A Neutrally Stable Quasi-Compliant Spherical Joint With a Remote Center of Rotation

D.C.W. Hogervorst

TU Delft

A Neutrally Stable Quasi-Compliant Spherical Joint With a Remote Center of Rotation

by

D.C.W. Hogervorst

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday March 8, 2022 at 14:00.

Student number:	4460383	
Project duration:	April 26, 2021 – March 8, 2022	
Thesis committee:	Prof. dr. ir. J.L. Herder,	TU Delft, Chair
	Dr. ir. G. Radaelli,	TU Delft, Supervisor
	Dr. ir. W. Mugge,	TU Delft

This thesis is confidential and cannot be made public until March 8, 2023.

TU Delft

Preface

After spending the entirety of my adult life so far as a student, this thesis finally marks the end of this chapter. During these years I have gained a lot of knowledge and have had many experiences for which I am grateful. The process of doing a thesis project is a remarkably good summary of all the previous years of studying combined. In both there were times of struggle, stress and pressure, but also breakthroughs, joy and fulfillment. Doing a thesis in the midst of the Corona pandemic definitely made it all more challenging. Although, in the end, it makes me more confident to know that even in difficult times, I was able to complete something as demanding as a thesis project. Of course, without the help of others this project might not have been possible.

First of all, I would like to express my gratitude to my supervisor Giuseppe for the many friendly meetings and insightful discussions that ultimately guided me on the path of my project. To Ali, Just and Werner as well for your feedback and advice in the group meetings. Furthermore I would like to thank my fellow graduate students, for sharing their thoughts and helping each other whenever needed, but also for the casual conversations outside of our projects. I owe further gratitude to Jelle Rommers, for allowing me to built my thesis on the foundations of his research and helping me get started. To Sjaak Kok from Yumen Bionics as well, for giving my project direction in the early stages and for his useful and enthousiastic feedback. To Patrick van Holst, Jos van Driel and the technical support staff, thank you all for helping me get set up in the lab and successfully perform my experiments. Last but not least, a big thanks to my family, friends and girlfriend, for always being there and providing me with much needed love, support and laughter.

Dion Hogervorst
Delft, March 2022

Summary

Certain muscular disorders, such as Duchenne muscular dystrophy, dramatically and progressively lower the amount of muscle mass within the human body. This is detrimental to the quality of everyday life of the patient, as common actions can become challenging at a certain point. To alleviate part of this problem, an exoskeleton can be used that supports the weight of the arms. As a result, the remaining muscle strength is free to be used for motion directly, without having to overcome gravity first.

In an attempt to improve the current exoskeleton technology, specifically of the shoulder joint, this thesis presents the design of a quasi-compliant spherical joint which is neutrally stable in two degrees of freedom and has a remote center of rotation. The foundation of this paper lies in previous work that has succeeded in designing a compliant spherical joint with a remote center of rotation, but without neutral stability. Therefore, the existing joint is adapted in order to achieve neutral stability, while also maintaining its other properties.

A model is made with ANSYS that allows rotation of the joint to be simulated in any desired direction from its neutral position. From this model, an energy field is acquired by simulating rotation of the end-effector to every location in its range of motion and measuring the energy within the joint. Neutral stability is achieved when the energy field is constant in the specified range of motion. As a first step to neutral stability, the joint is optimized for axi-symmetry in its energy field. A spring is then introduced that has a reversed energy field relative to the joint. It is optimized such that the combined energy field of both spring and joint, is as constant as possible within the range of motion, thereby achieving neutral stability.

The optimized results reveal that both axi-symmetry of the energy field and near neutral stability can be achieved by the joint. In order to verify the validity of the simulations, a prototype is constructed. Experimental validation of the simulation has been achieved through this prototype. Neutral stability, measured as the percentual moment reduction, is improved by 83.69% in the prototype compared to 89.71% predicted by the simulations.

Contents

1 Introduction

Muscular disorders that dramatically lower the body's strength can be detrimental to a patient's quality of everyday life. Perhaps the most known example of such a muscular disorder is Duchenne muscular dystrophy. It is a genetic disorder that is predominantly found in males and causes progressive loss of muscle mass starting at the moment of birth [1]. At a certain age, the muscles that power general body movement degenerate to a point where it is no longer possible to support the body's own weight. A partial solution to this problem exists in the form of a wearable exoskeleton that compensates the weight of the body, specifically of the arms. This allows the remaining muscle strength to be used for motion directly, without the need to overcome gravity. Such an exoskeleton currently exists and is commercially available [2]. However, improvements can still be made to the exoskeleton, for example in the external shoulder joint. Like any conventional joint, friction is present that resists motion to some degree. Because of the limited muscle force that is available, any friction within the joint is detrimental and should be avoided. Moreover, the overall size of the joint could be reduced, which would be an improvement in terms of ergonomic and aesthetic aspects. Therefore, an improved external shoulder joint could be made that experiences no friction and fits around the body more compactly.

To achieve this, the external shoulder joint can be made as a compliant mechanism. In contrast to the motion of the more conventional rigid body joints, compliant joints achieve motion through elastic deformation [3]. This gives compliant joints a number of advantages, such as the ability to displace without friction or backlash. Moreover, compliant joints have the additional advantage of requiring less parts to function and can be designed to be more compact as a result. These properties make compliancy an appealing aspect to be used in joint applications. However, while a compliant joint does not experience friction, it does still encounter unwanted resistance to the applied elastic deformation. This can be mitigated by designing it to be neutrally stable. Neutral stability is defined as a constant energy within the system for any configuration in the specified range of motion [4]. In practice this means that, ideally, no external work is required to displace the joint [5]. The human shoulder joint has three rotational degrees of freedom. To perfectly allow for the natural range of motion, the external shoulder joint would need three neutrally stable degrees of freedom as well. As a first leap towards this goal, neutral stability in two rotational degrees of freedom will be the target. Additionally, the center of rotation of the external joint must coincide with that of the human shoulder joint to avoid forces induced by a center mismatch. Thus, the external joint must rotate around a remote center that is not part of the mechanism.

The goal of this thesis is to design a compliant spherical joint which can achieve neutral stability in two rotational degrees of freedom and has a remote center of rotation. This aim is directly inspired and motivated by the hope to advance the effectiveness of exoskeleton technologies that alleviate the negative impact of degenerative muscular disorders. Moreover, the novel joint design presented in this thesis can perhaps be of interest in other fields as well.

2

Paper

The main contribution of the thesis is presented in the form of this paper. It covers the design of a joint with the aforementioned properties that could be implemented as an improved exoskeleton shoulder joint. While the thesis as a whole has the specific field of exoskeleton technology in mind, this paper approaches the subject from a broader perspective and aims for a general-purpose joint with the aforementioned properties. As a result, a wider range of relevant applications may be able to use this joint.

A Neutrally Stable Quasi-Compliant Spherical Joint With a Remote Center of Rotation

Dion Hogervorst, Giuseppe Radaelli and Just Herder

***Abstract*—This paper presents the design of a compliant spherical joint which is neutrally stable in two DoF and has a remote center of rotation. Such a joint can be used, for example, as an exoskeleton's shoulder joint. The foundation of this paper lies in previous work that has succeeded in designing a compliant spherical joint with a remote center of rotation, but without neutral stability. The existing joint is simulated and its energy properties are analysed. Thereafter it is adapted and optimized for an axi-symmetrical energy field. A spring is introduced to the joint and optimized such that the combined energy field of both spring and joint, is neutrally stable. Experimental verification of the simulation was achieved with a prototype for which a moment reduction of 83.69% was achieved through the addition of the spring.**

I. Introduction

In contrast to the motion of the more conventional rigid body joints, compliant joints achieve motion through elastic deformation [1]. This gives compliant joints a number of advantages, such as the ability to displace without friction or backlash. Moreover, compliant joints have the additional advantage of requiring less parts to function and can be designed to be more compact as a result. These properties make compliancy an appealing aspect to be used in joint applications. However, while a compliant joint does not experience friction, it does still encounter resistance to the applied elastic deformation, which is often unwanted when used as a joint. This can be mitigated by designing it to be neutrally stable, also known as static balancing. Neutral stability is defined as a constant energy within the system for any configuration in the specified range of motion [2]. In practice this means that, ideally, no external work is required to displace the joint [3]. If the range of neutral stability can be extended to two rotational degrees of freedom, a joint is acquired that is stable in any configuration of its spherical range of motion. When it also rotates about a center of rotation that is not located within the physical joint, it is said to have a remote center of rotation. Combining the properties of compliancy, neutral stability, spherical motion and a remote center of rotation into a single joint could be useful in certain applications. For example, in the field of exoskeletons as an external shoulder joint.

Compliant joints have been developed that contain some of these properties. Spherical compliant joints with a remote center of rotation, for instance, have been developed in decent numbers. A spherical chain consisting of two spherical flexures that allow two dimensional rotation about a remote center of rotation has been developed by Parvari Rad et al. [4]. The same

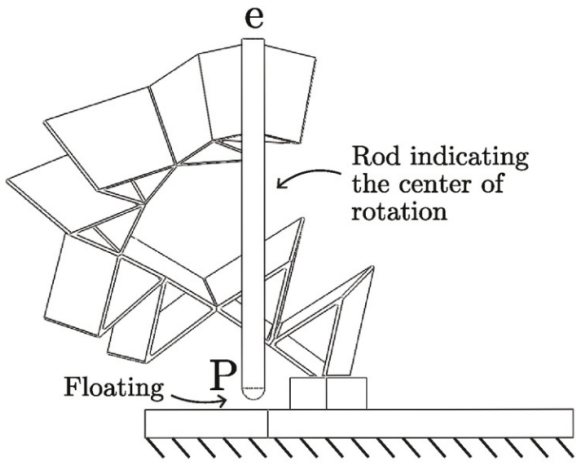


Fig. 1: The spherical joint 'Tetra I'. Retrieved from Rommers et al. [6].

principle has been applied in a three dimensional spherical joint where another chain of spherical flexures is nested in parallel that increases spherical performance and off-axis stiffness [5]. However, this parallel nesting also surrounds the remote center of rotation with joint components. A different design by Rommers et al. employs tetrahedra in series to construct a spherical joint with a remote center of rotation and high off-axis stiffness [6]. Another design is based on a spherical four-bar linkage with large rigid sections and small lumped compliance flexures that provide the relative motion between the links [7]. Rarely, research has been done on a compliant joint which combines a remote center of rotation and neutral stability for a single degree of freedom. Chandrasekaran et al. have achieved this with a linkage that displaces the center of rotation to a remote location. It is connected by compliant cruciform flexures and is statically balanced by a pretensioned serpentine flexure [8]. Hampali et al. created a tunable joint made out of open section shells of which the rotation is constrained by a guideway. Depending on the guideway and the amount of open section shells, the rotation can be forced around a remote center of rotation and made neutrally stable [9]. As of yet, a spherical compliant joint with neutral stability has been attempted only once [10]. The design is based on a number of pretensioned cables that are attached to an otherwise free-floating end-effector that lies in the center of rotation. By tuning the pretension in the cables, local zero stiffness behaviour was found for

small deflection angles. This design does not have a remote center of rotation however. A compliant joint which contains all previously mentioned properties combined, has not been reported as of yet.

This paper presents the design of a compliant spherical joint which can achieve neutral stability in two rotational degrees of freedom and has a remote center of rotation. The existing compliant joint made by Rommers et al., called the Tetra I, shows excellent properties in terms of remote center of rotation, the range of spherical motion and off-axis stiffness (Fig. 1) [6]. Because of this, the novel joint presented in this paper uses the design principles of the Tetra I as its foundation and aims to create a neutrally stable version of it.

The paper is structured as follows: First, section II elaborates on the principles behind the Tetra I and the methods used in the simulated model, solution strategy, optimizations and experimental validation. The results of the optimizations and experimental validation are shown in section III and discussed in section IV. Finally, section V summarises and concludes the paper.

II. Method

In this section, the Tetra I concept is first described in its original form as designed by Rommers et al. [6]. From there, a model is made to obtain and visualize its energy properties and a solution strategy is devised. Using the same model, an optimization is carried out to find the best result for the chosen solution strategy. Finally, a prototype is built to experimentally verify the concept and evaluate its effectiveness compared to the simulated model.

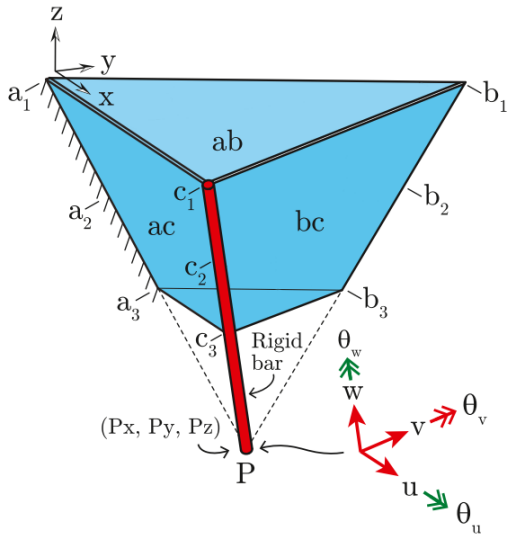


Fig. 2: A single tetraderon and its DoF, consisting of three blade flexure sides. Adapted from Rommers et al. [6].

A. Initial Tetra I design

The Tetra I joint (Fig. 1) was first designed by Rommers et al. as a high precision alternative to the more traditional ball-and-socket joint [6]. It is designed to achieve reliable spherical rotation about a practically constant remote center of rotation. This joint consists of two approximately semi-circular arms which are angled relative to each other, built from a number of tetrahedron elements. Each tetrahedron has two rotational degrees of freedom and consists of three blade flexure parts (Fig. 2), one of which is directly a part of the joint arm. To free the rotational DoF needed for spherical motion, the two joint arms must be angled relative to each other. All tetrahedra of both arms have their blade flexure constraint lines coincide in the same remote center of rotation (P). This configuration constrains all translations and allows all rotations about the remote center of rotation.

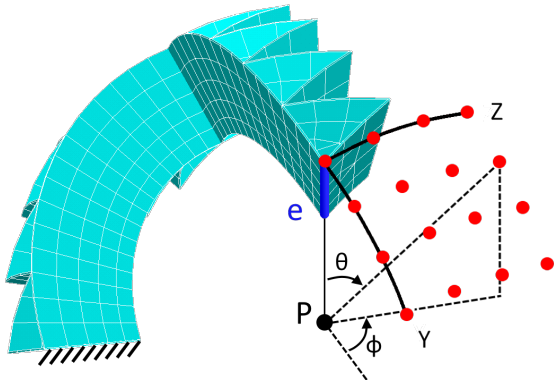


Fig. 3: Visualization of the spherical coordinate system, including examples of data points (red) to which the end-effector can be rotated.

