(k2-1)*cos(gamma4);
14
```

```

15 gamma2(1,1:2)=[2*atan((-E+sqrt(E^2-D*F))/(D));2*atan((-E-sqrt(E^2-D*F))/(D))];
16 gamma3(1,1:2)=[2*atan((-B-sqrt(B^2-A*C))/(A));2*atan((-B+sqrt(B^2-A*C))/(A))];
17 end

```

```

1 function [theta2,theta3] = Bottom_beam_gamma2(theta4,k1,k2,k3,k5,k6)
2 %BOTTOM_BEAM_THETA4 This function is used when solving for gamma2 with theta4
3 % For a given theta4 there are two possibilities for theta2 and theta3.
4 % Only in a singularity state both solutions are possible and can
5 % change.
6 A=k6+k3+(1+k5)*cos(theta4);
7 B=-sin(theta4);
8 C=k6-k3+(k5-1)*cos(theta4);
9
10 D=k1+k3+(1+k2)*cos(theta4);
11 E=-sin(theta4);
12 F=k1-k3+(k2-1)*cos(theta4);
13
14 theta2(1,1:2)=[2*atan((-E+sqrt(E^2-D*F))/(D));2*atan((-E-sqrt(E^2-D*F))/(D))];
15 theta3(1,1:2)=[2*atan((-B-sqrt(B^2-A*C))/(A));2*atan((-B+sqrt(B^2-A*C))/(A))];
16 end

```

```

1 function [theta3,theta4] = Bottom_beam_theta2(theta2,k1,k2,k3,k4,k5)
2 %BOTTOM_BEAM_THETA2 This function is used when solving for theta2
3 % For a given theta2 there are two possibilities for theta3 and theta4.
4 % Only in a singularity state there both solutions are possible and can
5 % change.
6 A=k1-k2+(1-k3)*cos(theta2);
7 B=-sin(theta2);
8 C=k1+k2-(1+k3)*cos(theta2);
9
10 D=k4+k2-(1+k5)*cos(theta2);
11 E=sin(theta2);
12 F=k4-k2+(1-k5)*cos(theta2);
13
14 theta3(1,1:2)=[2*atan((-E+sqrt(E^2-D*F))/(D));2*atan((-E-sqrt(E^2-D*F))/(D))];
15 theta4(1,1:2)=[2*atan((-B-sqrt(B^2-A*C))/(A));2*atan((-B+sqrt(B^2-A*C))/(A))];
16 end

```

```

1 function [beta2,beta3] = Rigid_Link_gamma2(beta4,k7,k8,k9,k10,k12)
2 %RIGID_LINK Summary of this function goes here
3 % Detailed explanation goes here
4 A=k12+k10*sin(beta4)+cos(beta4);
5 B=-k8-sin(beta4);
6 C=k12+k10*sin(beta4)-cos(beta4);
7
8 D=k9+k7*sin(beta4)+cos(beta4);
9 E=-k8-sin(beta4);
10 F=k9+k7*sin(beta4)-cos(beta4);
11
12 beta2(1,1:2)=[2*atan((-E+sqrt(E^2-D*F))/(D));2*atan((-E-sqrt(E^2-D*F))/(D))];
13 beta3(1,1:2)=[2*atan((-B-sqrt(B^2-A*C))/(A));2*atan((-B+sqrt(B^2-A*C))/(A))];
14
15 end

```

```

1 function [beta3,beta4] = Rigid_Link_theta2(beta2,k7,k8,k9,k10,k11)
2 %RIGID_LINK Summary of this function goes here
3 % Detailed explanation goes here
4 A=k9-k8*sin(beta2)+cos(beta2);
5 B=k7-sin(beta2);
6 C=k9-k8*sin(beta2)-cos(beta2);
7
8 D=k11-k10*sin(beta2)-cos(beta2);
9 E=sin(beta2)-k7;
10 F=k11-k10*sin(beta2)+cos(beta2);
11
12 beta3(1,1:2)=[2*atan((-E+sqrt(E^2-D*F))/(D));2*atan((-E-sqrt(E^2-D*F))/(D))];
13 beta4(1,1:2)=[2*atan((-B-sqrt(B^2-A*C))/(A));2*atan((-B+sqrt(B^2-A*C))/(A))];
14
15 end

```

B.4.3. Equation setup

```

1 function [Equation_array] = Equations_bot_beam(Etotal, theta2, theta3, theta4, gamma2,
   gamma3, gamma4, beta3, alpha2, lambda1, lambda2, lambda3, lambda4, lambda5, lambda6, lambda7,
   const1, const2, const3, const4, const5, const6, const7)
2 %EQUATION__all_variables_v3 is used in Freudenstein v3
3
4 eq1=diff(Etotal, theta2)-(lambda1*diff(const1, theta2)+lambda2*diff(const2, theta2)+
   lambda3*diff(const3, theta2)+lambda4*diff(const4, theta2)+lambda5*diff(const5, theta2)+
   lambda6*diff(const6, theta2)+lambda7*diff(const7, theta2));
5 eq2=diff(Etotal, theta3)-(lambda1*diff(const1, theta3)+lambda2*diff(const2, theta3)+
   lambda3*diff(const3, theta3)+lambda4*diff(const4, theta3)+lambda5*diff(const5, theta3)+
   lambda6*diff(const6, theta3)+lambda7*diff(const7, theta3));
6 eq3=diff(Etotal, theta4)-(lambda1*diff(const1, theta4)+lambda2*diff(const2, theta4)+
   lambda3*diff(const3, theta4)+lambda4*diff(const4, theta4)+lambda5*diff(const5, theta4)+
   lambda6*diff(const6, theta4)+lambda7*diff(const7, theta4));
7 eq4=diff(Etotal, gamma2)-(lambda1*diff(const1, gamma2)+lambda2*diff(const2, gamma2)+
   lambda3*diff(const3, gamma2)+lambda4*diff(const4, gamma2)+lambda5*diff(const5, gamma2)+
   lambda6*diff(const6, gamma2)+lambda7*diff(const7, gamma2));
8 eq5=diff(Etotal, gamma3)-(lambda1*diff(const1, gamma3)+lambda2*diff(const2, gamma3)+
   lambda3*diff(const3, gamma3)+lambda4*diff(const4, gamma3)+lambda5*diff(const5, gamma3)+
   lambda6*diff(const6, gamma3)+lambda7*diff(const7, gamma3));
9 eq6=diff(Etotal, gamma4)-(lambda1*diff(const1, gamma4)+lambda2*diff(const2, gamma4)+
   lambda3*diff(const3, gamma4)+lambda4*diff(const4, gamma4)+lambda5*diff(const5, gamma4)+
   lambda6*diff(const6, gamma4)+lambda7*diff(const7, gamma4));
10 eq7=diff(Etotal, beta3)-(lambda1*diff(const1, beta3)+lambda2*diff(const2, beta3)+lambda3*
   diff(const3, beta3)+lambda4*diff(const4, beta3)+lambda5*diff(const5, beta3)+lambda6*
   diff(const6, beta3)+lambda7*diff(const7, beta3));
11 eq8=diff(Etotal, alpha2)-(lambda1*diff(const1, alpha2)+lambda2*diff(const2, alpha2)+
   lambda3*diff(const3, alpha2)+lambda4*diff(const4, alpha2)+lambda5*diff(const5, alpha2)+
   lambda6*diff(const6, alpha2)+lambda7*diff(const7, alpha2));
12 eq9=const1==0;
13 eq10=const2==0;
14 eq11=const3==0;
15 eq12=const4==0;
16 eq13=const5==0;
17 eq14=const6==0;
18 eq15=const7==0;
19 Equation_array=[eq1, eq2, eq3, eq4, eq5, eq6, eq7, eq8, eq9, eq10, eq11, eq12, eq13, eq14, eq15];
20
21 end

1 function [Equation_array] = Equations_top_beam_v2(Etotal, theta2, theta3, theta4, gamma2,
   gamma3, gamma4, beta3, alpha2, lambda2, lambda3, lambda4, lambda5, lambda6, lambda7, const2,
   const3, const4, const5, const6, const7)
2 %EQUATION__all_variables_v3 is used in Freudenstein v3
3
4 eq1=diff(Etotal, theta2)-(lambda2*diff(const2, theta2)+lambda3*diff(const3, theta2)+
   lambda4*diff(const4, theta2)+lambda5*diff(const5, theta2)+lambda6*diff(const6, theta2)+
   lambda7*diff(const7, theta2));
5 eq2=diff(Etotal, theta3)-(lambda2*diff(const2, theta3)+lambda3*diff(const3, theta3)+
   lambda4*diff(const4, theta3)+lambda5*diff(const5, theta3)+lambda6*diff(const6, theta3)+
   lambda7*diff(const7, theta3));
6 eq3=diff(Etotal, theta4)-(lambda2*diff(const2, theta4)+lambda3*diff(const3, theta4)+
   lambda4*diff(const4, theta4)+lambda5*diff(const5, theta4)+lambda6*diff(const6, theta4)+
   lambda7*diff(const7, theta4));
7 eq4=diff(Etotal, gamma2)-(lambda2*diff(const2, gamma2)+lambda3*diff(const3, gamma2)+
   lambda4*diff(const4, gamma2)+lambda5*diff(const5, gamma2)+lambda6*diff(const6, gamma2)+
   lambda7*diff(const7, gamma2));
8 eq5=diff(Etotal, gamma3)-(lambda2*diff(const2, gamma3)+lambda3*diff(const3, gamma3)+
   lambda4*diff(const4, gamma3)+lambda5*diff(const5, gamma3)+lambda6*diff(const6, gamma3)+
   lambda7*diff(const7, gamma3));
9 eq6=diff(Etotal, gamma4)-(lambda2*diff(const2, gamma4)+lambda3*diff(const3, gamma4)+
   lambda4*diff(const4, gamma4)+lambda5*diff(const5, gamma4)+lambda6*diff(const6, gamma4)+
   lambda7*diff(const7, gamma4));
10 eq7=diff(Etotal, beta3)-(lambda2*diff(const2, beta3)+lambda3*diff(const3, beta3)+lambda4*
   diff(const4, beta3)+lambda5*diff(const5, beta3)+lambda6*diff(const6, beta3)+lambda7*
   diff(const7, beta3));
11 eq8=diff(Etotal, alpha2)-(lambda2*diff(const2, alpha2)+lambda3*diff(const3, alpha2)+
   lambda4*diff(const4, alpha2)+lambda5*diff(const5, alpha2)+lambda6*diff(const6, alpha2)+
   lambda7*diff(const7, alpha2));
12 % eq9=const1==0;

```

```

13 eq10=const2==0;
14 eq11=const3==0;
15 eq12=const4==0;
16 eq13=const5==0;
17 eq14=const6==0;
18 eq15=const7==0;
19 Equation_array=[eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8,eq10,eq11,eq12,eq13,eq14,eq15];
20
21 end

1 function [Equation_array] = Equations_top_beam(Etotal,theta2,theta3,theta4,gamma2,
    gamma3,gamma4,beta3,alpha2,lambda1,lambda2,lambda3,lambda4,lambda5,lambda6,lambda7,
    const1,const2,const3,const4,const5,const6,const7)
2 %EQUATION__all_variables_v3 is used in Freudenstein v3
3
4 eq1=diff(Etotal,theta2)-(lambda1*diff(const1,theta2)+lambda2*diff(const2,theta2)+
    lambda3*diff(const3,theta2)+lambda4*diff(const4,theta2)+lambda5*diff(const5,theta2)+
    lambda6*diff(const6,theta2)+lambda7*diff(const7,theta2));
5 eq2=diff(Etotal,theta3)-(lambda1*diff(const1,theta3)+lambda2*diff(const2,theta3)+
    lambda3*diff(const3,theta3)+lambda4*diff(const4,theta3)+lambda5*diff(const5,theta3)+
    lambda6*diff(const6,theta3)+lambda7*diff(const7,theta3));
6 eq3=diff(Etotal,theta4)-(lambda1*diff(const1,theta4)+lambda2*diff(const2,theta4)+
    lambda3*diff(const3,theta4)+lambda4*diff(const4,theta4)+lambda5*diff(const5,theta4)+
    lambda6*diff(const6,theta4)+lambda7*diff(const7,theta4));
7 eq4=diff(Etotal,gamma2)-(lambda1*diff(const1,gamma2)+lambda2*diff(const2,gamma2)+
    lambda3*diff(const3,gamma2)+lambda4*diff(const4,gamma2)+lambda5*diff(const5,gamma2)+
    lambda6*diff(const6,gamma2)+lambda7*diff(const7,gamma2));
8 eq5=diff(Etotal,gamma3)-(lambda1*diff(const1,gamma3)+lambda2*diff(const2,gamma3)+
    lambda3*diff(const3,gamma3)+lambda4*diff(const4,gamma3)+lambda5*diff(const5,gamma3)+
    lambda6*diff(const6,gamma3)+lambda7*diff(const7,gamma3));
9 eq6=diff(Etotal,gamma4)-(lambda1*diff(const1,gamma4)+lambda2*diff(const2,gamma4)+
    lambda3*diff(const3,gamma4)+lambda4*diff(const4,gamma4)+lambda5*diff(const5,gamma4)+
    lambda6*diff(const6,gamma4)+lambda7*diff(const7,gamma4));
10 eq7=diff(Etotal,beta3)-(lambda1*diff(const1,beta3)+lambda2*diff(const2,beta3)+lambda3*
    diff(const3,beta3)+lambda4*diff(const4,beta3)+lambda5*diff(const5,beta3)+lambda6*
    diff(const6,beta3)+lambda7*diff(const7,beta3));
11 eq8=diff(Etotal,alpha2)-(lambda1*diff(const1,alpha2)+lambda2*diff(const2,alpha2)+
    lambda3*diff(const3,alpha2)+lambda4*diff(const4,alpha2)+lambda5*diff(const5,alpha2)+
    lambda6*diff(const6,alpha2)+lambda7*diff(const7,alpha2));
12 eq9=const1==0;
13 eq10=const2==0;
14 eq11=const3==0;
15 eq12=const4==0;
16 eq13=const5==0;
17 eq14=const6==0;
18 eq15=const7==0;
19 Equation_array=[eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8,eq9,eq10,eq11,eq12,eq13,eq14,eq15];
20
21 end