B. Simulated model

The Tetra I joint is modelled using ANSYS APDL. Its arms are constructed as shell elements (SHELL181) with a generated mesh of approximately 4 nodes per square cm of shell surface. Since the simulated deflections are relatively large, geometric non-linearity is included in the model. Throughout this paper, titanium alloy (Ti6Al4V) will be used as material for the joint. The weight of the joint is not included in the initial simulation. A rigid link connects the end-effector (e) of the joint to its center of rotation (P), as can be seen in Fig. 3.

The model allows the end-effector to be rotated around P in any direction from the neutral position. Every location it can reach therefore lies on a spherical surface. Any location on this spherical surface is reached through a forced rotation θ, of which the direction is defined by the angle ϕ. The forced rotation is applied in the center of rotation and directly transmitted to the end-effector via the rigid link, while rotation of the end-effector around the rigid link is left free. The location of the end-effector on the spherical surface is defined by a spherical coordinate system. It is a fixed coordinate system

with two axes, Y and Z. Both axes have their origin in the neutral position of the joint. For every forced rotation toward a desired location on the spherical surface, the end-effector starts in the neutral position. Thereafter, it is rotated directly to the desired location, with a magnitude and direction corresponding to that specific location. Any other rotation that would prevent the end-effector from reaching the desired location on the spherical surface is constrained. All translational degrees of freedom are left free and are only constrained by the geometry of the joint itself. This is because the center of rotation might shift slightly and should not be assumed as constant. At each location on the spherical surface the end-effector is rotated to, the elastic energy within the entire joint is measured. The data is stored and processed as a square data matrix, where each data point represents a single location on the spherical surface to which the end-effector is rotated to. When rotation to all locations on the spherical surface is simulated within the range of motion, an energy field can be acquired that visualizes the energy of the entire joint across its range of motion (Fig. 4). This offers valuable insight, as this is the energy field that needs to be made neutrally stable.

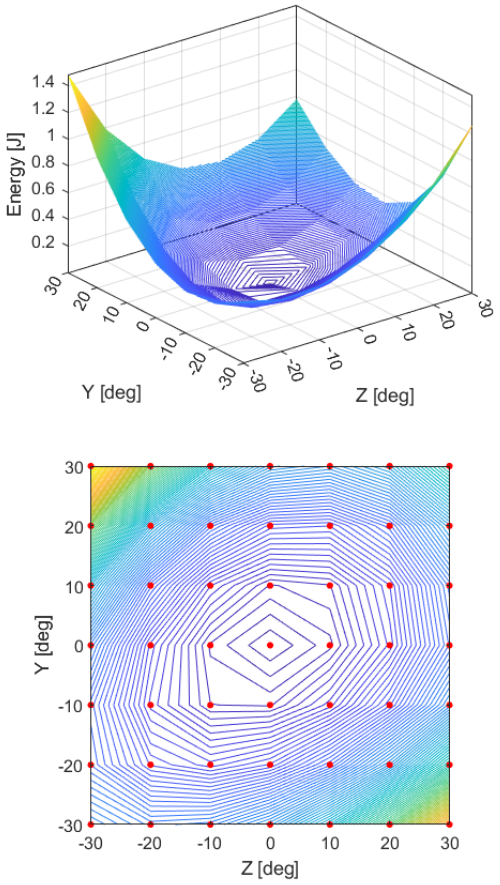


Fig. 4: The energy field of the original Tetra I joint. Top: Isometric perspective. Bottom: 2D perspective including data points (red).

TABLE I: The global and local parameters of the joint and the corresponding initial values.

	Parameter	Description	Initial value
Global	α [deg]	Angle of an arm	75
	β [deg]	Angular height of the tetrahedra	18
	γ [deg]	Angle between the arms	70
	T [#]	Number of tetrahedra per arm	4
	d [m]	Shell thickness	5e-3
	R_{in} [m]	Inner joint radius	6.7e-2
	ΔR [m]	Difference inner and outer joint radius	3.5e-2
Local	α_2 [deg]	Angle of the second arm	75
	β_2 [deg]	Angular height of the tetrahedra on second arm	18
	T_2 [#]	Number of tetrahedra on second arm	4
	d_2 [m]	Shell thickness of second arm	5e-3

According to the design principles used in the Tetra I, spherical motion about a remote center of rotation is maintained as long as the arms are built out of tetrahedra that virtually coincide in the same point. And, to obtain the required degrees of freedom, the two arms must have an angle between them. Using these principles, the parameters that can be used without losing the initial properties are devised. Two sets of parameters are chosen, one 'global' set which affects the entire joint and one 'local' set that affects only the second arm of the joint (Table I). By having this distinction between parameter sets, the parameter effects on the joint that are purely based on the proportion between the two arms can be identified. The local parameters overrule the global parameters if applied to the same section in unison. In that case, the global parameters only affect the first arm, while the local parameters apply to the second arm. A visualization of the global parameters can be seen in Fig. 5.

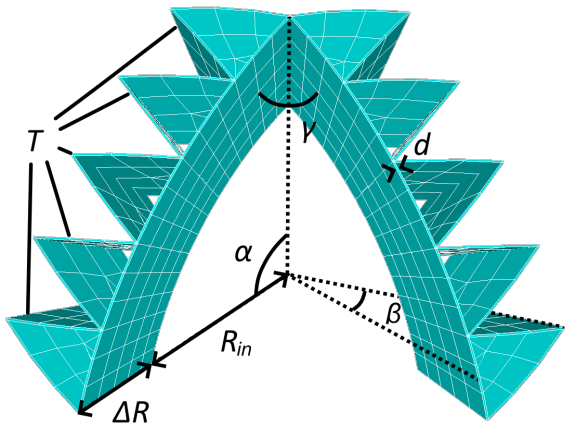


Fig. 5: Visualization of the global parameters.

C. Mathematical derivation

In order to make the joint neutrally stable, its energy field must be constant for the entire range of motion. The first step in achieving this, is to acquire further understanding of the general energy field shape of the joint. The aim of this evaluation is not to calculate an exact value of the joint energy,

but to find its general relation with the rotation angle. For this reason, the joint energy is expressed as a function of the rotation angle, without elaborating extensively on the variables that are independent of the rotation angle. The energy in the joint is a direct result of the work that is put into rotation of the end-effector, therefore the energy in the joint is defined by:

$$E_j = \int M d\theta \tag{1}$$

Here θ is again defined as the rotation angle of the end-effector, starting from the neutral position of the joint in an arbitrary direction of the spherical surface defined by ϕ. To rotate the end-effector by θ, a total moment M is required. The joint is in essence a series connection of curved flexures that are angled relative to each other and are stiff in torsion because of the tetrahedra. The rotation angle θ of the end-effector is facilitated by deformation of the flexures it consists of. For any location in the spherical surface the end-effector is rotated to, it is assumed that bending is the only loading mode that can achieve flexure deformation, because of a significantly lower stiffness relative to other loading modes [6]. According to [11], the nonlinear moment equation for bending of flexures with large deflection is defined as:

$$M = k_j(\phi) sin(\theta), \qquad 0 \leq \theta \leq 90 \quad [deg] \tag{2}$$

The joint stiffness k_j is a function of the Young's modulus, the second moment of area and the total flexure length that is in a bending mode. Since the flexures are curved in this analysis, not the entirety of a flexure has to be in a bending mode, an example of this can be seen in Fig. 6. Therefore, it is assumed that the curved flexures can be divided into an equivalent flexure length that is considered to be in a bending mode, which is valid to be used in k_j of eq. 2, and an equivalent flexure length that is not loaded in bending. The total rotation stiffness of the joint is thus dependent on the equivalent bending flexure length, which can vary with the rotation direction and its corresponding loading modes. Therefore in the general case, k_j varies with ϕ. The nonlinear strain energy in the joint can be found by substituting eq. 2 in eq. 1 and integrating over θ, resulting in:

$$E_j = C - k_j(\phi) cos(\theta), \qquad 0 \leq \theta \leq 90 \quad [deg] \tag{3}$$

The constant C arises a result of integration and can be found with the help of boundary conditions. It is known that there is no deflection and therefore zero stored energy at $\theta = 0$, i.e. $E_j(0) = 0$. This is true when $C = k_j$, hence eq. 3 becomes:

$$E_j = k_j(\phi)(1 - cos(\theta)), \qquad 0 \leq \theta \leq 90 \quad [deg] \tag{4}$$

Thus, the general behaviour of the energy field in a single direction can be described by eq. 4.

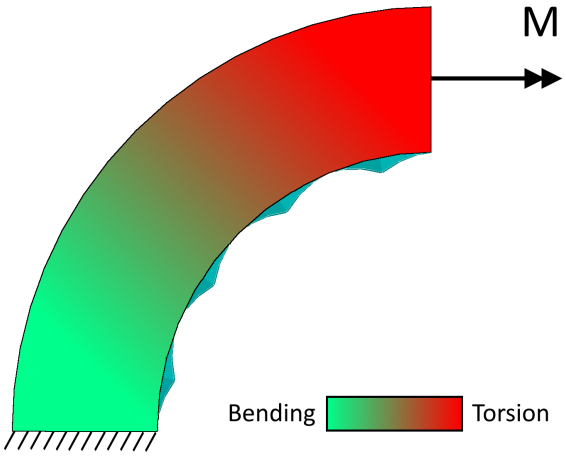


Fig. 6: Example of a gradual transfer from bending to torsion in a curved flexure as a result of an applied moment.

D. Central spring

To make the joint neutrally stable, the energy of the joint that is of the form as described by eq. 4 needs to be compensated by another element, such that their combined energy field is constant over the range of motion. A solution to this is the addition of a 'central spring'. This is a spring that is attached to the end-effector of the joint at one end, while its other end is attached somewhere on the line that goes through the center of rotation and the end-effector in its neutral position (Fig. 7). This spring can be either a tension spring or a compression spring. A tension spring has the advantage of intrinsic stability and ease of attachment. However, the disadvantage is that it coincides with the center of rotation, meaning that a tension spring prevents the joint from having a true remote center of rotation. A compression spring on the other hand, does not coincide with the center of rotation. If a tension spring is used, the end of the spring must be attached on the other side of the center of rotation, furthest from the end-effector. Alternatively, if a compression spring is used, the end of the spring must be attached on the side of the center of rotation that is closest to the end-effector, including locations that are further out than the end-effector itself. In both cases, the spring deflection is maximum when the joint is in its neutral position and decreases when the joint rotates away from its neutral position, resulting in a decreasing energy field of the spring as a function of θ. The parameters used to define the central spring are as follows:

- x_0: The attachment location of the spring along the line between the end-effector in the neutral position and the center of rotation. It is defined as zero in the center of rotation and its positive direction is towards the end-effector.
- L_0: The free length of the spring at which it does not exert a force.
- k_s: The stiffness of the spring.

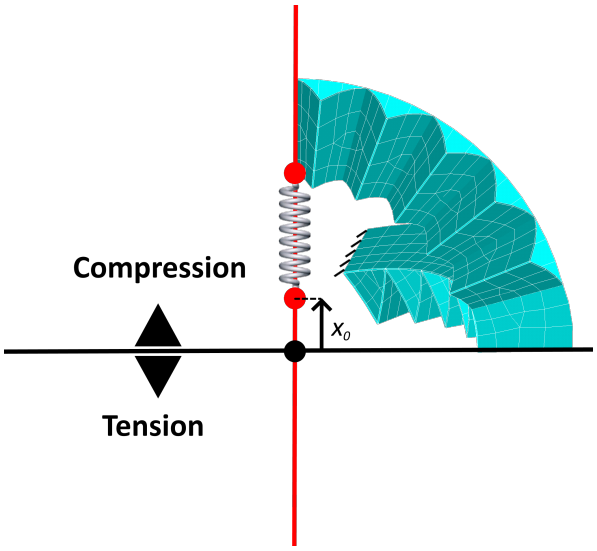


Fig. 7: The central spring solution concept.

To illustrate that the central spring is a suitable candidate to counteract the energy of the joint, a mathematical derivation of the spring energy is provided. The energy in the spring is defined by:

$$E_s = \frac{1}{2}k_s(L - L_0)^2 \tag{5}$$

The total spring length L can be described as a function of the joint rotation angle θ with the help of Pythagoras' theorem:

$$L = \sqrt{(R_{in}sin(\theta))^2 + (R_{in}cos(\theta) - x_0)^2} \tag{6}$$

After processing the quadratic terms, eq. 6 is reduced to:

$$L = \sqrt{R_{in}^2 + x_0^2 - 2R_{in}x_0cos(\theta)} \tag{7}$$

To acquire the spring energy as a function of θ, eq. 7 is substituted in eq. 5, which results in:

$$E_s = \frac{1}{2}k_s(\sqrt{R_{in}^2 + x_0^2 - 2R_{in}x_0cos(\theta)} - L_0)^2 \tag{8}$$

Eq. 8 approximates the negative of the joint energy equation (eq. 4) for certain values of R_{in}, x_0, L_0 and k_s. In the ideal case, L_0 is reduced to zero and becomes a zero length spring. If that were the case, the ideal spring equation would simplify to:

$$E_{s,ideal} = \frac{1}{2}k_s(R_{in}^2 + x_0^2 - 2R_{in}x_0cos(\theta)) \tag{9}$$

However, in practice L_0 will have an effect on the solution and is therefore not neglected in the optimization that is used for the spring design. Still, it will be shown in later chapters that non-zero realistic values of L_0 have an insignificant effect on neutral stability relative to the ideal case (Table IV). The result from eq. 9 is a function of θ with a constant offset that, for the correct parameter values, can equal the negative of eq. 4. Thus, the central spring has the potential to compensate the energy field of the joint in order to make it neutrally stable. However, the energy of the spring is independent of the rotation direction ϕ and thus has an energy field that is symmetrical around a single axis. The joint does not originally have an axi-symmetrical energy field which, if left unadjusted, can not be accurately compensated by the spring for certain rotation directions on the spherical surface. Therefore, the joint is first optimized for an axi-symmetrical energy field.

E. Joint optimization

1) Objective function: The joint is optimized for axi-symmetry in its energy field for a fixed range of motion of 30 degrees in each direction from the neutral position. The objective function to minimize uses a metric that calculates the total normalized RMS error between the current energy field and a completely axi-symmetrical energy field, which is expressed as:

$$f_{symmetry} = \frac{\sum_{p=1}^{m}\sqrt{\frac{\sum_{i=1}^{N}(\bar{E}-E(i))^2}{N}}}{\bar{E}_{total}} \tag{10}$$

First, the average energy value of a subset of N data points that are equidistant from the axis of symmetry is calculated. For this subset, the RMSE of each data point $E(i)$ compared to the average of the subset $\bar{E}$ is calculated. This is done for all data points, resulting in m subsets of equidistant data points from the axis of symmetry. The total RMSE can then be found by the sum of the subsets. Additionally, the RMSE is normalized by the average total energy of the entire energy field $\bar{E}_{total}$. The expected outcome of this optimization is a increasing energy field in all directions as defined by eq. 4. However, note that the optimization does not specifically optimize for that function, only for an axi-symmetrical energy field. This optimization makes use of the simplex search method [12].

2) Sensitivity analysis: The joint optimization is computationally expensive because it needs to complete one entire simulation for every data point. A single energy field requires many data points, dependent on the resolution, and only produces one objective function value. In an effort to reduce the computation time, the number of optimization variables must be reduced to only the relevant ones. A sensitivity analysis is performed to determine the change of the objective function with respect to a change in each individual parameter. It shows that there are many parameters that do not have any effect on the objective function and can thus excluded from the optimization variables. These parameters will maintain their original values as given by Table I in this optimization. The most influential parameters, those that are included in the optimization variables, are α, γ, α_2 and d_2. The effect on the metric of these parameters is shown in Fig. 8. It should be noted that the sensitivity analysis is performed one parameter at a time, while all other parameters maintain their original value. Therefore, the effect of a combination with non-initial value parameters is not considered.

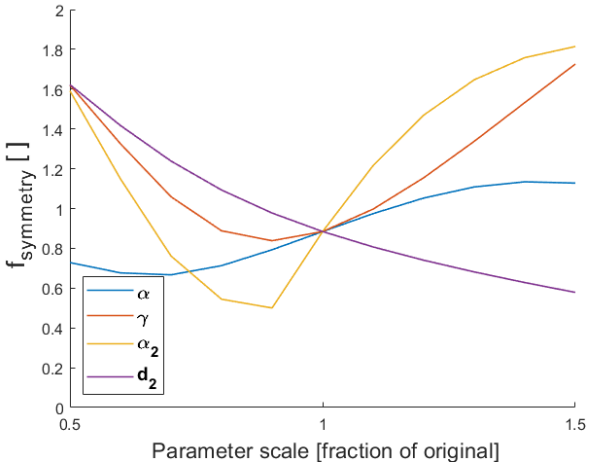


Fig. 8: Sensitivity analysis of the parameters α, α_2, γ and d_2, which shows the effect on $f_{symmetry}$ as a result of parametric variations.

F. Spring optimization

1) Objective function: The spring is optimized to create an energy field such that the combined energy field of both spring and joint, when added together, is as constant as possible in its range of motion. The objective function used for this is given by:

$$f_{balanced} = \sqrt{\frac{\sum_{i=1}^{N} (\bar{E}_{com} - E_{com}(i))^2}{N}} \tag{11}$$

First the average of the current combined energy field $\bar{E}_{com}$ of both spring and joint is calculated. Then the RMSE of each individual data point $E_{com}(i)$ relative to the average is calculated and used as the objective function to minimize. The parameters used in this optimization are x_0, L_0 and k_s. For this optimization, the interior point method is used and repeated with 500 different sets of initial values [13].

2) Constraint equations: This optimization is carried out for a number of relevant scenarios. Initially the theoretical optimum is found with the only constraints being a positive spring stiffness and positive free spring length. Furthermore, two scenarios are optimized where the spring has parameter boundaries and constraint equations to remain within realistically feasible parameter values for a compression and tension spring separately. An overview of all parameter boundaries imposed on the parameters for the different scenarios and their respective constraint equations can be found in Table II. To fully define the constraints, k_s must be subdivided into a number of temporary optimization parameters. These are the spring wire diameter d, spring coil diameter D and the number of coil windings n. Together, these parameters link the spring stiffness k_s to the constraint functions and are related according to [14]:

$$k_s = \frac{d^4 G}{8nD^3} \tag{12}$$

The relation between the shear modulus G and the Young's modulus E is given as $G = \frac{E}{2.6}$ and E has a value of 190 GPa, the Young's modulus of titanium alloy. A constraint function that applies to both a compression and tension spring, is based on the fatigue lifetime. In order to ensure a decent fatigue lifetime for repeated loading, the stress limit can generally be taken as approximately half the ultimate strength [15]. Therefore, this constraint equation is given as:

$$\tau_{max} < 0.5\sigma_{ult} \tag{13}$$

The maximum stress τ_{max} of a helical spring in eq. 13 is given by [15]:

$$\tau_{max} = \frac{8F_{spring}D}{\pi d^3} + \frac{4F_{spring}}{\pi d^2} \tag{14}$$

Similar to before, the fatigue lifetime of the joint itself must also be taken into account. This is again done by setting the stress limit to half the ultimate strength, but for the joint this time. Multiple stress simulations have been performed for the joint with the spring attached to find the maximum spring force for which the joint does not exceed the stress limit in any configuration. This leads to the following constraint equation:

$$F_{spring} < F_{fatigue} \tag{15}$$

For the compression spring, a constraint is needed that prevents the spring from buckling. This is given as [15]:

$$\frac{\Delta L}{L_0} < 0.812(1 - \sqrt{1 - 6.87(\frac{D}{L_0})^2}) \tag{16}$$

When the compression spring coil windings are all pressed together to the point of contact, the spring cannot compress further. Therefore a limit to the minimum compression spring length L_{min} must be set as:

$$nd < L_{min} \tag{17}$$

For a tension spring, a limit must be given to the value of L_0 to prevent it from converging to the unpractical expected value of zero. To do this, the mildly exaggerated assumption is made that no pretension can be given to the spring. Then, when all windings are in contact, the minimum value of L_0 is reached.

$$nd < L_0 \tag{18}$$

TABLE II: The boundaries and constraint equations imposed on the optimization for the different scenarios.

Parameter	Theoretical	Compression	Tension
x_0 [m]	[-∞, ∞]	[0, $3R_{in}$]	[-$2R_{in}$, 0]
L_0 [m]	[0, ∞]	[0, ∞]	[0, ∞]
d [m]	/	[0, ∞]	[0, ∞]
D [m]	/	[0, $0.5R_{in}$]	[0, $0.5R_{in}$]
n [#]	/	[1, ∞]	[1, ∞]
k_s [N/m]	[0, ∞]	/	/
Constraint equations	/	14,15,16,17	14,15,18

G. Experimental validation

1) Prototype: The prototype has minor adaptations relative to the optimized model. The number of tetrahedra has been increased on the first arm, such that the number of tetrahedra per arm length is equal on both arms. To reduce material usage, the prototype has been scaled down to an R_{in} of 5e-2 m and ΔR of 2e-2 m. According to the sensitivity analysis, these changes do not have an effect on the axi-symmetry of the joint energy field. Furthermore, the second arm has been reinforced in bending by placing thin walls on the outer edges of the arm in the spaces between the tetrahedra. This has been done after considering that the thickness of the second arm was optimized to 1.7e-3 m (Table III), while the first arm has a thickness of 5e-4 m (Table I). This large difference in thickness and the fact that bending stiffness scales with thickness to the power of three, leads to the assumption that the second arm can be considered as rigid. An interface has been placed on the base of the joint in order to be able to attach it to the setup. At the end-effector, another interface has been attached that can connect to the spring and to the setup through a hook bolt. The hook bolt reduces the attachment radius of the spring by a non-negligible amount of 2.6e-2 m. Therefore, the spring optimization has been redone for the new configuration, following the identical optimization procedure as described previously. A tension spring is chosen that closely resembles these new optimization parameter values, but is not identical because of spring availability. To mitigate this difference, x_0 is again optimized with the now known tension spring and its fixed values of k_s and L_0. The spring has a k_s of 4.954e2 N/m, L_0 of 2.94e-2 m and x_0 of -8.16e-2 m. The prototype joint with attached spring can be seen in Fig. 9

Fig. 9: The prototype joint with spring attached.

2) Setup: The setup consists of a rigid structure with three translational stages which are perpendicular to each other and two rotational stages which are also perpendicular (Fig. 10). A 6 DoF sensor with a moment resolution of 0.002 Nm is placed in the structure to measure the moments and forces, which are displayed in real time. An attachment slot for the joint is located on the fixed frame of the setup, such that when the joint is attached, the initial center of rotation of the joint coincides with the axes of rotation of the rotational stages. On the other end of the setup, there is an interface that can connect to the end-effector with the help of the hook bolt and screws. These screws have been pretensioned with the intend to allow rotation around the end-effector with minimal friction, while still having negligible play between the parts.

3) Measurement procedure: Before the end-effector is attached to the setup, the setup segment that connects to the end-effector is rotated to its most vertical position. All forces and moments are calibrated to zero at this instance. Next, the end-effector is attached to the setup. Each location on the spherical surface can be reached through a combined rotation of the two rotational stages. Every time the end-effector is rotated to a new location, the rotational stages are fixed in place with the help of an integrated rotation lock. After that, the reaction forces are observed. The reaction forces that are tangential to the rotation of the end-effector are directly correlated to the measured moment, according to $M = FL$. Any deviation from this relation is caused by a center shift of the joint that is being constrained by the setup. Moreover, the force normal to the rotation of the end-effector should be zero when the end-effector is rotating around the center of rotation. Any deviation from a normal force of zero is again caused by a center shift that is being constrained by the setup. Thus, to compensate for this, the three translational stages are manually adjusted until the observed reaction forces indicate no deviation from the center of rotation. When that is done, the measured moment is saved for that location. This procedure is repeated until the same locations on the spherical surface are reached as in the simulations, creating a moment field. An unwanted side-effect is that the moment caused by the weight of the setup segment that connects to the end-effector is also measured. This is accounted for by performing a separate measurement without the joint attached and is therefore purely a measurement of the moment caused by the weight of the setup segment itself. The resulting moment field is subsequently subtracted from the original measurement to obtain the true moment field of the joint. Two different experiments are performed: one without a spring to validate the axi-symmetrical moment field and one with a spring to validate the neutral stability performance.

III. Results

In this section, the results of the joint and spring optimization are shown, followed by the results of the experimental validation.

A. Optimization results

The results of the joint optimization for an axi-symmetrical energy field can be seen in Fig. 11. Its corresponding optimized parameters and the percentual improvement in objective function value can be found in Table III, where $\Delta f_{symmetry}$ is relative to the original joint. To compare the simulation and prototype, of which the latter is directly measured in moments, the neutral stability performance of both is given as the percentual moment reduction. This gives the difference in RMSE between the axi-symmetrical joint and neutrally stable joint as a percentage, relative to a constant zero moment field:

$$M_{red} = 100(1 - \frac{\sum_{i=1}^{N} M_{com}(i)}{\sum_{i=1}^{N} M(i)}) \tag{19}$$

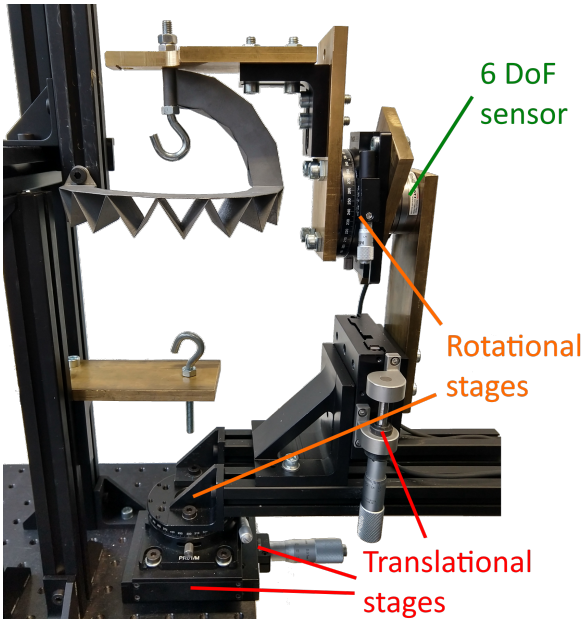


Fig. 10: The experimental setup with the prototype joint.

TABLE III: The optimized parameters of the joint.

Parameter	Value
α [deg]	169.4
γ [deg]	89.04
α_2 [deg]	91.22
d_2 [m]	1.70e-3
$\Delta f_{symmetry}$ [%]	92.51

The optimized spring parameters and its percentual moment reduction can be found in Table IV. The table displays the three different scenarios that have been optimized as described before. From left to right respectively, these are: the theoretical best result, the constrained result for a realistic compression spring and the constrained result for a realistic tension spring. For the theoretical best result, the combined energy field of both spring and joint, which forms the neutrally stable joint, can be found in Fig. 12.

TABLE IV: The optimized parameters of the spring.

Parameter	Theoretical	Compression	Tension
x_0 [m]	-3.738e-1	1.427e-2	-1.307e-1
L_0 [m]	0.000	9.749e-2	1.774e-2
k_s [N/m]	1.059e2	3.443e3	3.333e2
M_{red} [%]	92.92	87.86	92.36

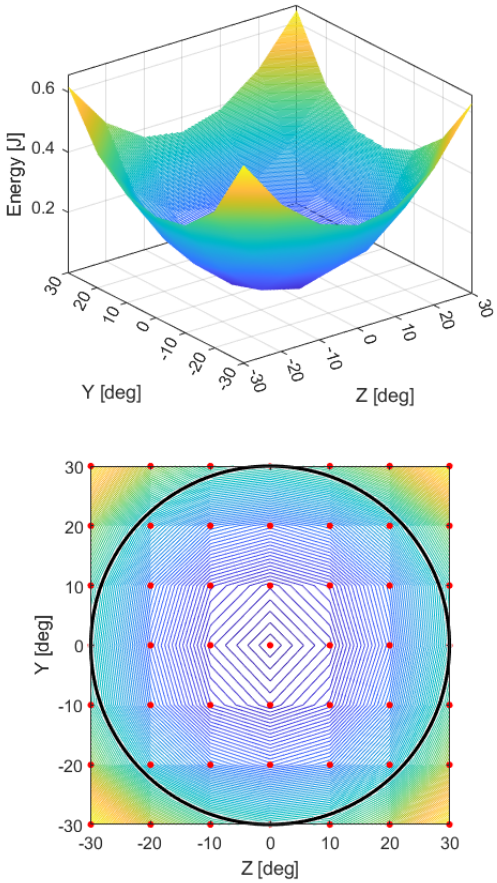


Fig. 11: The optimized axi-symmetrical energy field of the joint. Top: Isometric perspective. Bottom: 2D perspective including data points (red) and range of motion (black) up to which it is optimized for.