1 function [Equation_array] = Equations_top_beam_v2(Etotal,theta2,theta3,theta4,gamma2,
    gamma3,gamma4,beta3,alpha2,lambda2,lambda3,lambda4,lambda5,lambda6,lambda7,const2,
    const3,const4,const5,const6,const7)
2 %EQUATION__all_variables_v3 is used in Freudenstein v3
3
4 eq1=diff(Etotal,theta2)-(lambda2*diff(const2,theta2)+lambda3*diff(const3,theta2)+
    lambda4*diff(const4,theta2)+lambda5*diff(const5,theta2)+lambda6*diff(const6,theta2)+
    lambda7*diff(const7,theta2));
5 eq2=diff(Etotal,theta3)-(lambda2*diff(const2,theta3)+lambda3*diff(const3,theta3)+
    lambda4*diff(const4,theta3)+lambda5*diff(const5,theta3)+lambda6*diff(const6,theta3)+
    lambda7*diff(const7,theta3));
6 eq3=diff(Etotal,theta4)-(lambda2*diff(const2,theta4)+lambda3*diff(const3,theta4)+
    lambda4*diff(const4,theta4)+lambda5*diff(const5,theta4)+lambda6*diff(const6,theta4)+
    lambda7*diff(const7,theta4));
7 eq4=diff(Etotal,gamma2)-(lambda2*diff(const2,gamma2)+lambda3*diff(const3,gamma2)+
    lambda4*diff(const4,gamma2)+lambda5*diff(const5,gamma2)+lambda6*diff(const6,gamma2)+
    lambda7*diff(const7,gamma2));
8 eq5=diff(Etotal,gamma3)-(lambda2*diff(const2,gamma3)+lambda3*diff(const3,gamma3)+
    lambda4*diff(const4,gamma3)+lambda5*diff(const5,gamma3)+lambda6*diff(const6,gamma3)+
    lambda7*diff(const7,gamma3));

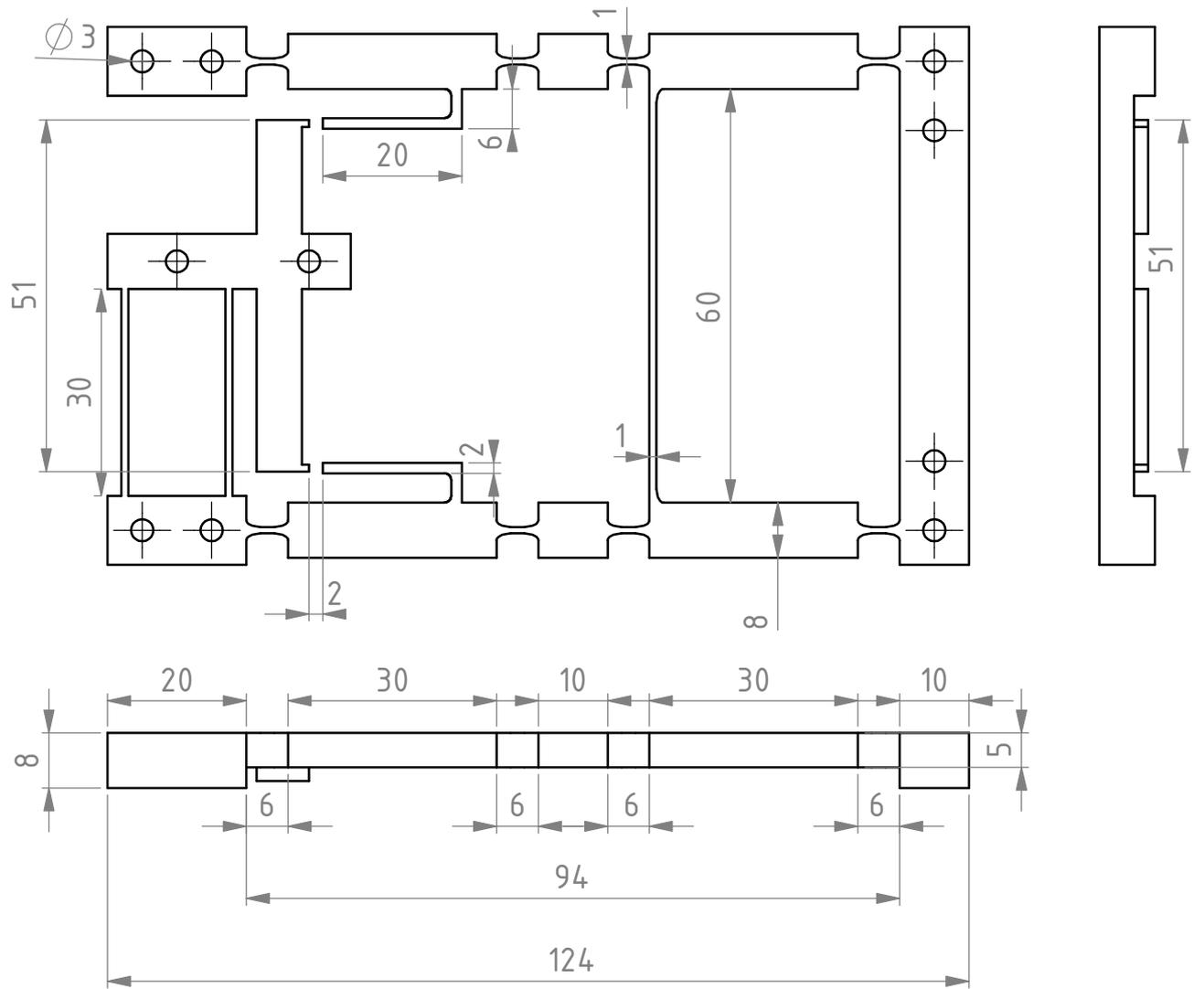
```

```
9 eq6=diff(Etotal,gamma4)-(lambda2*diff(const2,gamma4)+lambda3*diff(const3,gamma4)+
  lambda4*diff(const4,gamma4)+lambda5*diff(const5,gamma4)+lambda6*diff(const6,gamma4)
  +lambda7*diff(const7,gamma4));
10 eq7=diff(Etotal,beta3)-(lambda2*diff(const2,beta3)+lambda3*diff(const3,beta3)+lambda4*
  diff(const4,beta3)+lambda5*diff(const5,beta3)+lambda6*diff(const6,beta3)+lambda7*
  diff(const7,beta3));
11 eq8=diff(Etotal,alpha2)-(lambda2*diff(const2,alpha2)+lambda3*diff(const3,alpha2)+
  lambda4*diff(const4,alpha2)+lambda5*diff(const5,alpha2)+lambda6*diff(const6,alpha2)
  +lambda7*diff(const7,alpha2));
12 % eq9=const1==0;
13 eq10=const2==0;
14 eq11=const3==0;
15 eq12=const4==0;
16 eq13=const5==0;
17 eq14=const6==0;
18 eq15=const7==0;
19 Equation_array=[eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8,eq10,eq11,eq12,eq13,eq14,eq15];
20
21 end
```


C

Design and Fabrication

In the following pages technical drawings with complete measurements of the fabricated design are presented.



	units mm	scale 1:1	quantity <<nr>>	date 24-4-2023	remark All holes are 3 mm
material	PET	mass 24.70gr			
author M.F.H. de Jong - 4493370	group PME 46	 Delft University of Technology			
name Switching mechanism	format A4				
H:\My Documents\SolidWorks 2022\SW Files\					

D

Finite Element Analysis

The finite element analysis presented in the paper is performed in the SOLIDWORKS simulation environment because it easily enables interaction between contact surfaces. In the development of a model ANSYS simulations were performed to investigate the bi-stability of a lumped compliant beam and the bi-stability of a distributed compliant beam. The presented prbm in Figure 3.3a is modeled in ANSYS except for the rigid pushing rods. The complete script in text file can be found on the next page. This script can be run in the session editor in ANSYS 2020 R2. It can be seen that he observed peak stresses in this script are 80 MPa.

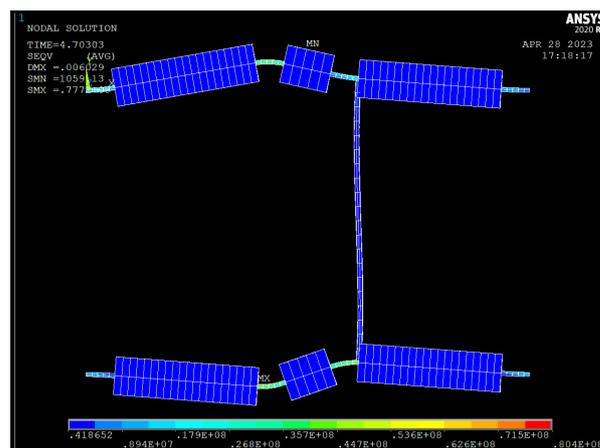


Figure D.1: Vector loop diagram

In the next figure the pushing rods are implemented. This model is different from the prbm because the pushing rods are not modeled as rigid. The input beams have been set to have a thickness of 1.5 mm. The top and bottom actuation are similar and do not result in a different force displacement or peak stresses. The maximum observed stress during the state switching is about 70 MPa for a preload of .75 mm and 81 MPa for 1 mm preload. The model would have been more alike if the connection of the pushing rods was on the surface of the beams. This also holds for the connecting link.

Looking at the force displacement curves the maximum input force required for mechanism is 6.5 N and 7.5 N for 0.75 and 1 mm preload respectively. With increased preload the input displacement also increases. After the mechanism has snapped the pushing rod is pushed a little further, then an almost linear force displacement profile is observed. This is the deformation of the pushing rod. With regards to the maximum actuation force it resembles the fabricated prototype, except in the ANSYS simulation the input mechanism is not present.

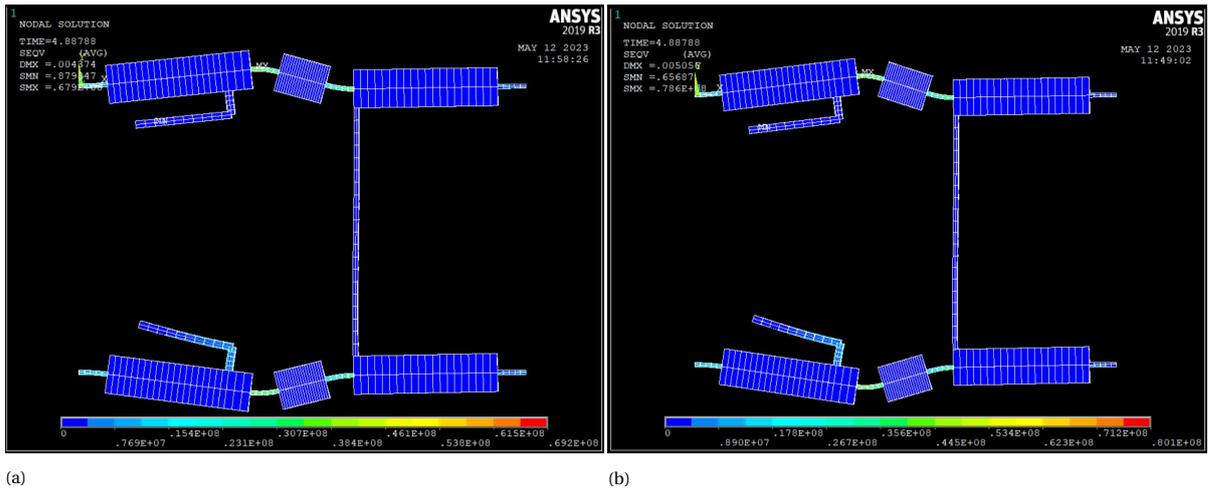


Figure D.2: Stress distribution in beam under several preloads a)0.75 mm b) 1mm

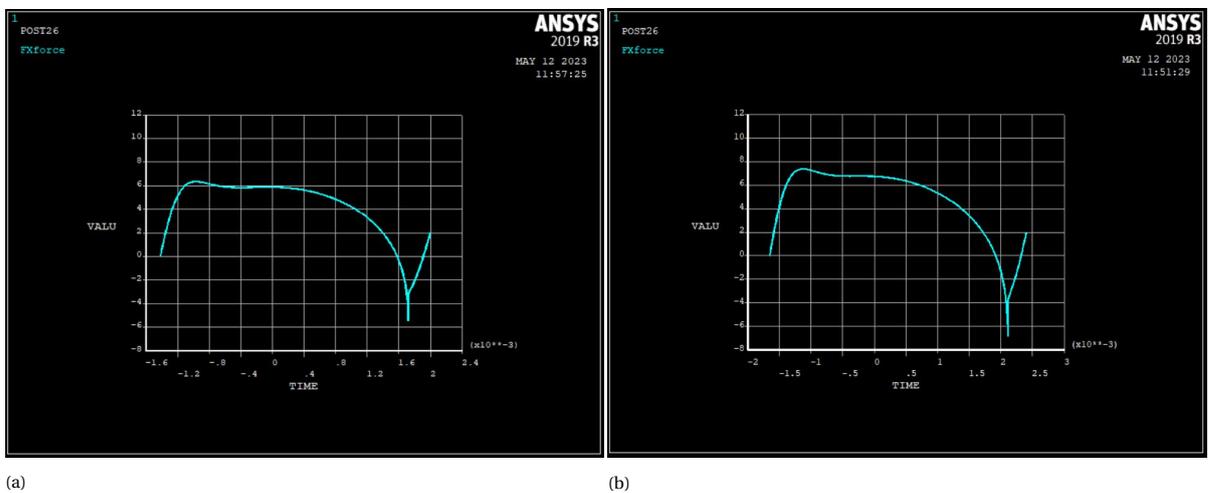


Figure D.3: Force displacement graph under several preloads a)0.75 mm b) 1mm

ANSYS script used for Figure D.1