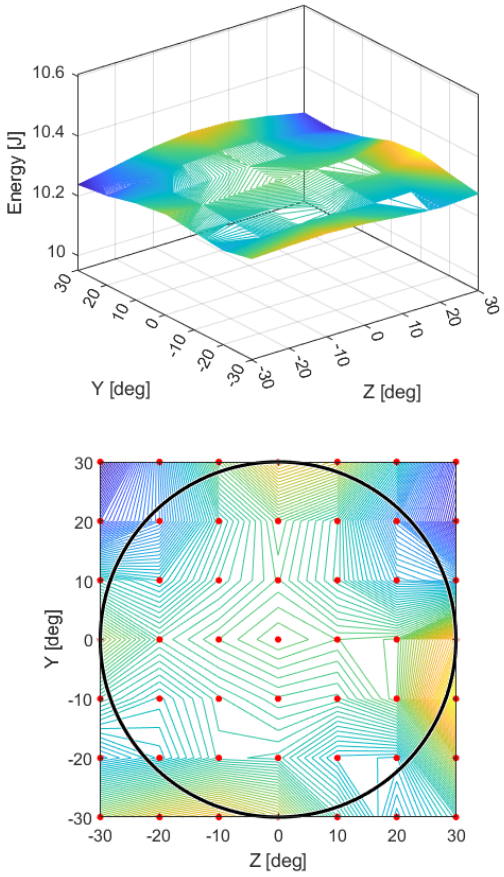


Fig. 12: The energy field of the theoretically best neutrally stable joint. Top: Isometric perspective. Bottom: 2D perspective including data points (red) and range of motion (black) up to which it is optimized for.

B. Experimental validation results

The results of the experimental validation are given in Fig. 13 for both the physical prototype and its simulation. For the prototype measurements, an average uncertainty of ± 0.0191 Nm is estimated. Multiple cross-sections of the moment field are displayed for a value of θ from 0 to 30 degrees. Each cross-section represents a single direction in the moment field given by ϕ. For every moment field, eight directions are viewed, each with a 45 degree difference of ϕ relative to the previous direction. Comparisons can be made in both axi-symmetry and neutral stability. For the latter, a percentual moment reduction of 83.69% was achieved by the prototype, compared to 89.71% in the simulation.

IV. Discussion

The axial symmetry of the optimized energy field in Fig. 11 shows roughly the desired result. As can be seen from Table III, an improvement of 92.51% in the axi-symmetrical objective function is achieved. The resulting parameters of the spring optimization can be seen in Table IV. The theoretical best optimization result achieves a moment reduction of 92.92%. Its negative value of x_0 indiciates a preference for a tension spring. Furthermore, L_0 is optimized to a value of zero, which was predicted from eq. 8. Both the scenarios that are constrained to a realistic compression or tension spring have a non-zero value of L_0, smaller absolute value of x_0 and larger value of k_s relative to the theoretical best result. Even though the parameter differences are significant, the difference in the percentual moment reduction is small. This suggests that there is a widespread range of parameter combinations that produce a local minimum of the objective function, which are near equal in performance. The magnitude of the combined energy field has no inherent limit in the optimization. This can be seen in Fig. 12, where the magnitude of the combined energy field of both spring and joint is significantly larger than the energy field of the joint alone (Fig. 11). Even though this will result in the best neutral stability, an energy field that is too large in magnitude might pose a twofold of problems in reality. First, the forces created by such a large constant

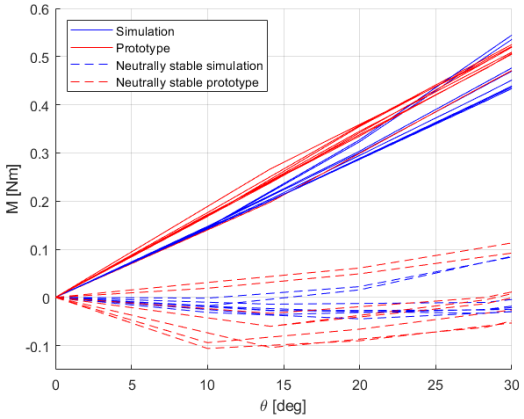


Fig. 13: Comparison between the simulation and prototype. Given as the moment as a function of θ for multiple directions in the moment field defined by ϕ.

energy in the system might decrease the fatigue lifetime of the mechanism. This is taken into account as a constraint function in the realistic compression and tension spring optimization. Secondly, the center of rotation is likely to shift more with increased spring force.

The results of the experimental validation show a clear similarity between the axi-symmetry of the simulation and the prototype, with only minor deviation (Fig. 13). The neutral stability performance shows some variation, while still maintaining general similarity in shape and magnitude. The slight decrease in neutral stability of the prototype has to be a result of the addition of the spring, since the variation in axi-symmetry of the joint moment field is too minor to explain the decrease. It is possible that the center shift is larger with the spring attached than predicted by the simulation, causing disturbances in the resulting moment field.

Finally, some additional effects on the aspects of the final design have been observed. Even though the central spring solution performs as expected, it also has a number of disadvantages. The spring has two attachment points: at the end-effector and the setup frame. These points allow a small amount of rotation that is strictly not compliant. Thus, while the joint and the spring themselves are compliant, the connection between the two is not, making the overall mechanism quasi-compliant. Moreover, the version of the joint that incorporates a tension spring does not have a center of rotation that is truly remote, because the tension spring coincides with the center of rotation. The version of the joint that uses a compression spring does have a true remote center of rotation. A side-effect of the final design is a constraint on the third rotational degree of freedom around the end-effector. This degree of freedom was originally largely provided by bending of the second arm, but since the second arm is now effectively rigid, the third degree of freedom is constrained as well. However, it is not constant, but still has minor rotations as a result of bending deformation in the first arm.

V. Conclusion

This paper presents the design of a quasi-compliant spherical joint which can achieve neutral stability in two rotational degrees of freedom and has a remote center of rotation. This is achieved by creating a neutrally stable version of the existing Tetra I joint [6]. The initial joint was simulated and its energy properties as a result of rotation on the spherical surface were visualized and described mathematically. For best performance, the joint was first optimized for an axi-symmetrical energy field in a range of motion of 30 degrees in all directions from the neutral position. To achieve neutral stability, a spring was added and optimized to minimize the energy deviation of the combined energy field. The optimized parameters were incorporated in a prototype to experimentally validate the theoretical results. For this, a setup was made that measures the moments and forces as a result of the rotation of the joint. Neutral stability, measured as the percentual moment reduction, was improved by 83.69% in the prototype, compared to 89.71% predicted by the simulations.

References

[1] Larry L Howell. “Compliant mechanisms”. In: *21st century kinematics*. Springer, 2013, pp. 189–216.

[2] Juan A Gallego and Just L Herder. “Criteria for the static balancing of compliant mechanisms”. In: *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Vol. 44106. 2010, pp. 465–473.

[3] Mark Schenk and Simon D Guest. “On zero stiffness”. In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 228.10 (2014), pp. 1701–1714.

[4] Farid Parvari Rad et al. “Stiffness Analysis of a Fully Compliant Spherical Chain with Two Degrees of Freedom”. In: *Advances in Robot Kinematics*. Springer, 2014, pp. 273–284.

[5] Farid Parvari Rad et al. “Design and stiffness evaluation of a compliant joint with parallel architecture realizing an approximately spherical motion”. In: *Actuators*. Vol. 7. 2. Multidisciplinary Digital Publishing Institute. 2018, p. 20.

[6] Jelle Rommers, Volkert van der Wijk, and Just L Herder. “A new type of spherical flexure joint based on tetrahedron elements”. In: *Precision Engineering* 71 (2021), pp. 130–140.

[7] Volkan Parlaktaş, Engin Tanık, and Çağıl Merve Tanık. “On the design of a novel fully compliant spherical four-bar mechanism”. In: *Advances in Mechanical Engineering* 11.9 (2019), p. 1687814019879548.

[8] Karthik Chandrasekaran, Adarsh Somayaji, and Asokan Thondiyath. “Realization of a statically balanced compliant planar remote center of motion mechanism for robotic surgery”. In: *Frontiers in Biomedical Devices*. Vol. 40789. American Society of Mechanical Engineers. 2018, V001T07A011.

[9] Shamanth Hampali, Anoosha Pai S, and GK Ananthasuresh. “A Tunable Variable-Torque Compliant Hinge Using Open-Section Shells”. In: *Journal of Mechanisms and Robotics* 12.6 (2020), p. 061010.

[10] Quentin Boehler et al. “Design and evaluation of a novel variable stiffness spherical joint with application to MR-compatible robot design”. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2016, pp. 661–667.

[11] Tat Joo Teo et al. “A generic approximation model for analyzing large nonlinear deflection of beam-based flexure joints”. In: *Precision Engineering* 34.3 (2010), pp. 607–618.

[12] Jeffrey C Lagarias et al. “Convergence properties of the Nelder–Mead simplex method in low dimensions”. In: *SIAM Journal on optimization* 9.1 (1998), pp. 112–147.

[13] Richard H Byrd, Jean Charles Gilbert, and Jorge Nocedal. “A trust region method based on interior point techniques for nonlinear programming”. In: *Mathematical programming* 89.1 (2000), pp. 149–185.

[14] Anton van Beek. *Werktuigbouw.nl formuleboekje*. Mikrocentrum, 2012.

[15] AM Wahl. *Mechanical Springs*. 1964.

3

Discussion

In this chapter, the results from the paper will be discussed in relation to the application of an exoskeleton shoulder joint. Furthermore, a number of research directions are proposed for future work.

3.1. Feasibility as exoskeleton shoulder joint

The joint that has been designed and optimized in the paper shows a moment reduction of 83.69% as a result of static balancing with the spring. This is promising for the application of an exoskeleton shoulder joint, as it has greatly reduced resistance to rotation. While it is a significant improvement in relative terms, there still remains an approximate maximum moment of 0.1 Nm to overcome for the exoskeleton's wearer. An additional source of potential resistance lies in the attachment point of the spring. It is attached to the end-effector and to the setup frame. These points allow a small amount of rotation that is strictly not compliant. Thus, while the joint and the spring themselves are compliant, the connection between the two is not, making the overall mechanism quasi-compliant. A way to decrease the required moment further is by changing the material, and therefore also the stiffness, of the joint. In the entire paper, titanium alloy is used as the material with a relatively high Young's modulus of 190 GPa. Choosing a material with reduced Young's modulus will lower the residual moments. However, this could have adverse effects on the off-axis stiffness of the joint, which should be kept into account.

Another point of interest lies in the degrees of freedom that have been made neutrally stable. The full rotational capability of the human shoulder joint is provided by three degrees of freedom. The external joint on the other hand, has been designed for two rotational degrees of freedom that are neutrally stable. Therefore, the current joint can accommodate only two rotations, whereas the third rotation around the end-effector would have to be facilitated by a conventional rigid body hinge at the moment.

For optimal performance, the center of rotation of the human shoulder joint and the external joint should coincide. The design incorporates a spring that could either be a tension spring or compression spring. In the case of a tension spring, it goes through the center of rotation and would therefore also need to go through the human shoulder. The tension spring variant is thus not feasible for exoskeleton applications. However, the compression spring variant can be used as an exoskeleton shoulder joint, since it does not coincide with the center of rotation. With a compression spring, the joint can fit reasonably compact around the human shoulder due to its semi-circular form. Only a relatively modest space between the skin and external joint would be required to fit the compression spring.

The joint is optimized for a range of motion of 30 degrees in all directions from the neutral position. In other words, each degree of freedom has a total range of motion of 60 degrees. Ideally, the human shoulder joint needs a larger range of motion to perform activities of daily living. A maximum of 167 degrees range of motion is needed to perform all activities for daily living, this is for flexion/extension and less for the other rotations [6]. Although the joint is optimized for a total range of motion of 60 degrees, rotations that are larger are possible, but are less neutrally stable. Furthermore, the stresses in the

joint will eventually reach a point where plastic deformation occurs. To further investigate the effect of a larger range of motion on neutral stability, an additional analysis has been performed (Appendix B.4).

3.2. Future work

There are a number of aspects that could be further explored in future work. Research can be done in making the joint truly compliant. This could be achieved by replacing the spring attachment points with compliant connections for example. Furthermore, a way could be devised to allow the use of tension springs while not compromising the remote center of rotation. This can be done by, for instance, having a virtual resulting force created by multiple spring elements in the place of the single physical spring. Throughout this thesis, gravity has been a negligible aspect of the joint, largely because the residual moment of the stiff and light titanium prototype is more dominant. If a version of a less stiff or more dense material would be made, gravity might become prominent. Therefore, a version could be created that is unaffected by gravity, either in a static or dynamic configuration. For further integration in an exoskeleton, and likely other applications, the third rotational degree of freedom can be made neutrally stable as well.

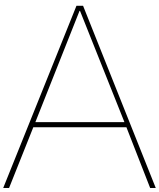
4

Conclusion

The goal of this thesis is to design a compliant spherical joint which can achieve neutral stability in two rotational degrees of freedom and has a remote center of rotation. This has been achieved with a solution based on the addition of two reversed energy fields. When combined, the resulting energy field is near constant and therefore neutrally stable.

Using the foundations of an existing compliant spherical joint with a remote center of rotation, a model has been created with ANSYS. Simulated rotation of the joint's end-effector to any location in its spherical range of motion is made possible by this model to obtain the energy field of the joint. In order to make the joint neutrally stable, two subsequential optimizations are performed. The joint is first optimized for an axi-symmetrical energy field. A spring is then added to the joint and optimized to minimize the energy deviation of the combined energy field.

In order to verify the performance of the final design and its correspondence to the simulations, a prototype is made and experimentally validated. Neutral stability, measured as the percentual moment reduction, was improved by 83.69% in the prototype, compared to 89.71% predicted by the simulations.