```

1 ! Last update: 23-2-2023
2 ! Lumped design PRBM
3
4 !! Initialize
5 FINISH      !some commands to clear the workspace
6 /CLEAR
7 /OUTPUT
8
9 !!!!parameters!!!!
10 E = 2.1e9    !Young's modulus (Pa)
11 v = 0.37    !Poisson ratio (-)
12 rho = 1240  !Density (Kg/m3)
13
14 h = 5e-3    !height (m)
15 w = 8e-3    !width (m)
16
17 t = 0.9e-3  !thicknesss of flexure
18 l= 6e-3    !length of flexure
19 Link= 60e-3 !Distance between the two beams
20 tl=1e-3    !Thickness of Link
21
22 r2=30e-3    !length beam 2
23 r3=10e-3    !length beam 3

```

```

24 r4=30e-3      !length beam 4
25
26 n = 20        !elements per flexible beam
27 n2=5
28 s = 100      !max substeps per iteration
29
30 d= 1e-3       !preload displacement
31 act=6e-3      !input displacement
32
33 !!!!preprocessing!!!!
34 /PREP7        !enter preprocessor menu
35
36 !define element types
37 ET,1,BEAM188  !element type 1
38
39 !define cross section
40 /ESHAPE,1
41 SECTYPE,1,beam,RECT  !choose rectangular beam section
42 SECDATA,t,h         !cross section with earlier defined parameters
43 SECTYPE,2,beam,RECT  !choose rectangular beam section
44 SECDATA,w,h         !cross section with earlier defined parameters
45 SECTYPE,3,beam,RECT  !choose rectangular beam section
46 SECDATA,tl,h        !cross section with earlier defined parameters
47
48 !material properties
49 MP,EX,1,E        !Young's modulus of material 1
50 MP,PRXY,1,v      !Poisson ratio of material 1
51
52 !Define keypoints
53
54 K,1,0,0,0       !keypoints x,y,z
55 K,2,1,0,0
56 K,3,1+r2,0,0
57 K,4,2*1+r2,0,0
58 K,5,2*1+r2+r3,0,0
59 K,6,3*1+r2+r3,0,0
60 K,7,3*1+r2+r3+r4,0,0
61 K,8,4*1+r2+r3+r4,0,0
62 K,9,0,-Link,0   !keypoints x,y,z
63 K,10,1,-Link,0
64 K,11,1+r2,-Link,0
65 K,12,2*1+r2,-Link,0
66 K,13,2*1+r2+r3,-Link,0
67 K,14,3*1+r2+r3,-Link,0
68 K,15,3*1+r2+r3+r4,-Link,0
69 K,16,4*1+r2+r3+r4,-Link,0
70
71 !define lines
72
73 *GET,ID1,LINE,0,NUM,MAXD !retrieve the max number used to identify a line and put in
  ID1
74 L,1,2
75 L,3,4
76 L,5,6
77 L,7,8
78 L,9,10
79 L,11,12
80 L,13,14
81 L,15,16
82
83 *GET,ID2,LINE,0,NUM,MAXD !retrieve the max number used to identify a line and put in
  ID2
84 L,2,3
85 L,6,7
86 L,10,11
87 L,14,15
88
89 *GET,ID3,LINE,0,NUM,MAXD !retrieve the max number used to identify a line and put in
  ID3
90 L,6,14
91

```

```

92 *GET, ID4, LINE, 0, NUM, MAXD !retrieve the max number used to identify a line and put in
    ID4
93 L, 4, 5
94
95 *GET, ID5, LINE, 0, NUM, MAXD !retrieve the max number used to identify a line and put in
    ID5
96 L, 12, 13
97
98 !Mesh lines
99
100 TYPE, 1 !select element type for mesh
101 SECNUM, 1 !select section for mesh
102 LSEL, S, LINE, , ID1+1, ID2 !select lines to be meshed
103 LESIZE, ALL, , , n2 !make all lines consist of n elements
104 LMESS, ALL !mesh all selected lines
105
106 ALLSEL, ALL !select all lines again otherwise ansys acts
107 !as if the not selected lines do not exist
108 TYPE, 1
109 SECNUM, 2 !select section for mesh
110 LSEL, S, LINE, , ID2+1, ID3 !select lines to be meshed
111 LESIZE, ALL, , , n !make all lines consist of 1 element
112 LMESS, ALL !mesh all
113
114 ALLSEL, ALL
115
116 TYPE, 1
117 SECNUM, 3 !select section for mesh
118 LSEL, S, LINE, , ID3+1, ID4 !select lines to be meshed
119 LESIZE, ALL, , , n !make all lines consist of 1 element
120 LMESS, ALL !mesh all
121
122 ALLSEL, ALL
123
124 TYPE, 1
125 SECNUM, 2 !select section for mesh
126 LSEL, S, LINE, , ID4+1, ID5 !select lines to be meshed
127 LESIZE, ALL, , , n2 !make all lines consist of 1 element
128 LMESS, ALL !mesh all
129
130 ALLSEL, ALL
131
132 TYPE, 1
133 SECNUM, 2 !select section for mesh
134 LSEL, S, LINE, , ID5+1, ID6 !select lines to be meshed
135 LESIZE, ALL, , , n2 !make all lines consist of 1 element
136 LMESS, ALL !mesh all
137 FINISH !exit pre-processor
138
139 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
140
141 !! ID S HERE !!
142
143 ksel, s, , , 1
144 nslk, s
145 *get, id_fix1, node, , num, min
146 allsel
147
148 ksel, s, , , 9
149 nslk, s
150 *get, id_fix2, node, , num, min
151 allsel
152
153 ksel, s, , , 8
154 nslk, s
155 *get, id_disp1, node, , num, min
156 allsel
157
158 ksel, s, , , 16
159 nslk, s
160 *get, id_disp2, node, , num, min

```

```

161 allsel
162
163 ksel, s, , , 3
164 nslk, s
165 *get, act_node1, node, , num, min
166 allsel
167
168 ksel, s, , , 11
169 nslk, s
170 *get, act_node2, node, , num, min
171 allsel
172
173 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
174 !!!!!!!! visualize !!!!!!!!
175 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
176 !Show meshed lines
177 /eshape, 1
178 /view, 1, 1, 1, 1
179 eplot
180
181 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
182 !!!!!!!! loads !!!!!!!!
183 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
184
185 /SOLU
186
187 antype, 0
188 nlgeom, on                ! Large deformations are expected
189 eqslv, sparse
190 outres, all, all
191 autots, on
192 neqit, 40
193 nsubst, 10, , 10
194
195 KBC,0
196 time,1
197 d, act_node1, uy, 0.001
198 d, act_node2, uy, 0.001
199 d,id_fix1, all
200 d,id_fix2, all
201 d,id_disp1, all
202 d,id_disp2, all
203 ddele,id_disp1, ux
204 ddele,id_disp2, ux
205 solve
206
207 time,2
208 deltim, 1e-3, 1e-5, 1/100, on    !defines stepsize for loadstep
209 *get, dispx1, node, id_disp1, u, x
210 *get, dispx2, node, id_disp2, u, x
211 d,id_disp1, ux, dispx1
212 d,id_disp2, ux, dispx2
213 solve
214
215 time, 3
216 ddele, act_node1, all
217 ddele, act_node2, all
218 d, id_disp1, ux, -1e-3 ! 1 mm preload displacement
219 d, id_disp2, ux, -1e-3 ! 1 mm preload displacement
220 solve
221
222 KBC,1 !Change to KBC,1 if you do not want a force in this timestep. KBC,0 otherwise
223 time, 4
224 !*get, dispy1, node, act_node1, u, y
225 *get, dispy2, node, act_node2, u, y
226 !d, act_node1, uy, dispy1
227 d, act_node2, uy, dispy2
228 solve
229
230 KBC,0
231 time,5

```

```

232 deltim, 1e-6, 1e-8, 1/250, on          !defines stepsize for loadstep
233 d,act_node2, uy, -Act
234 !ddelete, act_node1, all
235 solve
236
237 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
238 !! Display deformed state !!
239 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
240
241 /post1
242 /view, 1, 0, 0, 1
243 subset, last
244 pldisp, 1
245
246 !! Plot results !!
247 *GET, N_ACTUALSTEPS_1, active, 0, solu, ncmss ! Count the number of substeps to size
           the table correctly
248 /POST26
249 NUMVAR, 200                                ! Go to
           postprocessor menu
250 TIMERANGE,4,5 ! Plot data from loadstep 5 only (throw away data between time = 0 and
           time = 4)
251
252 NSOL,2,id_disp1,U,X,Displacement           ! Get the displacement data
           in the horizontal direction of the node with ID_DISPLACED_1
253 RFORCE,3,id_disp1,F,X,FXforce ! Get the reaction force in the horizontal direction
254 NSOL,4,id_disp2,U,X,Displacement! Get the displacement data in the horizontal direction
           of the node with
255 RFORCE,5,id_disp2,F,X,FXforce
256 NSOL,6,act_node1,U,Y,Displacement
257 RFORCE,7,act_node1,F,Y,FYforce
258 NSOL,8,act_node2,U,Y,Displacement
259 RFORCE,9,act_node2,F,Y,FYforce
260
261 XVAR,8          ! Plot displacement in y-direction of act_node2
262 PLVAR,9        ! Plot force in y-direction of act_node2

```

ANSYS script used for Figure D.2 and Figure D.3

```

1 ! Last update: 32-2-2023
2 ! Lumped design PRBM
3
4 !! Initialize
5
6
7 FINISH          !some commands to clear the workspace
8 /CLEAR
9 /OUTPUT
10
11 !!!!parameters!!!!
12
13 E = 2.1e9      !Young's modulus (Pa)
14 v = 0.37      !Poisson ratio (-)
15 rho = 1240    !Density (Kg/m3)
16
17 h = 5e-3      !height (m)
18 w = 8e-3      !width (m)
19 w2= 1.5e-3    !width arm
20
21 t = 0.9e-3    !thicknesss of flexure
22 l= 6e-3       !length of flexure
23 Link= 60e-3   !Distance between the two beams
24 tl=1e-3      !Thickness of Link
25
26 r2=30e-3     !length beam 2
27 r3=10e-3     !length beam 3
28 r4=30e-3     !length beam 4
29
30 arm=5e-3+w/2  !moment arm distance form beam
31 l_arm=20e-3   !length arm
32 ld=0         !difference in arm length for actuation

```