Literature report

Before the main thesis project, a literature review was performed in order to obtain more knowledge about existing zero stiffness compliant rotary joints. The findings were classified and compared to create a clear overview of the state of the art. The literature report can be found in this section.

Classification and comparison of zero stiffness compliant rotary joints

Dion Hogervorst

***Abstract*—Many advantages can be attributed to compliant mechanisms for joint applications. A downside however, is that energy is required to rotate because of its elastic deformation. Designing a compliant joint that is neutrally stable removes this disadvantage. Neutral stability is part of the overarching term of zero stiffness, with largely the same working principles. To give insight in the diversity and performance of zero stiffness compliant rotary joints, this paper aims to provide a classification and performance comparison of existing designs. The classification is based on a combination of the working principle of zero stiffness and the type of compliant rotary joint. It shows that a significant part of designs make use of leaf springs and opposed torque curves, where zero stiffness is achieved by counteracting the torque curve of one element with an equal and opposite torque curve of another element. Examples of the categorised mechanisms are given as well. The performance comparison shows the qualitative and quantitative criteria, based on the amount of available data. Large differences between designs can be seen both outside of, and within, the defined categories.**

I. Introduction

Compliant mechanisms have a number of advantages over their traditional rigid-body counterparts. By achieving motion through elastic deformation rather than sliding or rolling contact, friction and the wear that accompanies it, is non-existent [18, 19]. This removes the need for lubrication and reduces backlash as well, resulting in higher precision. Manufacturing costs are low due to a relatively small amount of components and reduced assembly. Furthermore, compliant mechanisms are relatively lightweight and have high potential for applications that require miniaturized mechanisms. All these aspects make compliant mechanisms attractive candidates for use as rotary joints. However, compliant mechanisms have disadvantages as well. For joints especially, one disadvantage is the fact that it requires energy to rotate in the desired direction due to its elastic deformation. This disadvantage can be counteracted by designing the compliant joint to be neutrally stable. If a mechanism is neutrally stable, its total potential energy remains constant for its desired range of motion [43]. This means that the mechanism has no preferred position and that no energy needs to be added to change it. Zero stiffness is often used as a synonym for neutral stability. Even though, strictly speaking, neutral stability implies constant energy and therefore zero torque, while zero stiffness refers to both constant torque and zero torque. In practice, the working principles for both constant and zero torque mechanisms are largely the same, therefore the scope of this paper is not limited to neutral stability, but encompasses zero stiffness as well. So far, work has been done to classify compliant joints in multiple ways [32, 41]. Methods for achieving zero stiffness have been reviewed as well [10, 22, 43]. However, no effort has been made to classify and compare joints that are both compliant and zero stiffness. Such a work could give valuable insight into the diversity and performance of the extraordinary joints that are both compliant and zero stiffness.

This paper aims to provide a classification of zero stiffness compliant rotary joints, based on the working principles for achieving zero stiffness and the type of compliant rotary joint that is used in the design. Furthermore, an effort is made to compare the different designs using performance criteria deemed relevant for zero stiffness joint applications. This paper is structured as follows. Section II explains the method for the literature search, classification and performance criteria. Section III shows the corresponding results in detail, which is discussed in section IV. Finally, section V summarises and concludes this paper.

II. Method

A. Literature search

In order to find relevant literature in a systematic way, a search string that encompasses synonyms of 'zero stiffness', 'compliant' and 'rotational' has been made. In the last column, words that can insinuate a rotational degree of freedom have been included as well. Some terms are allowed to contain one word in between the others and still be relevant, for example: 'zero bending stiffness', this property is shown with 'PRE/1'. Terms with an asterisk contain any words that can be made with the preceding word part. For a higher relevance, a limit of 30 words has been set between the different groups (Fig. 1). Three databases have been used for this search: Scopus, Web of Science and IEEE.

	Within 30 words		
OR	Zero PRE/1 stiffness	Complian*	Rotat*
	Zero PRE/1 torque	Flexible	Revolute
	Zero PRE/1 moment	Flexur*	Torsion*
	Constant PRE/1 torque	Elastic	Joint
	Constant PRE/1 moment		Hinge
	Static* balanc*		Pivot
	Neutral* stab*		Shell

Fig. 1. Search string used to find relevant literature.

For a paper to be included in this review, it needs to suffice in three conditions: compliant motion, zero stiffness and a rotational degree of freedom. A design is considered compliant if the rotation is facilitated by elastic deformation of an element. It is zero stiffness when the torque approaches either a constant or zero for a certain range of motion. Finally, it must have at least one rotational degree of freedom that is linked to compliancy and zero stiffness.

B. Classification

The classification will be done in two separate main categories. First, the *working principle of zero stiffness* and second, the *type of compliant rotary joint*, which are visualized on the top and left of Fig. 5 respectively. Each design is identified by a subcategory of both.

1) ***Working principle of zero stiffness****:* This main category can be split in two branches: *internal stresses* and *opposed torque curves*. The branches are partly inspired by Kok ([22]) and Tarnai ([45]). The branch of internal stresses contains all principles for approximating zero stiffness that rely on a single compliant element. The designs that fall under this category can still contain multiple elements, as long as they are individually designed to achieve zero stiffness. *Controlled deflection* is a subcategory of this branch. This contains single element designs that approach zero stiffness by carefully designed geometry without local continuity, in contrast to the category of *geometric continuity*, which will be introduced later in this chapter. The branch of opposed torque curves captures the principles that rely on multiple compliant elements, where a torque curve and an equal and opposite torque curve cancel each other out, either is produced by a different element or group of elements (Fig. 2). The lowest branches of these categories are the final subcategories where the designs are classified in and are therefore individually defined below.

a) Negative stiffness state: This category is characterized by an increasing torque curve that is created by a positive stiffness element and a decreasing torque curve created by an element in a negative stiffness state. Here, a negative stiffness state is defined as local negative stiffness behaviour as a result of stresses in the element, potentially in combination with a transmission. An example of this is buckling of a flexure. The respective elements do not individually transition between positive and negative stiffness across the zero stiffness range of motion.

b) Reversed stiffness state: When the elements that create the opposed torque curves cannot be distinguished between a continuously positive and continuously negative stiffness element, it falls in this category. The elements are constantly in a state of reversed stiffness relative to each other, be it positive or negative stiffness. Both elements must be prestressed and strain energy is redistributed between the elements during rotation. A transmission is usually in place that creates the desired reversed stiffness states and its non-linear torque

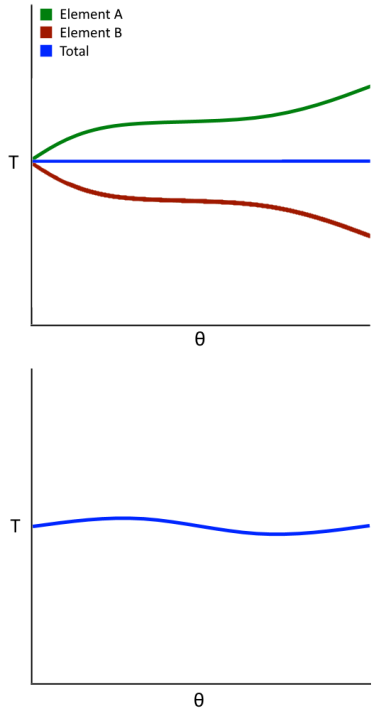


Fig. 2. Examples of torque vs angle for the branches *opposed torque curves* (top) and *internal stresses* (bottom). It can be seen that the branch of opposed torque curves relies on multiple elements, while the branch of internal stresses approximates zero stiffness with a single element.

curve, such that both elements cancel each other out. Often identical elements are used in these joints and high levels of symmetry can be observed.

c) Elastic controlled deflection: A single element that is in a state of elastic deformation. It is designed as such that when it deflects, it does so in a way that approximates zero stiffness behaviour. In reality the stiffness curve will fluctuate around a constant equilibrium. Often an optimization procedure is carried out to minimize this fluctuation.

d) Pseudo-elastic controlled deflection: Pseudo-elasticity is a phenomenon seen in shape memory alloys. With increasing stress, the material does not only elongate like any other, it also undergoes a structural phase transition that is reversible [13]. Because of this, a region of near constant stress and thereby also zero stiffness is created, see Fig. 3.

e) Geometric continuity: In this category, zero stiffness is maintained by ensuring that the designed geometry is locally continuous during deflection. This means that the local deflection does not change in anything but location. Since the reaction torque is produced by the local deflection, and the local deflection remains constant, the reaction torque is constant as well.

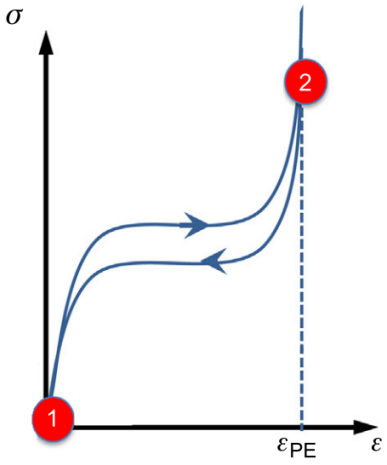


Fig. 3. General pseudo-elastic behaviour. Adapted from [13].

2) ***Type of compliant rotary joint****:* The branches of this main category are based on *lumped compliance* and *distributed compliance*. Lumped compliance refers to localised bending in a concentrated low stiffness region of the joint [1, 11]. Distributed compliance refers to joints that use a wider region of intended bending (Fig. 4). The following subcategories are largely inspired by Machekposhti et al. ([32]) and Sataloff et al. ([42]).

a) Notch joint: Any joint that is intended to bend locally as described earlier in the definition of lumped compliance, falls in this category. This is usually achieved by an area with thinner geometry, creating lower stiffness. Origami based designs are included in this category, since those rely on local crease-lines with lower stiffness to bend.

b) Wire spring: A wire spring is a thin, long and originally straight wire shaped spring that is axisymmetrical. It has low stiffness in all of its rotational degrees of freedom.

c) Leaf spring: Leaf springs are defined as shaped like a rectangle where the thickness is many times smaller than the width, which in turn is many times smaller than the length.

d) Curved spring: When a spring is not straight in its undeformed position, it is defined as a curved spring. A common example of this is a helical spring.

e) Tape spring: Any spring that has a transverse curvature and is straight in longitudinal direction, is considered a tape spring [21]. A well known example of a tape spring is a measuring tape.

f) Contact based: Contact based joints make use of direct contact between parts of the joint that are not physically attached to one another. This imposes boundary conditions on the motion of the joint. Note that while this category still represents compliant motion, the physical contact can result in small amounts of friction.

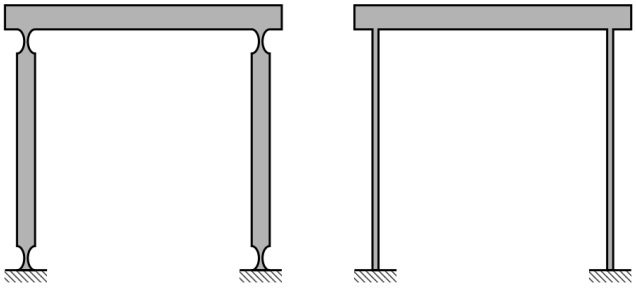


Fig. 4. Lumped compliance on the left and distributed compliance on the right. Retrieved from [1].

C. Performance criteria

Each found design will be compared using performance criteria that are deemed relevant for functioning as a zero stiffness compliant rotary joint, some of these criteria have been identified before by Trease et al. [47]. All performance criteria are described below.

a) Range of motion: The angle of rotation that can be achieved in the zero stiffness region is called the range of motion. This criterion is important when a minimum amount of rotation must be made with zero stiffness behaviour.

b) Zero stiffness error: Represents how close the design is to actual zero stiffness. Defined as the average absolute stiffness value across the range of motion.

c) Scaled zero stiffness error: Each design is tested using a specific material. Since stiffness scales linearly with the Young's Modulus, the zero stiffness error is highly dependent on material choice. Therefore, the scaled zero stiffness error is a representation of the zero stiffness error in a hypothetical situation where each design has equal Young's Modulus. The chosen value for this is 100 GPa. A more objective comparison of the designs can be given with this criterion.

d) Volume: The size of a joint is relevant, as each application requires specific sized joints. Furthermore, the size, just as material, has impact on the stiffness. However, each design has vastly different geometry and working principle and it is therefore not trivial to scale objectively to equal size. For this reason as well, the volume of the joint is useful to provide context for the values of the (scaled) zero stiffness error. This criterion is given as the smallest volume that encompasses the joint.

e) Development stage: Practice and theory can yield vastly different results. Therefore, the development stage of each design is stated. Three levels of development are recognised: analytical, simulation and prototype. Analytical represents a design where its performance is based on mathematical

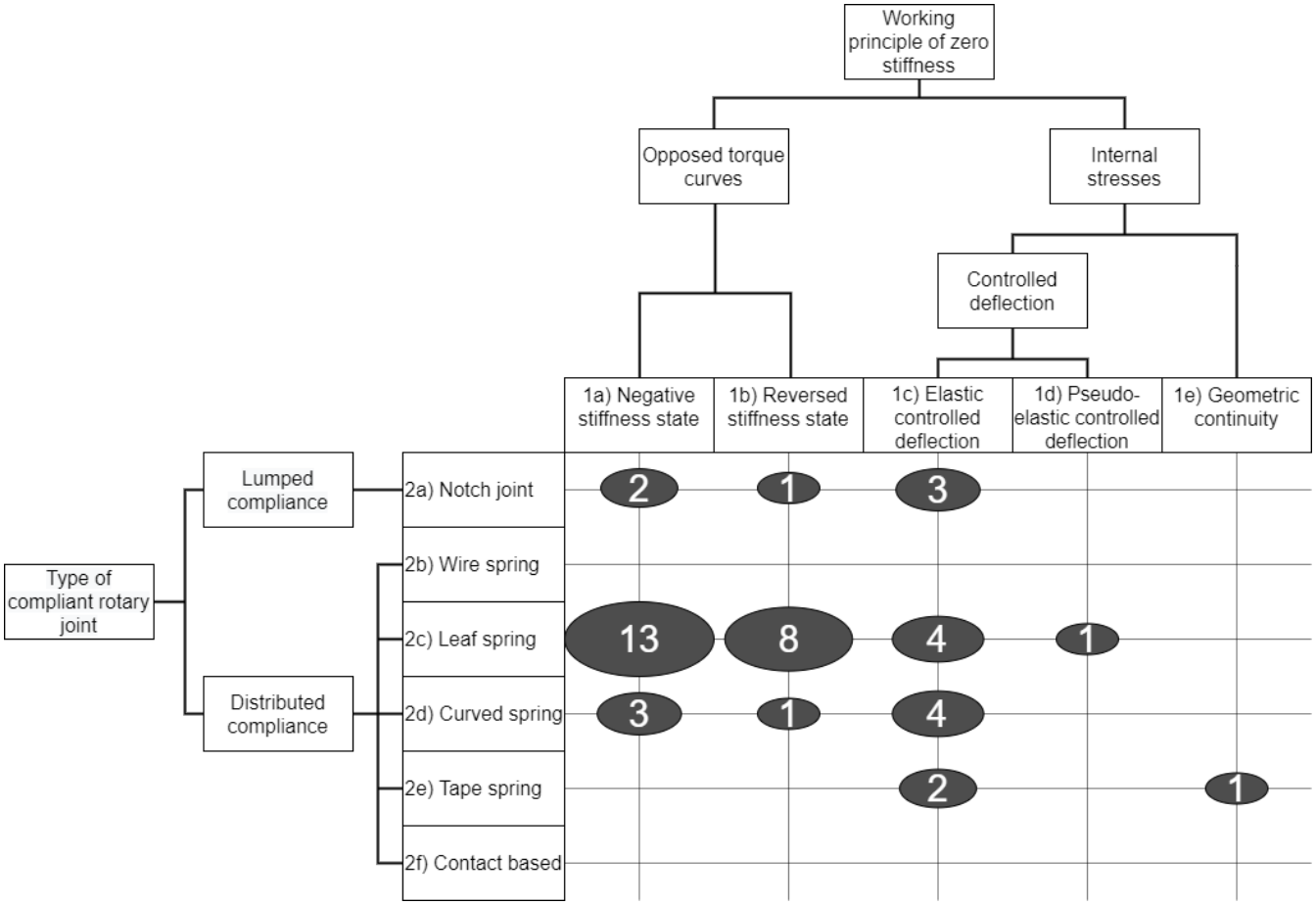


Fig. 5. The derivation of categories for both the type of compliant rotary joint and the working principle of zero stiffness. The amount of designs found in each combination of categories is shown as well.

formulation only. Simulation includes any physics simulation software, usually FEM. Prototype status is given when a physical model is fabricated and tested.

f) Off-axis stiffness: Defined as the stiffness in the other directions than the one that is meant to be zero. For many joint applications it is necessary to maintain a minimum off-axial stiffness for proper functioning. In essence, five degrees of freedom can be identified as off-axis. However, since orientation of the joint is adjustable, this can be reduced to three: rotational, perpendicular translational and parallel translational. Rotational is defined as the weakest of the two off-axis rotational stiffnesses. Perpendicular translational is the weakest of the two translational stiffnesses that act perpendicular to the intended zero stiffness axis of rotation. Parallel translational is the translational stiffness parallel to the intended zero stiffness axis of rotation.

g) Axis drift: When compliant joints deflect, it is sometimes accompanied by a shift in the center of rotation, known as axis drift. It is usually undesired for joint applications. A higher degree of symmetry leads to a decrease in the axis drift severity [15].

III. Results

A. Classification

The amount of designs that were found in each cross-category can be seen Fig. 5. Large differences can be seen in the representation of different cross-categories. Each represented cross-category will be described with an example below, sorted by working principles of zero stiffness. The corresponding papers are given as well in numerical order after the name of the cross-category.

a) Negative stiffness state & Notch joint [24, 39]: An example design consists of a positive stiffness notch joint that is extrapolated in a helical pattern and a hyperelastic tensioned element to provide the counteracting negative stiffness (Fig. 6) [39]. During rotation, the force created by the negative stiffness element changes in magnitude, direction and moment arm. All these aspects affect the produced torque. The total torque of the system is balanced at zero, creating a neutrally stable mechanism.

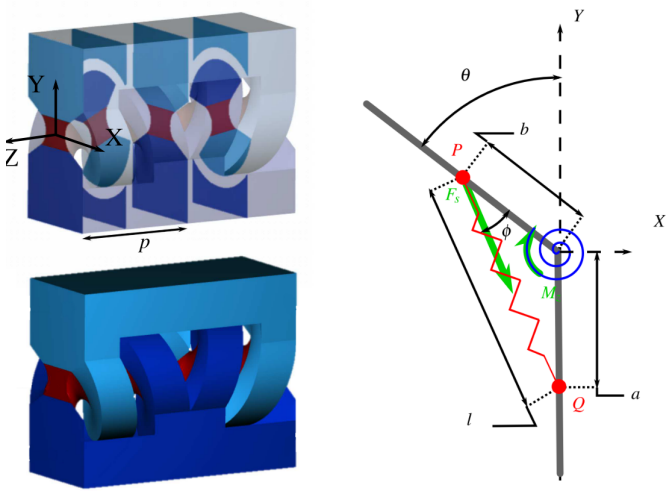


Fig. 6. *Negative stiffness state & notch joint* example. Left shows the central notch joint extrapolated in a helical pattern. Right schematically shows the entire joint, including negative stiffness element. Adapted from [39].

b) Negative stiffness state & Leaf spring [2, 3, 4, 6, 7, 23, 24, 25, 29, 31, 33, 34, 48, 50]: A clear example of this common cross-category is designed by Bilancia et al. [4]. It consists of an inner ring and outer ring that are connected by two general positive stiffness elements and by two negative stiffness elements in the form of pre-buckled leaf springs (Fig. 7). As the inner ring rotates relative to the outer ring, the positive and negative torque effectively cancel each other, resulting in zero stiffness behaviour.

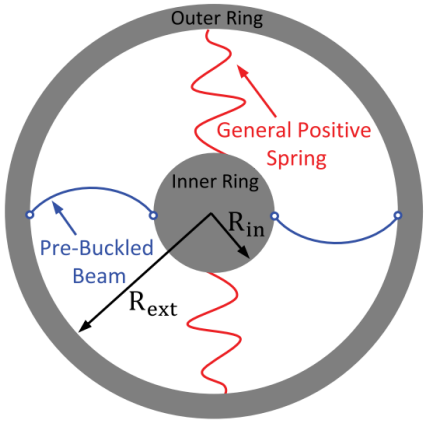


Fig. 7. *Negative stiffness state & leaf spring* example. Positive and negative stiffnesses cancel each other. The negative stiffness elements are given in blue and the positive stiffness elements in red. Retrieved from [4].

c) Negative stiffness state & Curved spring [4, 29, 33]: An example design is made using crossed leaf springs as the positive stiffness element and a curved spring in the form of a helical spring as the negative stiffness element, see Fig. 8 [33]. The helical spring is in a state of tension and counteracts the torque created by the leaf springs during rotation. It should be noted that the force created by the helical spring has changing magnitude, direction and moment arm during rotation, all of which affect the torque it provides.

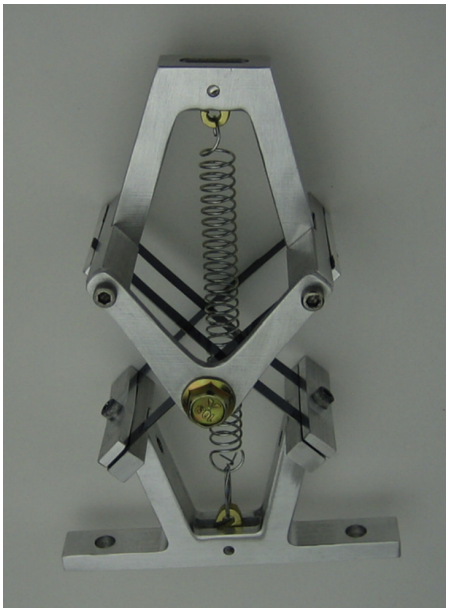


Fig. 8. *Negative stiffness state & curved spring* example. The helical spring is in a state of tension and counteracts the torque created by the leaf springs during rotation. Retrieved from [33].

d) Reversed stiffness state & Notch joint [30]: A notch joint design in the form of origami in a kresling pattern [30]. Two kresling modules are placed on top of each other in a mirrored fashion (Fig. 9). The height and rotation are coupled for both kresling modules. As one decreases its rotation angle, the other increases instead. Strain energy is redistributed between the kresling modules during rotation, resulting in a region of quasi zero stiffness.

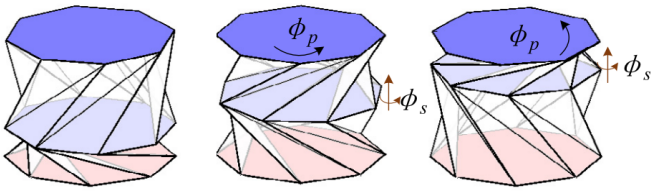


Fig. 9. *Reversed stiffness state & notch joint* example. Based on stacked kresling modules that rotate in reversed direction. Retrieved from [30].

e) Reversed stiffness state & Leaf spring [5, 9, 28, 35, 36, 46, 52]: In this design example, two sets of leaf springs have been applied to statically balance a cross-axis flexural pivot (Fig. 10) [36]. As the mechanism rotates, one set of leaf springs deflects further and the other set deflects less, redistributing the strain energy within the system, including the cross-axis flexural pivot. The two sets of leaf springs represent

a close approximation of zero length springs. Together with the cross-axis pivot, they create a resultant force vector that goes through the center of rotation, creating a zero torque motion. Although the cross-axis flexural pivot does create a torque on its own, the working principle here is mainly based on reversed stiffness states between sets of leaf springs. This is elegantly shown by Herder, where a similar design works without a central torque-producing element as well [16].

Fig. 10. *Reversed stiffness state & leaf spring* example. Two sets of leaf spring deflect in reversed direction while rotating. Together with the central pivot, this creates a statically balanced joint. Retrieved from [36].

f) Reversed stiffness state & curved spring [20]: An example design of this group is made out of a single curved spring shaped in a way where it acts like two separate torque producing elements [20]. A number of constraints are in place that are enforced by rigid-body contact in reality. Initially, the design is put in a stressed position, as can be seen in Fig. 11. During rotation, energy is transferred from one torsion bar to the other. This results in a near zero stiffness range of motion.

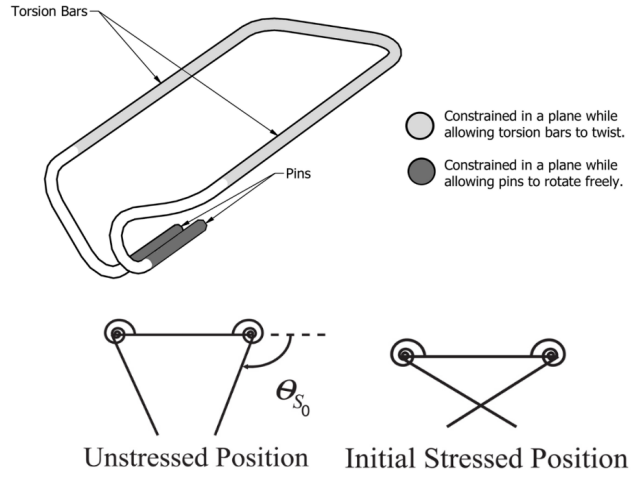


Fig. 11. *Reversed stiffness state & Curved spring* example. The curved spring is shaped to behave as two separate torque producing bars. During rotation, energy is transferred from one bar to the other, keeping the overall energy constant. Retrieved from [20].

g) Elastic controlled deflection & Notch joint [40]: An adaptable design based on origami is an example of this group. It consists of a folded sheet with a number of crease lines with localised low stiffness, see Fig. 12 [40]. During deflection, the internal reaction torque approaches a constant. This is created by the combined reaction torque of the individual crease lines. Three different versions of this design are given by Rommers et al., all are zero stiffness. One is a constant moment joint, the other a gravity compensating joint and the last a zero torque joint.

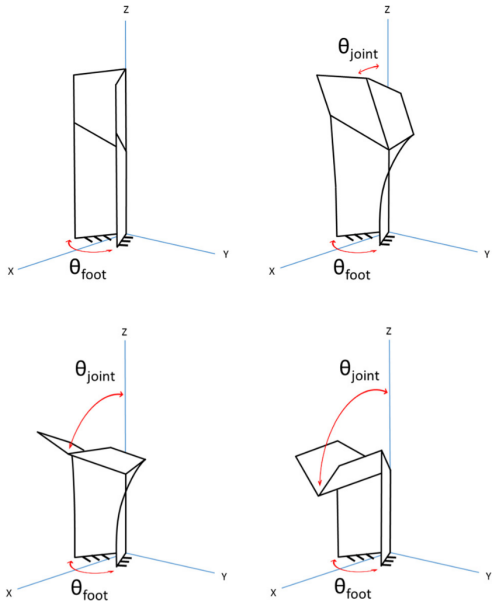


Fig. 12. *Elastic controlled deflection & notch joint* example. While rotating, the combined reaction torques of the crease lines approaches a constant. Retrieved from [40].

h) Elastic controlled deflection & Leaf spring [12, 26, 49]: Consisting of a inner and outer ring connected by leaf springs, this design is an example of multiple elements where each element has individual zero stiffness (Fig. 13) [49]. When a leaf spring buckles, a local negative stiffness region is created. In this case, the leaf springs are optimized such that when it buckles it closely resembles zero stiffness behaviour in a finite range of motion. The leaf springs are not buckled in the initial configuration and thus require an applied pre-stress before it can function as a zero stiffness joint.

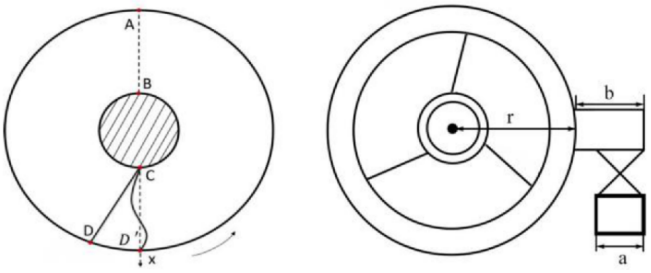


Fig. 13. *Elastic controlled deflection & leaf spring* example. Each individual leaf spring buckles such that it approaches zero stiffness within the range of motion. Retrieved from [49].

i) Elastic controlled deflection & Curved spring [17, 27, 37, 38]: Similar to the previous example, this design consists of an inner and outer ring connected by three curved springs [37]. Each individual curved spring has been optimized to be as near to constant torque as possible during deflection. The outcome can be seen in Fig. 14.