```

33 connect=5e-3      !connection node compensate for arm thickness
34
35 n = 20           !elements per flexible beam
36 n2 = 5           !elements per flexure
37 n3= 10           !elements per connecting arm
38
39 s = 100          !max substeps per iteration
40
41 d= 0.75e-3       !preload displacement
42 act=2.4e-3       !input displacement end position
43
44 !!!!preprocessing!!!!
45
46 /PREP7           !enter preprocessor menu
47
48 !define element types
49 ET,1,BEAM188     !element type 1
50 ET,2,MPC184      !element type 2
51 keyopt,2,1,1
52
53 !define cross section
54 /ESHAPE,1
55 SECTYPE,1,beam,RECT !choose rectangular beam section
56 SECDATA,t,h      !cross section with earlier defined parameters
57 SECTYPE,2,beam,RECT !choose rectangular beam section
58 SECDATA,w,h      !cross section with earlier defined parameters
59 SECTYPE,3,beam,RECT !choose rectangular beam section
60 SECDATA,tl,h     !cross section with earlier defined parameters
61 SECTYPE,4,beam,RECT !choose rectangular beam section
62 SECDATA,w2,h     !cross section with earlier defined parameters
63
64 !material properties
65 MP,EX,1,E        !Young's modulus of material 1
66 MP,PRXY,1,v      !Poisson ratio of material 1
67
68 !Define keypoints
69
70 K,1,0,0,0        !keypoints x,y,z
71 K,2,1,0,0
72 K,3,1+r2,0,0
73 K,4,2*1+r2,0,0
74 K,5,2*1+r2+r3,0,0
75 K,6,3*1+r2+r3,0,0
76 K,7,3*1+r2+r3+r4,0,0
77 K,8,4*1+r2+r3+r4,0,0
78 K,9,0,-Link,0   !keypoints x,y,z
79 K,10,1,-Link,0
80 K,11,1+r2,-Link,0
81 K,12,2*1+r2,-Link,0
82 K,13,2*1+r2+r3,-Link,0
83 K,14,3*1+r2+r3,-Link,0
84 K,15,3*1+r2+r3+r4,-Link,0
85 K,16,4*1+r2+r3+r4,-Link,0
86
87 K,17,1+r2-L_arm-connect,-arm,0
88 K,18,1+r2-connect,-arm,0
89 K,19,1+r2-L_arm-connect,arm-Link,0
90 k,20,1+r2-connect,arm-Link,0
91
92 K,21,1+r2-connect,0,0
93 K,22,1+r2-connect,-Link,0
94
95 K,23,1+r2-connect,-w/2,0
96 K,24,1+r2-connect,-Link+w/2,0
97 K,25,3*1+r2+r3+tl/2,-w/2,0
98 k,26,3*1+r2+r3+tl/2,-Link+w/2,0
99
100 !define lines
101 !flexible lines
102
103 *GET,ID1,LINE,0,NUM,MAXD !retrieve the maximum number used to identify a line (0) and

```

```

        put it in parameter ID1
104 L,1,2
105 L,3,4
106 L,5,6
107 L,7,8
108 L,9,10
109 L,11,12
110 L,13,14
111 L,15,16
112
113 *GET, ID2, LINE, 0, NUM, MAXD !retrieve the maximum number used to identify a line (2) and
        put it in parameter ID2
114 L,2,21
115 L,4,5
116 L,6,7
117 L,10,22
118 L,12,13
119 L,14,15
120
121 *GET, ID3, LINE, 0, NUM, MAXD !retrieve the maximum number used to identify a line (4) and
        put it in parameter ID4
122 L,21,23
123 L,22,24
124 L,6,25
125 L,14,26
126
127 *GET, ID4, LINE, 0, NUM, MAXD !retrieve the maximum number used to identify a line (3) and
        put it in parameter ID3
128 L,25,26
129
130 *GET, ID5, LINE, 0, NUM, MAXD !retrieve the maximum number used to identify a line (3) and
        put it in parameter ID3
131 L,3,21
132 L,11,22
133
134 *GET, ID6, LINE, 0, NUM, MAXD !retrieve the maximum number used to identify a line (3) and
        put it in parameter ID3
135 L,20,24
136
137 *GET, ID7, LINE, 0, NUM, MAXD !retrieve the maximum number used to identify a line (3) and
        put it in parameter ID3
138 L,17,18
139
140 *GET, ID8, LINE, 0, NUM, MAXD
141 L,19,20
142
143 *GET, ID9, LINE, 0, NUM, MAXD
144 L,18,23
145
146
147
148 !Mesh lines
149
150 TYPE,1          !select element type for mesh
151 SECNUM,1       !select section for mesh
152 LSEL,S,LINE,,ID1+1,ID2 !select lines to be meshed
153 LESIZE,ALL,,,n2 !make all lines consist of n2 elements
154 LMESH,ALL      !mesh all selected lines
155
156 ALLSEL,ALL     !select all lines again otherwise ansys acts as if the not selected
        lines do not exist
157
158 TYPE,1
159 SECNUM,2       !select section for mesh
160 LSEL,S,LINE,,ID2+1,ID3 !select lines to be meshed
161 LESIZE,ALL,,,n !make all lines consist of n elements
162 LMESH,ALL     !mesh all
163
164 ALLSEL,ALL
165
166 TYPE,2

```

```
167 SECNUM,4      !select section for mesh
168 LSEL,S,LINE,,ID3+1,ID4      !select lines to be meshed
169 LESIZE,ALL,,1      !make all lines consist of 1 element
170 LMESH,ALL      !mesh all
171
172 ALLSEL,ALL
173
174 TYPE,1
175 SECNUM,3      !select section for mesh
176 LSEL,S,LINE,,ID4+1,ID5      !select lines to be meshed
177 LESIZE,ALL,,n      !make all lines consist of n elements
178 LMESH,ALL      !mesh all
179
180 ALLSEL,ALL
181
182 TYPE,1
183 SECNUM,2      !select section for mesh
184 LSEL,S,LINE,,ID5+1,ID6      !select lines to be meshed
185 LESIZE,ALL,,3      !make all lines consist of 3 elements
186 LMESH,ALL      !mesh all
187
188 ALLSEL,ALL
189
190 TYPE,1
191 SECNUM,4      !select section for mesh
192 LSEL,S,LINE,,ID6+1,ID7      !select lines to be meshed
193 LESIZE,ALL,,5      !make all lines consist of 5 elements
194 LMESH,ALL      !mesh all
195
196 ALLSEL,ALL
197
198 TYPE,1
199 SECNUM,4      !select section for mesh
200 LSEL,S,LINE,,ID7+1,ID8      !select lines to be meshed
201 LESIZE,ALL,,10      !make all lines consist of 5 elements
202 LMESH,ALL      !mesh all
203
204 ALLSEL,ALL
205
206 TYPE,1
207 SECNUM,4      !select section for mesh
208 LSEL,S,LINE,,ID8+1,ID9      !select lines to be meshed
209 LESIZE,ALL,,10      !make all lines consist of 5 elements
210 LMESH,ALL      !mesh all
211
212 ALLSEL,ALL
213
214 TYPE,1
215 SECNUM,4      !select section for mesh
216 LSEL,S,LINE,,ID9+1,ID10      !select lines to be meshed
217 LESIZE,ALL,,5      !make all lines consist of 5 elements
218 LMESH,ALL      !mesh all
219
220 FINISH      !exit pre-processor
221
222 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
223
224 !! ID S HERE !!
225
226 ksel, s, , , 1
227 nslk, s
228 *get, id_fix1, node, , num, min
229 allsel
230
231 ksel, s, , , 9
232 nslk, s
233 *get, id_fix2, node, , num, min
234 allsel
235
236 ksel, s, , , 8
237 nslk, s
```

```

238 *get, id_disp1, node, , num, min
239 allsel
240
241 ksel, s, , , 16
242 nslk, s
243 *get, id_disp2, node, , num, min
244 allsel
245
246 ksel, s, , , 17
247 nslk, s
248 *get, act_node1, node, , num, min
249 allsel
250
251 ksel, s, , , 19
252 nslk, s
253 *get, act_node2, node, , num, min
254 allsel
255
256 ksel, s, , , 3
257 nslk, s
258 *get, id_pre1, node, , num, min
259 allsel
260
261 ksel, s, , , 11
262 nslk, s
263 *get, id_pre2, node, , num, min
264 allsel
265
266 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
267 !!!!!!!! visualize !!!!!!!!
268 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
269
270 !Find nodes for constraint equation
271 !/PNUM,NODE,1
272 !NPLOT
273
274 /eshape, 1
275 /view, 1, 1, 1, 1
276 eplot
277
278 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
279 !!!!!!!! loads !!!!!!!!
280 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
281
282 /SOLU
283
284 antype, 0
285 nlgeom, on
286 eqslv, sparse
287 outres, all, all
288 autots, on
289 neqit, 40
290 nsubst, 10, , 10
291
292 KBC,0
293 time,1
294 d, id_pre1, uy, 0.0015
295 d, id_pre2, uy, 0.0015
296 d,id_fix1, all
297 d,id_fix2, all
298 d,id_disp1, all
299 d,id_disp2, all
300 !d,act_node1, uz
301 !d,act_node2, uz
302 ddele,id_disp1, ux
303 ddele,id_disp2, ux
304 solve
305
306
307 time,2
308 deltim, 1e-6, 1e-8, 1/250, on !defines stepsize for loadstep

```

```

309 *get, dispx1, node, id_disp1, u, x
310 *get, dispx2, node, id_disp2, u, x
311 d,id_disp1, ux, dispx1
312 d,id_disp2, ux, dispx2
313 solve
314
315
316 time, 3
317 ddele, id_pre1, all
318 ddele, id_pre2, all
319 d, id_disp1, ux, -d ! 1 mm preload displacement
320 d, id_disp2, ux, -d ! 1 mm preload displacement
321 solve
322
323 KBC,1                                !Change to KBC,1 if you do not want a force in
      this timestep. KBC,0 otherwise
324 time, 4
325 !*get, dispx1, node, act_node1, u, x
326 *get, dispx2, node, act_node2, u, x
327 *get, dispy2, node, act_node2, u, y
328 !d, act_node1, ux, dispx1
329 d, act_node2, ux, dispx2
330 d, act_node2, uy, dispy2
331 solve
332
333 KBC,0
334 time,5
335 deltim, 1e-6, 1e-8, 1/250, on        !defines stepsize for loadstep
336 d,act_node2, ux, Act
337 !ddele, act_node1, all
338 solve
339
340 KBC,0
341 time,6
342 deltim, 1e-6, 1e-8, 1/250, on        !defines stepsize for loadstep
343 ddele,act_node2, all
344 solve
345
346 KBC,1                                !Change to KBC,1 if you do not want a force in
      this timestep. KBC,0 otherwise
347 time, 7
348 *get, dispx1, node, act_node1, u, x
349 *get, dispy1, node, act_node1, u, y
350 d, act_node1, ux, dispx1
351 d, act_node1, uy, dispy1
352 solve
353
354 KBC,0
355 time,8
356 deltim, 1e-6, 1e-8, 1/250, on        !defines stepsize for loadstep
357 d,act_node1, ux, Act
358 solve
359
360 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
361 !! Display deformed state !!
362 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
363
364 /post1
365 /view, 1, 0, 0, 1
366 subset, last
367 pldisp, 1
368
369 !! Plot results !!
370 *GET, N_ACTUALSTEPS_1, active, 0, solu, ncmss ! Count the number of substeps to size
      the table correctly
371 /POST26
372 NUMVAR, 200                                ! Go to
      postprocessor menu
373 TIMERANGE,4,8
      ! Plot data from loadstep 5 only (throw away data between time = 0 and time = 4)
374

```

```
375 NSOL,2,id_disp1,U,X,Displacement ! Get the displacement data
    in the horizontal direction of the node with ID_DISPLACED_1
376 RFORCE,3,id_disp1,F,X,FXforce ! Get the reaction force
    in the horizontal direction
377 NSOL,4,id_disp2,U,X,Displacement ! Get the displacement data
    in the horizontal direction of the node with ID_DISPLACED_1
378 RFORCE,5,id_disp2,F,X,FXforce ! Get the reaction force
    in the horizontal direction
379 NSOL,6,act_node1,U,X,Displacement
380 RFORCE,7,act_node1,F,X,FXforce
381 NSOL,8,act_node2,U,X,Displacement
382 RFORCE,9,act_node2,F,X,FXforce
383 NSOL,10,act_node1,U,Y,Displacement
384 NSOL,11,act_node2,U,Y,Displacement
385
386 xvar,6
387 plvar,7
```

End of ANSYS script

E

Literature Paper

On the next pages the literature paper about mechanical information storage can be found. This literature paper was written for the literature research part with course code ME56010.

Mechanical information storage

M.F.H. de Jong^{a,c,*}

^aTU Delft, Mekelweg 2, Delft, 2628 CD, The Netherlands

ARTICLE INFO

Keywords:
Mechanical
Logic
Memory

ABSTRACT

Modern-day society finds itself in the information age and relies on digital ways of storing data. In some applications a mechanical storage system is needed to achieve memory functionalities. Mechanical information storage does not require a continuous energy supply and can even function in harsh environments where information storage systems are limited. Therefore it is imperative to investigate the possibilities to mechanically store information in a functional system. A information storage system can be created by using mechanical logic gates or unit cells with an interconnected network. This research could assist in the development of programmable behaviour of materials. A programmable metamaterial can be created that can change its physical properties depending on the boundary conditions.

1. Introduction

Since the beginning of time information is stored and passed on through DNA from generation to generation. Humans have stored information in order to teach one another. Their documentation started with drawings on the wall or by carving into stone. Such as the cuneiform or hieroglyphs. With more knowledge available there also became a need for calculation tools. These tools can be classified as structural memory (1) and started as simple rulers that have a certain length or differential rulers that functioned as an analog computer and eventually evolved into analytical machines. Modern-day society finds itself in the information age and relies on digital ways of storing data. For example the Hard Disk Drive (HDD) that uses magnetic data storage, the Solid State Drive (SSD) that uses flash memory or Random Access memory (RAM). However, all of these require an electrical power supply and circuit to store and retrieve Data. RAM even requires continuous energy to maintain its information.

This paper explores the possibilities to store information in a purely mechanical manner. A key advantage is that most of these systems do not require a continuous energy supply to store data. What would be the use of a memory element that can store information? For example if we look at a material that has a positive Poisson's ratio. If a force is exerted on the material, the material will stretch, due to this stretching, the cross-section of the material will decrease in size. When the same load is applied to a material with a Poisson's ratio of zero, the cross-section of the material will remain the same. In a material with a negative Poisson's ratio, the material will shrink in the longitudinal and lateral direction under the influence of the same load. What if it is possible to create a material element that can be in all three states? In the first state the material has a positive ratio, in the second state a ratio of zero and in the third state a negative ratio. In that case the unit cell will have to have some form of memory to remember which state it is in.

In harsh environments the existing functional information storage systems are limited due to low convection rates or high radiation. There is a need for devices that generate no electromagnetic signature and are insensitive to radiation damage. This requires a mechanical solution. If we make a system that is mechanical there will be other side effects to keep in mind, such as damping, friction, wear, backlash. Some of these problems can be limited by making a compliant design, preferably completely monolithic. This will reduce the amount of parts used and thereby the friction between several parts. With a monolithic design there is no lubricant needed and thus requires less maintenance than a rigid body equivalent design. Next to that there will be other production processes available that may reduce the price for a single product. Due to the reduced wear and backlash between parts, the motion will be precise and the operation of the parts will be predictable. With a monolithic design it is also easier to scale it in size depending on the application.

This brings us to the research objective: Investigate the possibilities to mechanically store information in a functional system. In order to achieve this, how information is stored in digital systems will be investigated in section 2.1. This will be followed by section 2.2 and 2.3 where will be explored if these digital systems can be translated to the mechanical domain or to systems with other working principles. Then a classification method will be presented in section 3 to identify all the mechanical information storage systems. The results of this classification method will be given in the section 4. In section 5 the metrics will be discussed, followed by a discussion in section 6 in which the knowledge gaps will be presented. In section 7 the research objective and results will be concluded.

2. Combinational Logic

Every subsection within the combinational logic section will comprise of a few parts. The history, the working principle involved in existing systems and the state of the art.

 mfhdejong@student.tudelft.nl (M.F.H.d. Jong)
ORCID(s):

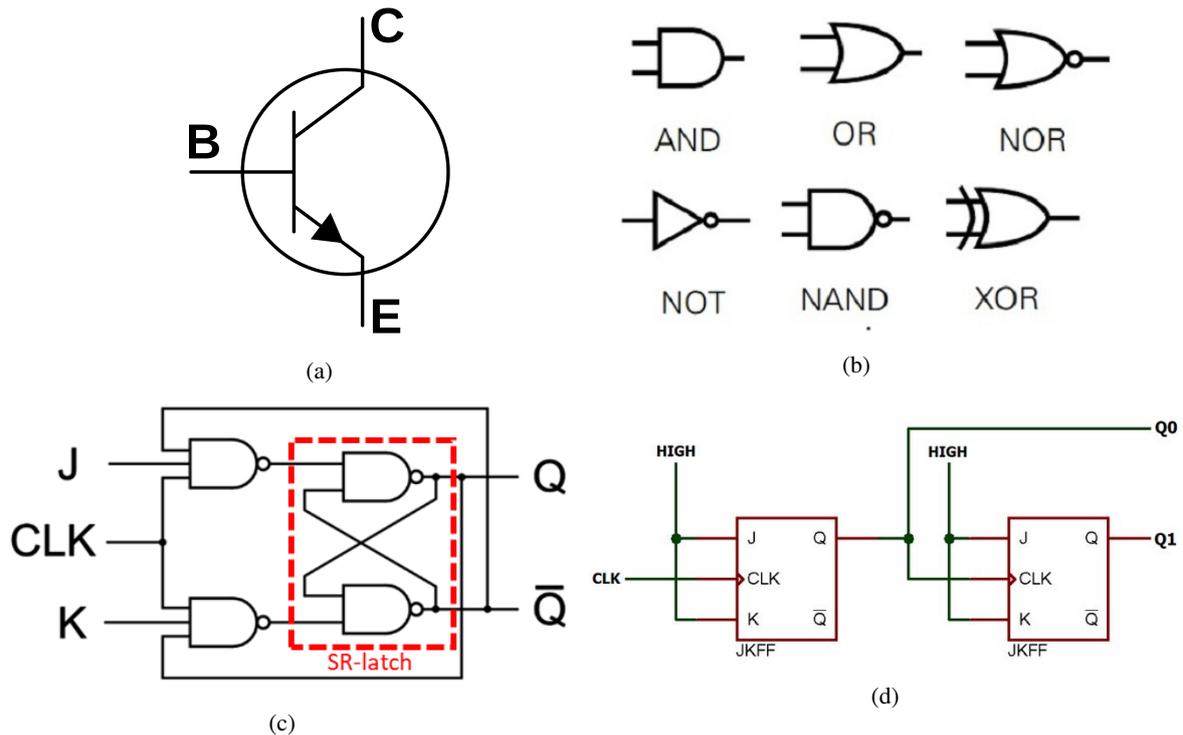


Figure 1: Building blocks for Digital systems and early stage memory systems. (a) A NPN type bipolar junction transistor consisting of a Base, Connector and an Emitter. (b) Logic gates that perform boolean operations (2). (c) JK Flipflop, the sr latch element is enclosed by a red dashed box (3) (d) Two bit counter circuit comprised of two JK flip flops, all entries of J and K are connected to a high input.

2.1. Digital systems

A combinational circuit in a digital memory system processes the input by encoding the information and determining the location where it should be stored. A combinational circuit uses Boolean functions to perform logical operations (4). The binary number system invented by Leibniz in 1703 (5) laid down the foundation of modern computing methods. This binary type of calculation called for elements that can be in two states. In an electronic system this first started with relays, that work with a electrically induced magnetic field. The switching circuits generated where further optimized by replacing the relays with vacuum tubes. These machines were still bulky but did their job. With the invention of the transistor in 1947 by Nobel prize winners in Physics William B. Shockley, John Bardeen and Walter H. Brattain (6) the systems could be further miniaturized. Since then, many transistor types have been developed, but all rely on the semiconductor operating principles. An example of a NPN type bipolar transistor is given in Figure 1a. A small current at the base terminal will change the permittivity between the collector and emitter terminals and thereby control the current flow between these terminals. These small sized switches made it possible to create integrated circuits. All of the Boolean logic operations can also be created with these type of switching elements. In Figure 1b an overview is given of all the logic gates and their representation in digital systems. With these gates the computational capabilities are endless.

These logic gates can be organized in such a manner that a memory element can be created. An application where this could be useful is in a flash storage system. That uses floating gate transistors. Another way to use logic gates is in a counting circuit that updates the output state when the input goes from low to high. An example of such a circuit is a sequential counting mechanism. This circuit can consist of several logic gates and a memory element called a Flip-Flop. The logic gates are used to represent the conditions for the output to be updated. The Flip-Flop (see Figure 1c for a JK flipflop) holds the output state when the input J and K are 0.

There are several types of flipflops that all have their advantages and disadvantages depending on the field of use. A type of flipflop has multiple ways to be created from the basic logic gates. In every flipflop there is a SR latch that consists of two nand or nor gates, where the output of a gate is the input to the other gate. A D-flipflop has two inputs, data and clock(clk), that are connected to an inverter and a nand gate next to an sr latch. A D-flipflop consists of four nand gates and an inverter in total. A JK-flipflop is similar to a d-type flipflop but it has an extra input instead of an inverter. When J and K are both high on a rising edge of a clock operation the output states Q and \bar{Q} toggle. If the propagation delay of the system is lower than the clock frequency the output states will keep switching states as long as the clock input is high. This is an undesired phenomenon called the race around condition. This problem can be solved by increasing the propagation delay or the clock frequency or

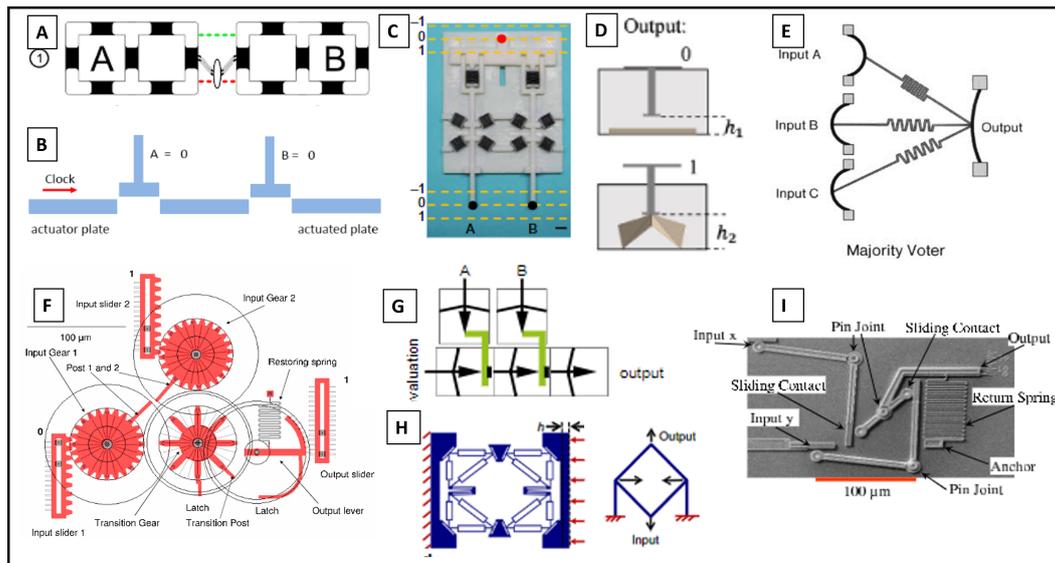


Figure 2: Overview of Mechanical logic gates. (A) A bi-stable compliant mechanism constructed of a metamaterial (9). (B) A dynamic mechanical AND gate consisting of sliders based on the design principles of Konrad Zuse's mechanical computers (10). (C) An AND gate is created with scalable tristable elements (denoted in black). These elements determine the actuation force and output amplitude. (11). (D) A bi stability-based foldable origami element is used to make a mechanical NOT gate. When extended the origami element is less resistant to bending (12). (E) A three input mechanical AND gate consisting of four bistable beams and connecting spring element (13). (F) A NAND gate for microelectromechanical systems is created by using sliders, gears and a restoring spring (14). (G) Bi-stable building blocks are used to form mechanical logic gates. The bi-stable cells have to be reset manually after actuation (15) (H) Micro mechanical logic gates are fabricated starting with a bi-stable beam element and (non-)volatile operation (16). (I) Micro-Mechanical logic gates are designed with an optical output (17).