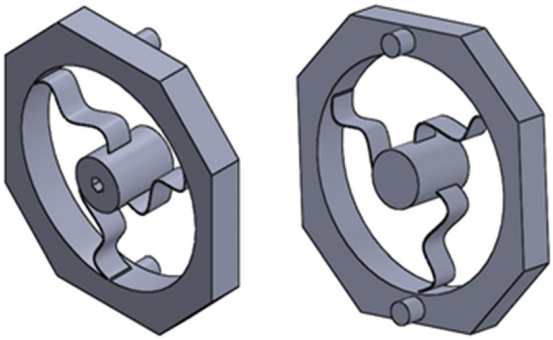


Fig. 14. *Elastic controlled deflection & curved spring* example. Curved springs are used that deflect in a way that approaches zero stiffness. Retrieved from [37].

j) Elastic controlled deflection & Tape spring [8, 14]: This design uses two tape springs to create a variety of stiffness behaviours, among them is zero stiffness (Fig. 15) [14]. The way the deflection is controlled is rather unique: a guideway profile is carefully designed that leads the deflection of the tape springs along a desired path. The design of the guideway determines the stiffness behaviour. Because two tape springs are used, there is a remote center of rotation.

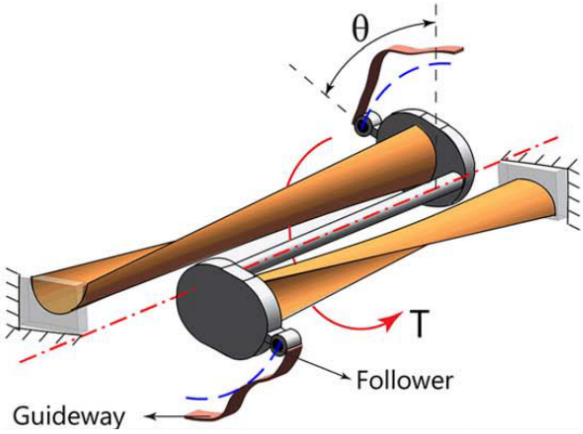


Fig. 15. *Elastic controlled deflection & tape spring* example. Two tape springs follow a guideway to create specific stiffness behaviour, including zero stiffness. Retrieved from [14].

k) Pseudo-elastic controlled deflection & Leaf spring [44]: The only design found in this category uses the previously explained pseudo-elastic behaviour to approximate zero stiffness (Fig. 3). A pseudo-elastic leaf spring is used with braced and unbraced parts, to better control the deformations and focus bending in the unbraced areas (Fig. 16) [44].

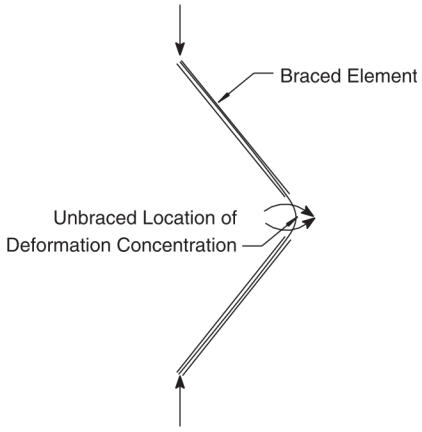


Fig. 16. *Pseudo-elastic controlled deflection & leaf spring* example. Pseudo-elastic material is used to create zero stiffness. Bending is focused in the unbraced parts of the leaf spring. Retrieved from [44].

l) Geometric continuity & Tape spring [51]: This example consists of a classic tape spring, similar to a common measuring tape. When in unbend configuration, it has non-zero stiffness [51]. Resistance is met during initial bending, until a certain point is reached where it suddenly snaps through, creating a flat localised fold (Fig. 17). This localised fold contains virtually all strain energy in the tape spring. The edges of this fold expand outward during bending and do not change in geometry, resulting in a constant reaction torque.

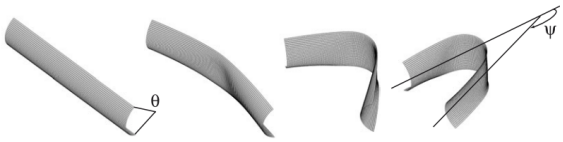


Fig. 17. *Geometric continuity & tape spring* example. After a certain amount of bending, a fold is created with continuous geometry that has zero stiffness. Retrieved from [51].

B. Performance comparison

Not all of the performance criteria have been measured or even mentioned in all of the papers. Because of this, a division has been made between quantitative performance criteria and qualitative performance criteria. The qualitative performance criteria are either not given numerically or are barely mentioned in all papers. Corresponding criteria are 'development stage', 'off-axis stiffness' and 'axis drift'. Off-axis stiffness is separated into three stiffnesses as discussed before: rotational stiffness (K_R), perpendicular translational stiffness ($K_{T\perp}$) and parallel translational stiffness ($K_{T\parallel}$). The qualitative performance is given as '+ +', '+', '-' and '- -', from best to worst respectively and is based on an evaluation of the data that is still available in the papers. The development stage is given here as well in the three previously described levels: analytical (A), simulation (S) and prototype (P), see Fig. 18. The quantitative performance criteria contain factual

		NN39	NL2	NL3	NL6	NL7	NL25	NL31	NL34	NL48	NL50	NLC4	NLC29	NLC33	NLN24	RN30	RL5	RL9	RL23	RL35	RL36	RL46a	RL46b	RL52	RC20	EN40a	EN40b	EN40c	EL12	EL26	EL28	EL49	EC17	EC27	EC37	EC38	ET8	ET14	PL44	GT51
Development stage		P	P	P	P	P	P	P	P	P	P	P	A	P	P	A	A	A	P	P	P	S	S	P	P	P	A	A	P	A	P	P	P	P	P	P	P	S	A	S
Off-axis stiffness	K_R	+ +	-	+ +	+	- -	+	+ +	+		-	-	- -	+	-	+ +		-	+	+ +	+	- -	-	-	+	-	-	-	-	+	-	+	-	-	-	- -	+	+		-
	$K_{T\perp}$	+	- -	+ +	+ +	- -	- -	+ +	+		+	-	- -	+	-	+ +		-	+	- -	+	- -	-	+	+	+ +	+ +	+ +	- -	+ +	-	+ +	-	+	+	- -	-	+ +		- -
	$K_{T//}$	+	+	+ +	+	- -	+	+ +	+ +		+ +	+	- -	+ +	+	- -		-	+	+ +	+ +	+	+	+ +	+	+ +	+ +	+ +	+	- -	-	+ +	+	+	+	- -	-	+ +		+
Axis drift		+	-	+ +	+	+ +	+ +	+ +	-		+	+ +	+ +	-	+	+ +		+ +	+ +	-	-	+ +	+ +	-	-	+	+	+	+ +	+ +	+	+ +	+	+	+	+	+	+		- -

Fig. 18. Comparison of the designs according to the qualitative performance criteria.

Fig. 19. Comparison of the designs according to the quantitative performance criteria, using a logarithmic scale.

values that are derived from the papers. Some values are given directly and are confidently correct, these are given in green. Other values are found indirectly, for example by differentiating data from a figure, these values are given in orange (Fig. 19). To distinguish between a value of zero and no data, the name of the design is removed in the criteria where no data is present. The individual designs are named as follows: the first letter represents the category for the working principle of zero stiffness, the second letter represents the type of compliant rotary joint and the number after that equals the corresponding paper reference. The letters of the categories for working principle of zero stiffness are: *negative stiffness state* (N), *reversed stiffness state* (R), *elastic controlled deflection* (E), *pseudo-elastic controlled deflection* (P) and *geometric continuity* (G). For the type of compliant rotary joint, the categories and letters are: *notch joint* (N), *leaf spring* (L), *curved spring* (C) and *tape spring* (T). Rarely, a design falls into multiple categories of compliant rotary joints. When that is the case, multiple letters are given for the type of compliant rotary joint. Some papers contain multiple designs, these are labelled with an extra letter at the end to distinguish between them. For clarity, those designs are briefly described here. Thanaki et al. created two designs, R_L_46a is the design made out of 2 beams, while R_L_46b is the one made out of 3 beams [46]. Rommers et al. created three designs, E_N_40a is the constant moment joint, E_N_40b is the gravity compensating joint and E_N_40c is the zero moment joint [40].

IV. Discussion

From the categorisation in Fig. 5, it can be concluded that the vast majority of designs is based on leaf springs and the branch of opposed torque curves. A possible explanation for this occurrence, is that leaf springs are commonly used in general purpose compliant mechanisms and are one of the less complex building blocks, making it an attractive option. Moreover, the branch of opposed torque curves is arguably more intuitive and known to be used as a working principle of zero stiffness. The wire spring is not used once, this can be explained by the fact that wire springs are the three-dimensional equivalent of leaf springs in terms of degrees of freedom and most designs are based on a two-dimensional configuration. The off-plane stiffness that a leaf spring provides would then make leaf springs the better option. The underrepresentation of tape springs in the opposed torque curves branch is to be expected, since tape springs have an inherent tendency to have zero stiffness properties as a single element and thus do not necessarily require multiple elements [21]. The possibility of friction in contact based designs could explain the lack of designs in that category. This is because friction is a non-conservative force and converts kinetic energy to thermal energy. This excludes a constant energy level in a passive mechanism, making neutral stability impossible. Still, contact based designs can be made frictionless if there are no sliding or rollings contacts and should therefore not be prematurely discarded as a viable option. Indeed, if properly designed, contact based joints have the potential to provide an elegant solution for achieving zero stiffness in designs that would otherwise require more complex compliant elements. The working principles of pseudo-elasticity and geometric continuity are both only represented by a single design without any apparent explanation and therefore have potential for novel designs in future research as well.

The qualitative and quantitative performance criteria (Fig. 18 and 19) both show that within each category for working principle of zero stiffness and type of compliant rotary joint, there are vast differences in performance. This was to be expected, as the same working principle can yield a myriad of design solutions. Still, some trends can be found in the data. The N_L group for example, has relatively high scaled and unscaled zero stiffness error on average, as well as large volume. It should be noted that this group has an development stage consisting almost exclusively of prototypes and is therefore more likely to have worse performance compared to more theoretical stages.

Another interesting correlation must be highlighted that is not readily visible in the data. The range of motion of a design is largely based on what the authors themselves consider the zero stiffness range. However, the joint is usually able to rotate further than that, at the cost of a larger zero stiffness error. Thus, the range of motion and the (scaled) zero stiffness error are inversely related.

V. Conclusion

The goals of this paper are to provide a classification of existing zero stiffness compliant rotary joints and to give a comparison between the designs using performance criteria. A classification has been made based on the working principle of zero stiffness and the type of compliant rotary joint. Both categories have been branched out in their underlying subcategories and each design was attributed to one subcategory of both branches. Example designs are given of the represented cross-categories. Furthermore, research gaps are identified in the combinations of categories that are not represented without logical reason. Qualitative and quantitative performance criteria are used to compare the designs, based on the data that is available in the found literature. Together, these criteria give an impression of the designs and how well they perform as a zero stiffness compliant rotary joint.

Future research can be done in designing a joint that uses pseudo-elasticity or geometrical continuity as working principles to achieve zero stiffness. These working principles are not well explored for this application and therefore have potential for novel designs in future research. Furthermore, contact based designs can be further explored, since no zero stiffness design has been made using this type of compliant joint.

References

[1] Juan Andres Gallego Sanchez. *Statically balanced compliant mechanisms: theory and synthesis*. November. 2014. ISBN: 9789461862150. DOI: 10.13140/2.1.3093.2163.

[2] Luc Berntsen, Daan H Gosenshuis, and Just L Herder. "DESIGN OF A COMPLIANT MONOLITHIC INTERNALLY STATICALLY BALANCED FOUR-BAR MECHANISM". In: (2016), pp. 1–13.

[3] Shusheng Bi et al. "Study of Zero Stiffness Flexural Pivot Based on Spring-crank Mechanism". In: *Jixie Gongcheng Xuebao/Journal of Mechanical Engineering* 54.13 (2018), pp. 86–93. ISSN: 05776686. DOI: 10.3901/JME.2018.13.086.

[4] Pietro Bilancia et al. "Zero Torque Compliant Mechanisms Employing Pre-buckled Beams". In: *Journal of Mechanical Design* 142.11 (2020), pp. 1–12. ISSN: 1050-0472. DOI: 10.1115/1.4046810.

[5] Guimin Chen. "Multistability of Compliant Sarrus Mechanisms". In: (2016), pp. 1–11.

[6] Stephen Daynes, Xavier Lachenal, and Paul M. Weaver. "Concept for morphing airfoil with zero torsional stiffness". In: *Thin-Walled Structures* 94 (2015), pp. 129–134. ISSN: 02638231. DOI: 10.1016/j.tws.2015.04.017. URL: http://dx.doi.org/10.1016/j.tws.2015.04.017.

[7] A. G. Dunning, N. Tolou, and J. L. Herder. "A compact low-stiffness six degrees of freedom compliant precision stage". In: *Precision Engineering* 37.2 (2013), pp. 380–388. ISSN: 01416359. DOI: 10.1016/j.precisioneng.2012.10.007. URL: http://dx.doi.org/10.1016/j.precisioneng.2012.10.007.

[8] Alexander E Ehret et al. "A ZERO-STIFFNESS ELASTIC SHELL STRUCTURE". In: 6.7 (2011).

[9] Thomas A Evans et al. "Multistable Behavior of Compliant Kaleidocycles". In: (2016), pp. 1–7.

[10] Juan A. Gallego and Just Herder. "Criteria for the static balancing of compliant mechanisms". In: *Proceedings of the ASME Design Engineering Technical Conference* 2.PARTS A AND B (2010), pp. 465–473. DOI: 10.1115/DETC2010-28469.

[11] Juan A. Gallego and Just Herder. "Synthesis methods in compliant mechanisms: An overview". In: *Proceedings of the ASME Design Engineering Technical Conference* 7.PARTS A AND B (2009), pp. 193–214. DOI: 10.1115/DETC2009-86845.

[12] Ishit Gandhi and Hong Zhou. "Synthesizing constant torque compliant mechanisms using precompressed beams". In: *Journal of Mechanical Design, Transactions of the ASME* 141.1 (2019), pp. 1–7. ISSN: 10500472. DOI: 10.1115/1.4041330.