by connecting two JK-flipflops in series and creating a master slave flipflop (4). The Flip-flop seen in Figure 1c is used to create a counter circuit in Figure 1d. In the field of digital systems there are a lot of evolving technologies such as the emerging nonvolatile memory technologies (7) and the development of memristors for memory and neuromorphic computing (8). These digital systems, apart from memristors, have one thing in common, they work with elements that can be in two states, low or high.

2.2. Mechanical

Following up on the digital systems and the logic gates that are a fundamental part of modern computation, the logic gates do not have to be electronic but can be made with other working principles. These mechanical logic gates may have the same possibilities as their electronic equivalent. As stated in the previous section, the first digital logic operations were performed with relays. This can be defined as an electromechanical system because it has an electrical input (potential difference) and a mechanical output (displacement). For mechanical logic the interest is in a purely mechanical system, thus the input and output are mechanical such as displacement, velocity, acceleration, jerk, force, etc..

The first mechanical systems that performed computations are Pascals calculator in 1645, the Arithmometer in 1820, the Differential Analyser in 1822 and the Analytical engine by Charles Babbage developed in 1837 (18). After these analytical systems the first mechanical calculators were based on a pinwheel with a carry mechanism. This

was the Comptometer in year 1887. They were further developed and ingenious memory elements were implemented in the 10-key machines. It revolutionized the industry and many memory systems followed such as the punch cards, magnetic drum or even the compact disc (19).

Nowadays, mechanical realizations of digital computing have reached a new level. Non-volatile realizations have predominantly assumed a binary form using bi-stable configurations (20). By combining Links and rotary joints any traditional 2-input logic gate, including AND, NAND, NOR, NOT, OR, XNOR and XOR can be created (21). These logic gates have also been made compliant, such as the 3-input AND port and inverter are suggested in (13). Complete computational models are created with gears, links and rotary joints in (14). In some cases the logic functions are implemented by starting with the design of a mechanical building block and extending this to a logic gate (15), (16). Most of these systems are scalable and can even be designed for MEMS applications (17). In some cases it is desired to alter physical properties by using a meta-material (9). In this example parts of the structure are more flexible than others, that will cause the structure to transition from one state to another when actuated. There are also designed re-configurable logic gates based on programmable multi-stable mechanisms (22). Where two inputs are used to adjust the stability behaviour of the mechanism, with the experimental setup they were able to create six logic operations. Depending on the application a sequential counting mechanism can be formed as shown

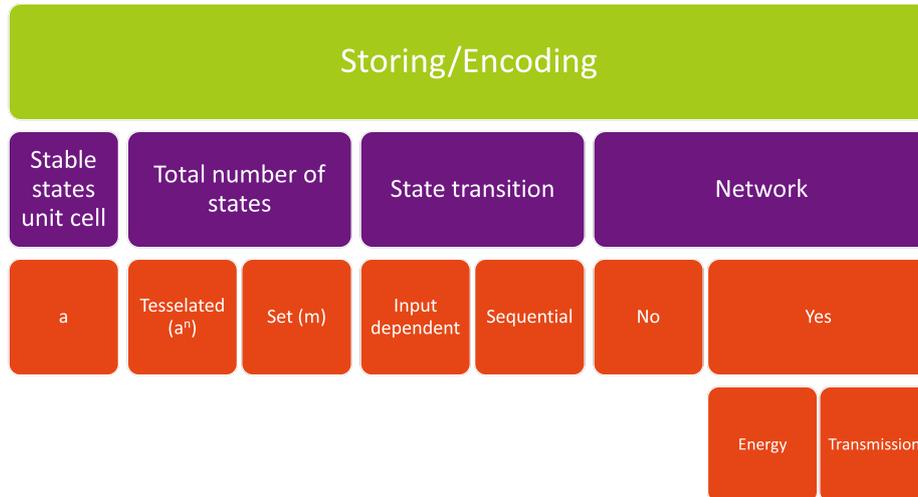


Figure 3: The classification tree, every researched memory system is placed within this classification

in Figure 1d by using the mechanical logic gates given in Figure 2.

2.3. Other domains

Next to mechanical there are other physical domains where logic gates can be present. For instance in Acoustics, Chemical, Electromechanical, Pneumatic, Magnetic and even in the Molecular domain. Depending on the application the input with a system based on acoustics can be a mechanical, thermal or electromechanical induced vibration. In (23) the response to a vibrational input is altered by pre-actuation modification of initial elements and coupling between these elements. That results in a bi-stable logic-gate elastic metamaterial to correctly execute simple wave logic operations. In (24) a system is designed that enables the propagation of a signal through soft media by using a bi-stable element array. Chemical Logic is used in (25) for targeted drug delivery. This targeted drug delivery works on the principle of hydrogels with precise degradative responsiveness by using multiple environmental cues to trigger reactions that operate user programmable Boolean Logic. In (26) are pneumatic gates that work with pressure differences and a flexible membrane that prevents flow through a certain tube. That tube is kinked or un-kinked. With this bi-stable behaviour the possibilities are endless, even a pneumatic gripper that fully functions on pressurized air is made. Another pneumatic application is a pneumatic random access memory (RAM) consisting of logic gates. An 8 bit storage system is created by 16 pneumatic gates (27). These gates operate in a similar manner as flip flops except that the system requires some energy to maintain the pressure at a certain level. In the magnetic domain a current driven magnetic domain-wall logic is proposed (28). In (29) all the state of the art molecular logic gates from the past 25 years are presented. In (30) binary logic operations are performed with artificial molecular machines. An electromechanical application is a structure that has four input configurations.

Depending on the metal tracks integrated in the structure, the system performs boolean logic operations (31).

3. Classification

In this section a memory classification method will be proposed. This classification follows from the existing literature and the distinction between the three stage of memory that is made in psychology: encoding, storing and retrieving. Encoding is defined as the initial learning of information; storage refers to maintaining information over time; retrieval is the ability to access information when needed (32). If we translate these concepts to a mechanical information storage system: encoding of the input is done by transforming the perceived input to a desired information format in order to process the information. This can be from single input to array storage, specific location to store(local or global), physical domain change, ... etc. The storage is defined as the material element that holds a certain information state. The retrieve part is how the information stored in the storage block is used for further processing or is perceived by the operator. This could also be defined as the decoding phase.

In this work the existing literature is classified in four main categories: Stable states of the unit cell, total number of states, state transition and the network. In this proposed classification method, illustrated in Figure 3, the encoding and storing is combined. It focuses on how an input is processed, how it affects an individual unit storing cell and the pathways from one storing state to another. This however does omit the physical domains/working principles, but in order to have a functional memory system the focus lies on the processing of the input and the storing itself. A system is classified as a memory system if it has a value for each purple category. The categories involved will be further explained in the following subsections.

3.1. Stable states of the unit cell

The number of states for a storing element are the feasible (stable) states that a unit storing cell can have. Several possibilities within this category are bi-stable, meta-stable, multi-stable or it could be a statically balanced system that has a non-invasive read-out. The variable for the number of states for a storing element is defined as a .

3.2. Total number of states

Depending on the number of unit cells a system has, the number of states of the complete system has a maximum. Not all configurations of a set of unit cells is possible or even desired in certain applications. The value for the maximum number of states is defined as a^n where n is the number of unit cells used. Therefore a distinction is made between systems that consist of tessellated unit cells and could reach the theoretical maximum of a^n and systems that do not have this property but have a certain set of storing states m .

3.3. State transition

The state transition is defined as the path the information takes between storing states. For a single bi-stable unit cell this is from 0 to 1. When multiple unit cells are connected this state transition is more complex and depends on the network that is embedded in the system between the unit cells. This implies that when multiple unit cells are connected there are several pathways with the same amount of states (33). Next to that the state transition depends on the combinational circuit that is between the input and the set of unit cells. Therefore there are two categories defined within the state transition, sequential and input dependent. Sequential means that consecutive inputs lead to a certain order of storing. The state transitions that lead from an input are predefined, thus the next output is dependent on the previous output. For example if we have two bi-stable unit cells the state transitions that will occur are [00, 01, 10, 11]. But it could also be predefined as [10, 00, 11, 01]. Input dependent means that a certain input leads to a specific output. Depending on the input the signal follows a certain path. This means that these systems have multiple input by definition.

3.4. Network

As stated before the network has an influence on the sequence between the states. Next to that it defines the number of states of the complete system. If a system has no network the unit cells do not interfere with one another and each individual unit cell is connected to the input. In a system with an network there are interconnections between the unit cells, these interconnections can be energy or transmission based. With energy based there is a certain threshold energy between two unit cells that needs to be met in order for one cell to change state through another cell. An example of this is given in Figure 4b. A Transmission based attached network has a certain transmission ratio or defined direction that affects a neighboring unit cell.

4. Results

In this section the results from the literature review and the classification are presented. The respective categories from the previous section are used to give a clear overview of the state of the art in mechanical memory systems. The references to the papers and their respective categories in the classification are given in the Appendix in Table 1.

4.1. Stable states of the unit cell

In Figure 4a the stable states of the classified systems are given. The stable states of the unit cell is denoted in red. As can be seen there are 45 of the 47 storing systems that have a bi-stability based unit cell. Only a few systems have a unit cell that has a larger amount of stable storing states. This may be due to the fact that existing literature did not focus on creating a unit cell but focused on other aspects, such as creating a logic gate as presented in Figure 2 or initiate a shape change (36).

4.2. Total number of states

In Figure 4a the total number of states of a system is given in green. There is an interesting shift compared to the stable states of a single unit cell. The rightmost bar consists of 19 systems that have more than five storing states and 18 of these systems are build up from bi-stable unit cells that are either tessellated or combined in such way that multi-stable behaviour is achieved. By tessellating the unit cells a higher amount of total storing states can be achieved. The limit of this tessellation is dependent on multiple factors but in a system with an embedded network this is mainly caused by energy losses.

4.3. State transition

Following from the classification there are 27 systems that have sequential state transitions and 19 systems that are input dependent. In a sequential system consecutive inputs of the system lead to a certain pathway while in an input dependent system a specific input leads to one output. In the previous section it was noted that there 18 systems are build up from tessellated bi-stable unit cells. Out of these, there are 8 systems that have input dependent state transitions and 10 systems have a sequential state transition behaviour. In some cases this sequential snapping is predefined by design (37). Once fabricated some physical properties can be altered by human interactions (38) or the overall behaviour can be altered by changing the temperature in the environment (39),(40),(41),(42),(43).

4.4. Network

Out of the investigated papers there are 23 systems that do not have a network and 23 systems that have a network. Five of the 23 network based systems are transmission based and 19 are energy based. Interesting to note about the energy based systems is that most of these are tessellated bi-stable unit cells that sequentially snap in an array (34). Moreover the networks that are present either have a predictable behaviour that is defined before production of the system,

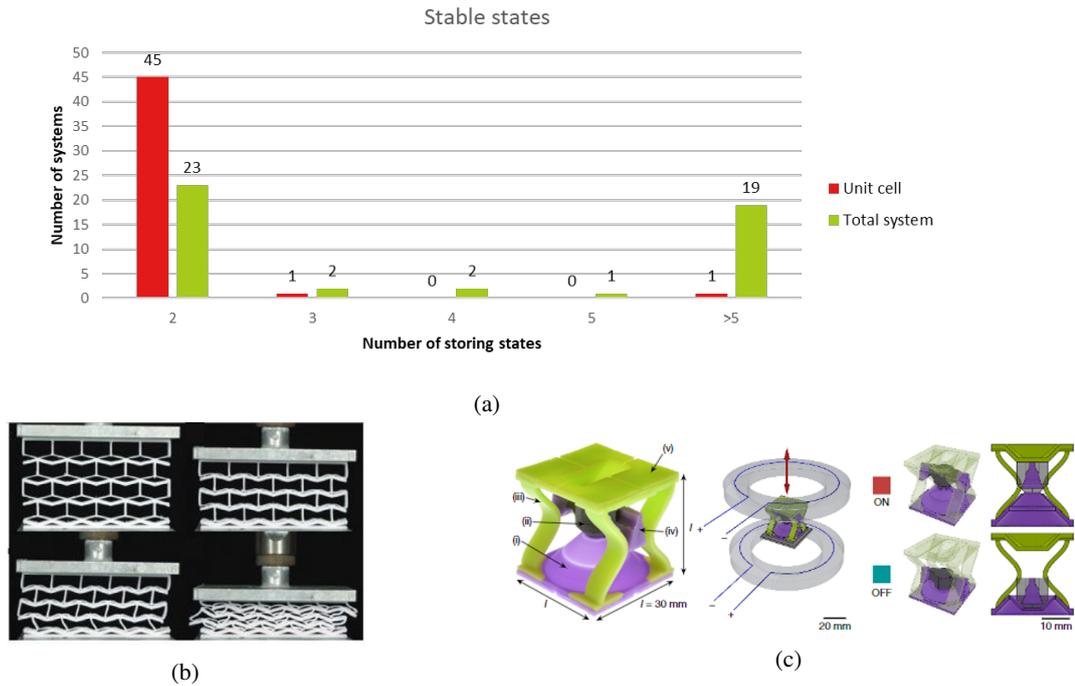


Figure 4: (a) Number of stable states of unit cells compared to the stable states of the complete memory systems found in available literature. (b) A Sequential snapping mechanism, this system is classified as bi-stable, tessellated, sequential with an energy based network (34). (c) This mechanism is classified as bi-stable, tessellated, input dependent and without a network (35)

e.g. they are not re programmable, or they require multiple inputs. For example there is a mechanical metamaterial with re programmable logical function that has a transmission based mechanism (44). However this system still requires each unit cell to be actuated independently.

5. Metrics

A mechanical information storage system will need to comply to a certain set of properties. Depending on the application these may vary. With regard to the unit cell, boundaries to set are the amount of storing states and the actuation force required to transition from one state to another. When looking at connected unit cells via a network, the threshold energy or transmission ratio between these cells is of interest. In turn, the connection between unit cells will influence the overall actuation force. If the design is made monolithic or partly compliant there will more deformation than in a rigid structure. In that case, one of the requirements for the system to be completely reversible is that all components of the system must remain in the elastic domain. If parts of the system go through plastic deformation, the repeatability of the operation will be compromised.

6. Discussion

The results indicate that there are not any multi stable unit cell while this does seem like a valuable research topic. If the stable states of a unit cell can be increased, the storing capacity of the entire material will increase exponentially. A 65

pitfall to this approach is that the multi stable unit cell must be compatible with a certain network in order to work.

If there is no network involved and each unit cell is actuated independently it would resemble a mechanical random access memory (RAM). In that case the state transitions are input dependent and the unit cells can be tailor made to function with a certain required type of actuation.

Another result is that most energy based networks are sequential snapping mechanisms. These are actuated array by array and therefore lack a certain programmable behaviour after the system has been produced. It can be said that these systems are reversible and can be reset by applying a force or displacement in the opposite direction than in the actuation phase. Next to that the transmission based mechanisms require some form of contact between unit cells to induce a state transition. This type of state transition can increase the actuation force of the entire system but can also add a programmable behaviour to the system that can't be resolved with an energy based system alone.

A functional mechanical information storage system can be created by combining mechanical logic gates as discussed in section 2.2. Since 20 logic gates are needed to create a two bit counter this may be impractical due to the energy losses between each cascaded logic gate.

Another way to create a functional mechanical information storage system is to combine the transmission based and energy based working principles. This will create a network of unit cells that has a tunable behaviour post production. First the interaction between two unit cells must be resolved and then it can be further extended to an array or a grid. In

that case the state transitions will be boundary dependent, because the system will behave differently if a network is designed that connects all the elements like a snake or a network that one unit cell only effects the neighboring unit cells. In other words the pathways that are implemented in the network are in close relation with the state transitions possible for a certain configuration of the unit cells.

7. Conclusion

A mechanical information storage system can be realized with digital or analogue storing principles. Analogue systems require a continuous energy supply and have not been explored. A digital information system can be created by using mechanical logic gates or unit cells with an interconnected network. Mechanical logic gates will result in a complex system due to the amount of gates needed. Mechanical realizations of unit cells primarily use bi-stable mechanisms and can be combined with a transmission based or energy based network.

The follow up research that will be performed is the design of a programmable network of unit cells. The unit cell has a physical property change depending on its storing state and the network determines the state transitions. This research could assist in the development of programmable behaviour of materials.

8. Bibliography

A. Tables

In Table 1 an overview of the classified memory systems is given.

References

[1] J. V. D. Vegt, H. Buffart, and C. V. Leeuwen, "Psychological research the structural memory: A network model for human perception of serial objects," 1989.

[2] N. Elprocus, "Basic logic gates with truth tables - digital logic circuits," Dec 2020.

[3] A. Dheeraj, "JK flip flop explained in detail - dcaclab blog," Jan 2020.

[4] S. Brown and Z. Vranesic, *Fundamentals of Digital Logic with VHDL Design*, vol. 3rd edition. London: Raghathan Srinivasan, 2009.

[5] D. R. Lande, "Development of the binary number system and the foundations of computer science," *Mathematics Enthusiast*, vol. 11, pp. 513–540, 2014.

[6] "The nobel prize in physics 1956."

[7] J. S. Meena, S. M. Sze, U. Chand, and T. Y. Tseng, "Overview of emerging nonvolatile memory technologies," *Nanoscale Research Letters*, vol. 9, pp. 1–33, 2014.

[8] A. S. Sokolov, H. Abbas, Y. Abbas, and C. Choi, "Towards engineering in memristors for emerging memory and neuromorphic computing: A review," *Journal of Semiconductors*, vol. 42, p. 013101, 1 2021.

[9] U. Waheed, C. W. Myant, and S. N. Dobson, "Boolean and/or mechanical logic using multi-plane mechanical metamaterials," *Extreme Mechanics Letters*, vol. 40, 10 2020.

[10] R. Rojas, "The design principles of konrad zuse's mechanical computers," 2015.

[11] H. Zhang, J. Wu, D. Fang, and Y. Zhang, "Hierarchical mechanical metamaterials built with scalable tristable elements for ternary logic operation and amplitude modulation," 2021.

Table 1
Overview of classified memory systems

Category	References
Stable states unit cell	
2	(16),(24),(9),(26),(35),(45),(46),(12),(47),(48),(49),(50),(51),(52),(53),(54),(55),(17),(14),(15),(13),(23),(34),(56),(57),(58),(59),(60),(39),(61),(62),(63),(27),(40),(64),(41),(42),(65),(66),(67),(68),(69),(70),(71),(72),(44)
3	(22)
≥ 5	(11)
Total number of states	
2	(16),(24),(9),(22),(15),(48),(51),(53),(17),(14),(13),(58),(39),(61),(62),(63),(40),(64),(42),(66),(12),(71)
3	(45),(54)
4	(34),(67)
≥ 5	(72),(26),(35),(11),(47),(49),(70),(44),(23),(27),(52),(55),(56),(57),(59),(60),(41),(65),(69),(68)
State transition	
Input dependent	(16),(24),(9),(26),(35),(11),(22),(47),(48),(49),(54),(17),(14),(15),(13),(23),(27),(70),(44)
Sequential	(45),(12),(51),(52),(53),(55),(34),(56),(57),(58),(59),(54),(39),(61),(62),(63),(40),(64),(41),(42),(65),(66),(67),(68),(69),(71),(72)
Network	
No	(16),(9),(26),(35),(11),(22),(48),(51),(53),(54),(17),(14),(13),(57),(58),(39),(61),(62),(63),(40),(42),(66),(71)
Yes	
Energy	(24),(45),(47),(49),(52),(55),(15),(34),(56),(59),(60),(64),(41),(65),(69),(70),(72),(44)
Transmission	(12),(23),(27),(67),(68)

[12] Z. Meng, W. Chen, T. Mei, Y. Lai, Y. Li, and C. Q. Chen, "Bistability-based foldable origami mechanical logic gates," *Extreme Mechanics Letters*, vol. 43, 2 2021.

[13] R. C. Merkle, "Two types of mechanical reversible logic," 1990.

[14] K. C. Bradley, "Part of the electro-mechanical systems commons recommended citation recommended citation bradley," 2003.

[15] A. Ion, L. Wall, R. Kovacs, and P. Baudisch, "Digital mechanical metamaterials," vol. 2017-May, pp. 977–988, Association for Computing Machinery, 5 2017.

[16] Y. Song, R. M. Panas, S. Chizari, L. A. Shaw, J. A. Jackson, J. B. Hopkins, and A. J. Pascall, "Additively manufacturable micro-mechanical logic gates," *Nature Communications*, vol. 10, 12 2019.

[17] P. Kladitis, "Mems micro-mechanical logic gates for mechanical computing in machine only environments," pp. 215–218, Transducer Research Foundation Inc. (TRF), 1 1999.

[18] G. Tweedale, "Martin campbell-kelly (ed.). the works of charles babbage. london: Pickering chatto (publishers) ltd, 1989. 11 vols. 66

- isbn 1-85196-005-8. £500.," *The British Journal for the History of Science*, vol. 22, no. 4, p. 481–482, 1989.
- [19] "Memory amp; storage: Timeline of computer history: Computer history museum."
- [20] H. Yasuda, P. R. Buskohl, A. Gillman, T. D. Murphey, S. Stepney, R. A. Vaia, and J. R. Raney, "Mechanical computing," *Nature*, vol. 598, pp. 39–48, 10 2021.
- [21] R. C. Merkle, R. A. Freitas, T. Hogg, T. E. Moore, M. S. Moses, and J. Ryley, "Mechanical computing systems using only links and rotary joints," 1 2018.
- [22] M. Zanaty, H. Schneegans, I. Vardi, and S. Henein, "Reconfigurable logic gates based on programable multistable mechanisms," *Journal of Mechanisms and Robotics*, vol. 12, 1 2020.
- [23] Z. Ren, L. Ji, R. Tao, M. Chen, Z. Wan, Z. Zhao, and D. Fang, "Smp-based multi-stable mechanical metamaterials: From bandgap tuning to wave logic gates," *Extreme Mechanics Letters*, vol. 42, 1 2021.
- [24] J. R. Raney, N. Nadkarni, C. Daraio, D. M. Kochmann, J. A. Lewis, and K. Bertoldi, "Stable propagation of mechanical signals in soft media using stored elastic energy," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 113, pp. 9722–9727, 8 2016.
- [25] B. A. Badeau, M. P. Comerford, C. K. Arakawa, J. A. Shadish, and C. A. Deforest, "Engineered modular biomaterial logic gates for environmentally triggered therapeutic delivery," *Nature Chemistry*, vol. 10, pp. 251–258, 3 2018.
- [26] D. J. Preston, P. Rothmund, H. J. Jiang, M. P. Nemitz, J. Rawson, Z. Suo, and G. M. Whitesides, "Digital logic for soft devices," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 116, pp. 7750–7759, 4 2019.
- [27] S. Hoang, K. Karydis, P. Brisk, and W. H. Grover, "A pneumatic random-access memory for controlling soft robots," *PLoS ONE*, vol. 16, 7 2021.
- [28] Z. Luo, A. Hrabec, T. P. Dao, G. Sala, S. Finizio, J. Feng, S. Mayr, J. Raabe, P. Gambardella, and L. J. Heyderman, "Current-driven magnetic domain-wall logic," *Nature*, vol. 579, pp. 214–218, 3 2020.
- [29] S. Erbas-Cakmak, S. Kolemen, A. C. Sedgwick, T. Gunnlaugsson, T. D. James, J. Yoon, and E. U. Akkaya, "Molecular logic gates: The past, present and future," 4 2018.
- [30] F. Nicoli, E. Paltrinieri, M. T. Bakić, M. Baroncini, S. Silvi, and A. Credi, "Binary logic operations with artificial molecular machines," 2 2021.
- [31] C. E. Helou, P. R. Buskohl, C. E. Tabor, and R. L. Harne, "Digital logic gates in soft, conductive mechanical metamaterials," *Nature Communications*, vol. 12, 12 2021.
- [32] A. W. Melton, "Implications of short-term memory for a general theory of memory," *Journal of Verbal Learning and Verbal Behavior*, vol. 2, no. 1, pp. 1–21, 1963.
- [33] M. V. Hecke, "Profusion of transition pathways for interacting hysterons," *Physical Review E*, vol. 104, 11 2021.
- [34] Z. Meng, M. Liu, Y. Zhang, and C. Q. Chen, "Multi-step deformation mechanical metamaterials," *Journal of the Mechanics and Physics of Solids*, vol. 144, 11 2020.
- [35] T. Chen, M. Pauly, and P. M. Reis, "A reprogrammable mechanical metamaterial with stable memory," *Nature*, vol. 589, pp. 386–390, 1 2021.
- [36] H. Fu, K. Nan, W. Bai, W. Huang, K. Bai, L. Lu, C. Zhou, Y. Liu, F. Liu, J. Wang, M. Han, Z. Yan, H. Luan, Y. Zhang, Y. Zhang, J. Zhao, X. Cheng, M. Li, J. W. Lee, Y. Liu, D. Fang, X. Li, Y. Huang, Y. Zhang, and J. A. Rogers, "Morphable 3d mesostructures and microelectronic devices by multistable buckling mechanics," *Nature Materials*, vol. 17, pp. 268–276, 3 2018.
- [37] K. Che, C. Yuan, J. Wu, H. J. Qi, and J. Meaud, "Three-dimensional-printed multistable mechanical metamaterials with a deterministic deformation sequence," *Journal of Applied Mechanics, Transactions ASME*, vol. 84, 1 2017.
- [38] H. Tao, F. Danzi, C. E. Silva, and J. M. Gibert, "Heterogeneous digital stiffness programming," 8 2021.
- [39] S. Ma, Z. Jiang, M. Wang, L. Zhang, Y. Liang, Z. Zhang, L. Ren, and L. Ren, "4d printing of pla/pcl shape memory composites with controllable sequential deformation," *Bio-Design and Manufacturing*, vol. 4, pp. 867–878, 12 2021.
- [40] L. Cera, G. M. Gonzalez, Q. Liu, S. Choi, C. O. Chantre, J. Lee, R. Gabardi, M. C. Choi, K. Shin, and K. K. Parker, "A bioinspired and hierarchically structured shape-memory material," *Nature Materials*, vol. 20, pp. 242–249, 2 2021.
- [41] X. Xin, L. Liu, Y. Liu, J. Leng, X. Xin, L. Liu, Y. Liu, and J. Leng, "4d pixel mechanical metamaterials with programmable and reconfigurable properties," *Advanced Functional Materials*, p. 2107795, 10 2021.
- [42] M. Wan, K. Yu, and H. Sun, "4d printed programmable auxetic metamaterials with shape memory effects," *Composite Structures*, vol. 279, p. 114791, 1 2022.
- [43] J. Mueller, J. A. Lewis, K. Bertoldi, J. Mueller, J. A. Lewis, K. Bertoldi, and J. A. Paulson, "Architected multimaterial lattices with thermally programmable mechanical response," *Advanced Functional Materials*, p. 2105128, 2021.
- [44] T. Mei, Z. Meng, K. Zhao, and C. Q. Chen, "A mechanical metamaterial with reprogrammable logical functions," *Nature Communications*, vol. 12, 2021.
- [45] Z. Zhang, S. Pusateri, B. Xie, and N. Hu, "Tunable energy trapping through contact-induced snap-through buckling in strips with programmable imperfections," *Extreme Mechanics Letters*, vol. 37, 5 2020.
- [46] J. A. Faber, J. P. Udani, K. S. Riley, A. R. Studart, and A. F. Arrieta, "Dome-patterned metamaterial sheets," *Advanced Science*, vol. 7, 11 2020.
- [47] M. Serra-Garcia, "Turing-complete mechanical processor via automated nonlinear system design," *Physical Review E*, vol. 100, 10 2019.
- [48] P. R. Kuppens, M. A. Bessa, J. L. Herder, and J. B. Hopkins, "Monolithic binary stiffness building blocks for mechanical digital machines," *Extreme Mechanics Letters*, vol. 42, 1 2021.
- [49] T. Jules, A. Reid, K. E. Daniels, M. Mungan, and F. Lechenault, "The delicate memory structure of coupled origami switches," 6 2021.
- [50] T. C. Draper, C. Fullarton, N. Phillips, B. P. de Lacy Costello, and A. Adamatzky, "Mechanical sequential counting with liquid marbles," vol. 10867 LNCS, pp. 59–71, Springer Verlag, 2018.
- [51] Y. Jiang, L. M. Korpas, and J. R. Raney, "Bifurcation-based embodied logic and autonomous actuation," *Nature Communications*, vol. 10, 12 2019.
- [52] N. Kidambi and K. W. Wang, "Dynamics of kresling origami deployment," *Physical Review E*, vol. 101, 6 2020.
- [53] T. G. Sano and H. Wada, "Snap-buckling in asymmetrically constrained elastic strips," *Physical Review E*, vol. 97, 1 2018.
- [54] N. Yang, C. W. Chen, J. Yang, and J. L. Silverberg, "Emergent reconfigurable mechanical metamaterial tessellations with an exponentially large number of discrete configurations," *Materials and Design*, vol. 196, 11 2020.
- [55] B. Trembl, A. Gillman, P. Buskohl, and R. Vaia, "Origami mechanologic," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 115, pp. 6916–6921, 7 2018.
- [56] A. Rafsanjani, A. Akbarzadeh, and D. Pasini, "Snapping mechanical metamaterials under tension," *Advanced Materials*, vol. 27, pp. 5931–5935, 10 2015.
- [57] F. Pan, Y. Li, Z. Li, J. Yang, B. Liu, and Y. Chen, "3d pixel mechanical metamaterials," *Advanced Materials*, vol. 31, 6 2019.
- [58] M. F. Berwind, A. Kamas, and C. Eberl, "A hierarchical programmable mechanical metamaterial unit cell showing metastable shape memory," *Advanced Engineering Materials*, vol. 20, 11 2018.
- [59] H. Yang and L. Ma, "Multi-stable mechanical metamaterials by elastic buckling instability," *Journal of Materials Science*, vol. 54, pp. 3509–3526, 2 2019.
- [60] H. Yang and L. Ma, "1d and 2d snapping mechanical metamaterials with cylindrical topology," *International Journal of Solids and Structures*, vol. 204–205, pp. 220–232, 11 2020.

- [61] G. D. Cole and M. Aspelmeyer, "Cavity optomechanics: Mechanical memory sees the light: A nanomechanical beam coupled to an optical cavity can be operated as a non-volatile memory element," *Nature Nanotechnology*, vol. 6, pp. 690–691, 2011.
- [62] H. Noh, S. B. Shim, M. Jung, Z. G. Khim, and J. Kim, "A mechanical memory with a dc modulation of nonlinear resonance," *Applied Physics Letters*, vol. 97, 7 2010.
- [63] M. A. A. Hafiz, L. Kosuru, A. Ramini, K. N. Chappanda, and M. I. Younis, "In-plane mems shallow arch beam for mechanical memory," *Micromachines*, vol. 7, 10 2016.
- [64] L. M. Korpas, R. Yin, H. Yasuda, and J. R. Raney, "Temperature-responsive multistable metamaterials," *ACS Applied Materials and Interfaces*, vol. 13, pp. 31163–31170, 7 2021.
- [65] X. Tan, S. Chen, B. Wang, J. Tang, L. Wang, S. Zhu, K. Yao, and P. Xu, "Real-time tunable negative stiffness mechanical metamaterial," *Extreme Mechanics Letters*, vol. 41, p. 100990, 11 2020.
- [66] L. Medina, R. Gilat, and S. Krylov, "Latching in bistable electrostatically actuated curved micro beams," *International Journal of Engineering Science*, vol. 110, pp. 15–34, 1 2017.
- [67] Y. Zhang, M. Tichem, and F. van Keulen *International Journal of Mechanical Sciences*, 3.
- [68] Y. Zhang, Q. Wang, M. Tichem, and F. van Keulen, "Design and characterization of multi-stable mechanical metastructures with level and tilted stable configurations," *Extreme Mechanics Letters*, vol. 34, p. 100593, 1 2020.
- [69] T. R. Giri and R. Mailen, "Controlled snapping sequence and energy absorption in multistable mechanical metamaterial cylinders," *International Journal of Mechanical Sciences*, vol. 204, p. 106541, 8 2021.
- [70] H. Ma, K. Wang, H. Zhao, R. Mu, and B. Yan, "A reusable metastructure for tri-directional energy dissipation," *International Journal of Mechanical Sciences*, vol. 214, p. 106870, 1 2022.
- [71] J. Qiu, J. H. Lang, and A. H. Slocum, "A curved-beam bistable mechanism," *Journal of Microelectromechanical Systems*, vol. 13, pp. 137–146, 4 2004.
- [72] S. Chen, X. Tan, J. Hu, S. Zhu, B. Wang, L. Wang, Y. Jin, and L. Wu, "A novel gradient negative stiffness honeycomb for recoverable energy absorption," *Composites Part B: Engineering*, vol. 215, p. 108745, 6 2021.

F

Design variations

Unequal preload

Figure F.1a presents a figure of an unequal preload as a result of inaccuracies in the fabrication process or setting up the boundary conditions in the experiments. In Figure F.1a the varied parameter is presented and in Figure F.1b the resulting force displacement graphs for top and bottom beam actuation is presented. In this figure the contact location of the input mechanism is not taken into account. It can be seen that the absolute input displacement required is equal. But the input displacement required for contact with the switching mechanism is different. This difference in contact point is not observed in the experiments. It appears that if there is an unequal preload between the top and bottom beam it will not be as presented in Figure F.1a. The hypothesis is that the difference in preload is on the right side of the mechanism. This parameter variation is not embedded in the PRBM because it would require a change of the defined vector loop for the link between the two beams.

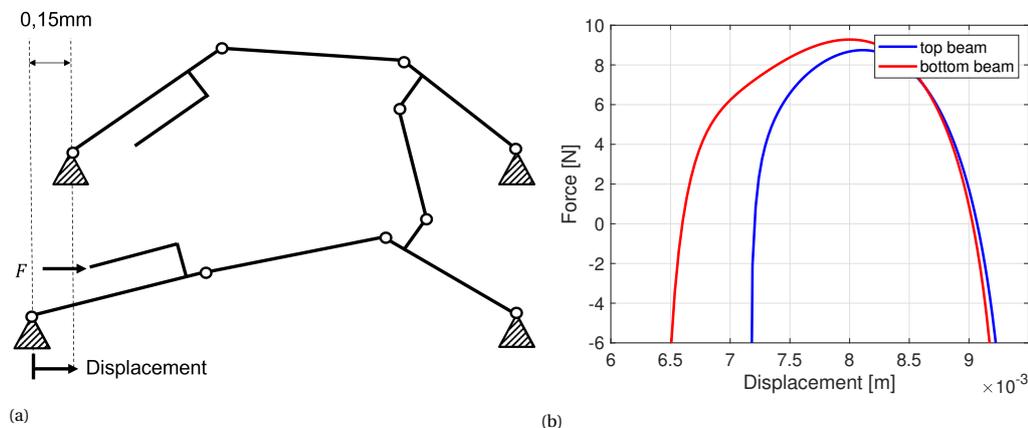


Figure F.1: Difference in preload top and bottom beam a) Schematic implementation of preload difference b) Resulting force displacement curves for the actuation of the top and bottom beam

Tip contact

Figure F.2 presents close up pictures of the input mechanism. In this figure it can be seen that the hook on the top side of the mechanism is slightly rounded towards the middle. At the bottom of the input mechanism this inside corner is relatively sharp. If the bottom beam is actuated it makes clear contact with the input mechanism. If the top beam is actuated it makes contact with the tip of the input mechanism. This tip contact may be caused by the parasitic motion of the input mechanism. If the preload is reduced the required input displacement for contact with the pushing rods increases.

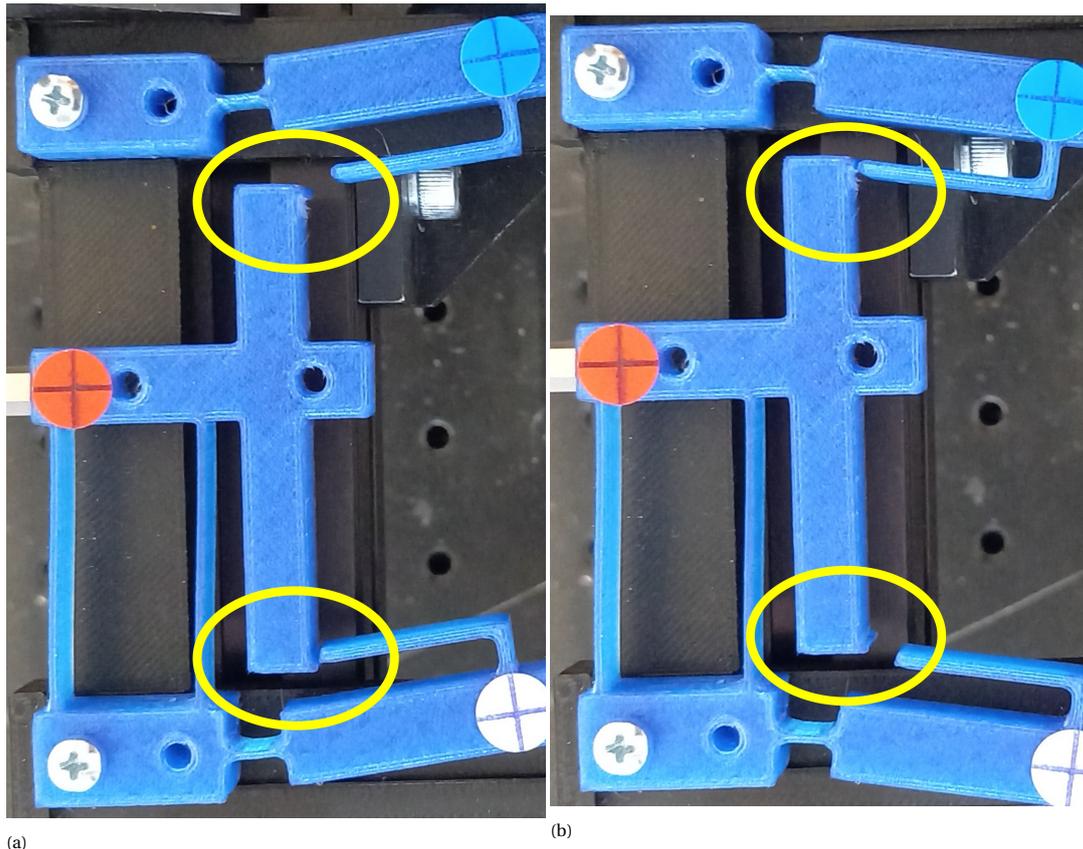


Figure E2: Close up of input mechanism. The relative position of the pushing rods with respect to the input mechanism is of interest. a) bot actuation makes clear contact with the mechanism b) top actuation makes contact with the pushing rod at the tip of the input mechanism