[13] Martin Gurka. *Active hybrid structures made of shape memory alloys and fiber-reinforced composites*. iii. Elsevier Inc., 2015, pp. 727–751. ISBN: 9780323265034. DOI: 10.1016/B978-0-323-26434-1.00024-6. URL: http://dx.doi.org/10.1016/B978-0-323-26434-1.00024-6.

[14] Shamanth Hampali, S. Anoosha Pai, and G. K. Ananthasuresh. "A Tunable Variable-Torque Compliant Hinge Using Open-Section Shells". In: *Journal of Mechanisms and Robotics* 12.6 (2020), pp. 1–10. ISSN: 19424310. DOI: 10.1115/1.4047440.

[15] Xiao Bing He et al. "Effect of degree-of-symmetry on kinetostatic characteristics of flexure mechanisms: A comparative case study". In: *Chinese Journal of Mechanical Engineering (English Edition)* 31.2 (2018). ISSN: 21928258. DOI: 10.1186/s10033-018-0235-4. URL: https://doi.org/10.1186/s10033-018-0235-4.

[16] J.L. Herder. *Energy-free systems: theory, conception, and design of statically balanced spring mechanisms*. November. 2001, p. 268. ISBN: 9789037001921. DOI: 10.13140/RG.2.1.3942.8966. URL: http://repository.tudelft.nl/view/ir/uuid:8c4240fb-0315-462a-8b3b-efbd0f0e68b6/.

[17] Chia Wen Hou and Chao Chieh Lan. "Functional joint mechanisms with constant-torque outputs". In: *Mechanism and Machine Theory* 62 (2013), pp. 166–181. ISSN: 0094114X. DOI: 10.1016/j.mechmachtheory.2012.12.002. URL: http://dx.doi.org/10.1016/j.mechmachtheory.2012.12.002.

[18] Larry L Howell. "Compliant mechanisms". In: *21st century kinematics*. Springer, 2013, pp. 189–216.

[19] Larry L. Howell, Spencer P. Magleby, and Brian M. Olsen. *Handbook of Compliant Mechanisms*. 2013. 2019. ISBN: 9781119953456.

[20] Brian D Jensen and Cesare H Jenkins. "DESIGN OF SMALL-SCALE STATICALLY BALANCED COMPLIANT JOINTS". In: (2016).

[21] Marinus G de Jong, Werner W P J van de Sande, and Just L Herder. "Properties of Twofold Tape Loops: The Influence of the Subtended Angle". In: 11.2 (2019), p. 20912.

[22] Sjaak Kok. *Literature review of occurrences and working principles of elastic neutral stability*. Tech. rep. 2020.

[23] P. R. Kuppens et al. "Compliant Mechanisms That Use Static Balancing to Achieve Dramatically Different States of Stiffness". In: *Journal of Mechanisms and Robotics* 13.2 (2021), pp. 1–6. ISSN: 19424310. DOI: 10.1115/1.4049438.

[24] P. R. Kuppens et al. "Monolithic binary stiffness building blocks for mechanical digital machines". In: *Extreme Mechanics Letters* 42 (2021), p. 101120. ISSN: 23524316. DOI: 10.1016/j.eml.2020.101120. URL: https://doi.org/10.1016/j.eml.2020.101120.

[25] X. Lachenal, S. Daynes, and P. M. Weaver. "A zero torsional stiffness twist morphing blade as a wind turbine load alleviation device". In: *Smart Materials and Structures* 22.6 (2013). ISSN: 09641726. DOI: 10.1088/0964-1726/22/6/065016.

[26] X. Lachenal, P. M. Weaver, and S. Daynes. "Multi-stable composite twisting structure for morphing applications". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 468.2141 (2012), pp. 1230–1251. ISSN: 14712946. DOI: 10.1098/rspa.2011.0631.

[27] Chao Chieh Lan and Jung Yuan Wang. "Design of adjustable constant-force forceps for robot-assisted surgical manipulation". In: *Proceedings - IEEE International Conference on Robotics and Automation* (2011), pp. 386–391. ISSN: 10504729. DOI: 10.1109/ICRA.2011.5979556.

[28] Baokun Li and Guangbo Hao. "On Generating Expected Kinetostatic Nonlinear Stiffness Characteristics by the Kinematic Limb-Singularity of a Crank-Slider Linkage with Springs". In: *Chinese Journal of Mechanical Engineering (English Edition)* 32.1 (2019). ISSN: 21928258. DOI: 10.1186/s10033-019-0369-z. URL: https://doi.org/10.1186/s10033-019-0369-z.

[29] Jinyou Li et al. "Torsional negative stiffness mechanism by thin strips". In: *Theoretical and Applied Mechanics Letters* 9.3 (2019), pp. 206–211. ISSN: 20950349. DOI: 10.1016/j.taml.2019.01.012. URL: http://dx.doi.org/10.1016/j.taml.2019.01.012.

[30] Zhen Li et al. "Uncovering rotational multifunctionalities of coupled Kresling modular structures". In: *Extreme Mechanics Letters* 39 (2020), p. 100795. ISSN: 23524316. DOI: 10.1016/j.eml.2020.100795. URL: https://doi.org/10.1016/j.eml.2020.100795.

[31] Tian Liu et al. "Research on zero-stiffness flexure hinge (ZSFH) based on spring four-bar linkage(4BSL)". In: *Mechanism and Machine Theory* 143 (2020), p. 103633. ISSN: 0094114X. DOI: 10.1016/j.mechmachtheory.2019.103633. URL: https://doi.org/10.1016/j.mechmachtheory.2019.103633.

[32] D. Farhadi Machekposhti, N. Tolou, and J. L. Herder. "A review on compliant joints and rigid-body constant velocity universal joints toward the design of compliant homokinetic couplings". In: *Journal of Mechanical Design, Transactions of the ASME* 137.3 (2015). ISSN: 10500472. DOI: 10.1115/1.4029318.

[33] Ezekiel G. Merriam and Larry L. Howell. "Non-dimensional approach for static balancing of rotational flexures". In: *Mechanism and Machine Theory* 84 (2015), pp. 90–98. ISSN: 0094114X. DOI: 10.1016/j.mechmachtheory.2014.10.006. URL: http://dx.doi.org/10.1016/j.mechmachtheory.2014.10.006.

[34] Ezekiel G. Merriam, Kyler A. Tolman, and Larry L. Howell. "Integration of advanced stiffness-reduction techniques demonstrated in a 3D-printable joint". In: *Mechanism and Machine Theory* 105 (2016), pp. 260–271. ISSN: 0094114X. DOI: 10.1016/j.mechmachtheory.2016.07.009. URL: http://dx.doi.org/10.1016/j.mechmachtheory.2016.07.009.

[35] Ezekiel G. Merriam et al. "The design of a fully compliant statically balanced mechanism". In: *Proceedings of the ASME Design Engineering Technical Conference* 6 A.August (2013). DOI: 10.1115/DETC2013-13142.

[36] Femke M. Morsch and Just L. Herder. "Design of a generic zero stiffness compliant joint". In: *Proceedings of the ASME Design Engineering Technical Conference* 2.PARTS A AND B (2010), pp. 427–435. DOI: 10.1115/DETC2010-28351.

[37] Hari Nair Prakashah and Hong Zhou. "Synthesis of constant torque compliant mechanisms". In: *Journal of Mechanisms and Robotics* 8.6 (2016), pp. 1–8. ISSN: 19424310. DOI: 10.1115/1.4034885.

[38] Bhavanam Praveen Reddy and Hong Zhou. "SYNTHESIZING BIDIRECTIONAL CONSTANT TORQUE COMPLIANT MECHANISMS". In: (2018), pp. 1–10.

[39] Pierre Renaud. "TOWARDS STATICALLY BALANCED COMPLIANT JOINTS USING MULTIMATERIAL 3D PRINTING". In: (2016), pp. 1–10.

[40] Jelle Rommers, Giuseppe Radaelli, and Just L. Herder. "A design tool for a single vertex compliant-facet origami mechanism including torsional hinge lines". In: *Journal of Mechanisms and Robotics* 9.6 (2017). ISSN: 19424310. DOI: 10.1115/1.4038008.

[41] Davide Russo and Antonio Caputi. "How to Classify Compliant Mechanisms". In: *Lecture Notes in Mechanical Engineering* (2020), pp. 552–564. ISSN: 21954364. DOI: 10.1007/978-3-030-31154-4_47.

[42] Robert T Sataloff, Michael M Johns, and Karen M Kost. *Foundations of ultraprecision mechanism design*. ISBN: 9781626239777.

[43] Mark Schenk and Simon D. Guest. "On zero stiffness". In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 228.10 (2014), pp. 1701–1714. ISSN: 20412983. DOI: 10.1177/0954406213511903.

[44] Parviz Soroushian, Habibur Chowdhury, and Ali Nossoni. "Design and experimental verification of pseudoelastic-based constant-force springs". In: *Journal of Intelligent Material Systems and Structures* 14.8 (2003), pp. 475–481. ISSN: 1045389X. DOI: 10.1177/104538903038018.

[45] T. Tarnai. "Zero stiffness elastic structures". In: *International Journal of Mechanical Sciences* 45.3 (2003), pp. 425–431. ISSN: 00207403. DOI: 10.1016/S0020-7403(03)00063-8.

[46] Monik Thanaki and Hong Zhou. "Synthesizing bidirectional constant torque compliant mechanisms using precompressed beams". In: *ASME International Mechanical Engineering Congress and Exposition, Proceedings (IMECE)* 4A-2018 (2018), pp. 1–9. DOI: 10.1115/IMECE2018-86469.

[47] Brian P. Trease, Yong Mo Moon, and Sridhar Kota. "Design of large-displacement compliant joints". In: *Journal of Mechanical Design, Transactions of the ASME* 127.4 (2005), pp. 788–798. ISSN: 10500472. DOI: 10.1115/1.1900149.

[48] V.I. Vetrenko. “Application of three-ribbon elastic support with quasi-zero reactive torque in test benches”. In: *Izvestiâ vysših učebnyh zavedenij. Priborostroenie* (2017), pp. 742–752. ISSN: 00213454. DOI: 10.17586/0021-3454-2017-60-8-742-752.

[49] Piyu Wang, Sijie Yang, and Qingsong Xu. “Design and Optimization of a New Compliant Rotary Positioning Stage with Constant Output Torque”. In: *International Journal of Precision Engineering and Manufacturing* 19.12 (2018), pp. 1843–1850. ISSN: 20054602. DOI: 10.1007/s12541-018-0213-x.

[50] Qizi Yang, Shusheng Bi, and Tian Liu. “Static balancing of flexural pivots with two symmetrically arranged pre-compressing springs”. In: *IOP Conference Series: Materials Science and Engineering* 892.1 (2020). ISSN: 1757899X. DOI: 10.1088/1757-899X/892/1/012121.

[51] J. G.H. Yee, O. Soykasapt, and S. Pellegrino. “Carbon fibre reinforced plastic tape springs”. In: *Collection of Technical Papers - AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference* 5.April (2004), pp. 3305–3313. ISSN: 02734508. DOI: 10.2514/6.2004-1819.

[52] Hongzhe Zhao et al. “Analysis and evaluation of a near-zero stiffness rotational flexural pivot”. In: *Mechanism and Machine Theory* 135 (2019), pp. 115–129. ISSN: 0094114X. DOI: 10.1016/j.mechmachtheory.2019.02.003. URL: https://doi.org/10.1016/j.mechmachtheory.2019.02.003.

B

Additional analyses

A number of additional analyses have been performed that were either used in the design of the joint or used for better prediction and understanding of its performance. These analyses can be found in this appendix.

B.1. Parametric sweep

In this section, the performed parametric sweeps are shown. For each figure, the top and bottom rows show a 2D perspective and isometric perspective respectively. The left column shows the energy field for a 50% smaller parameter value, the middle column is the original value and the right column shows the energy field for a 50% larger parameter value. The material used for the simulated joint in the parametric sweep is PLA, which affects the height of the energy field.

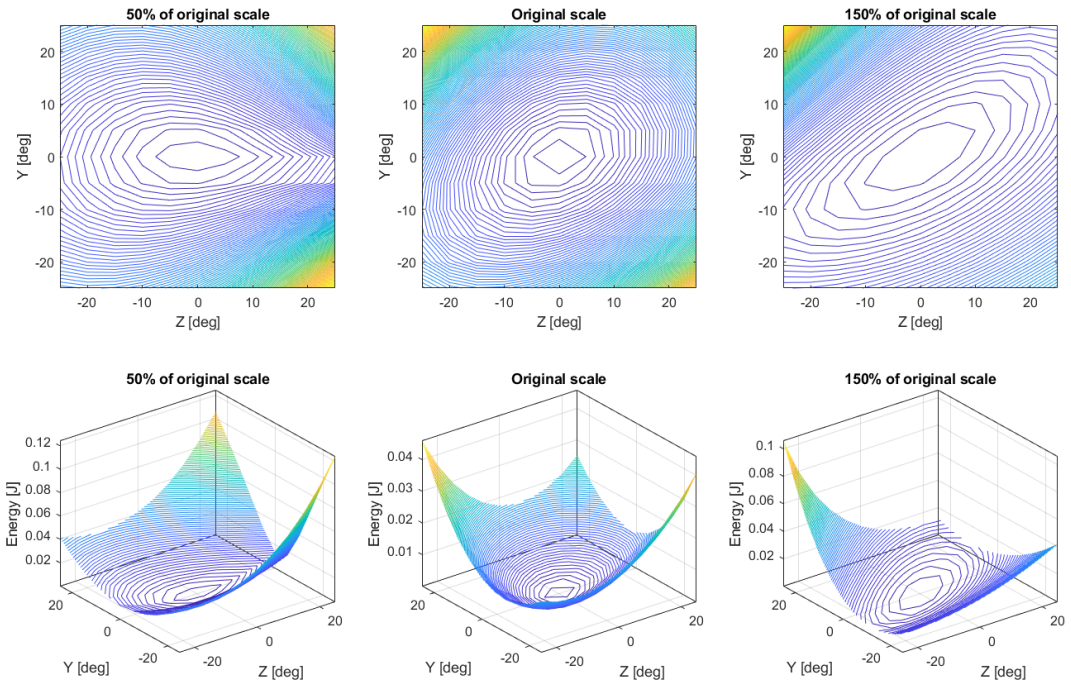


Figure B.1: Parametric sweep of parameter γ

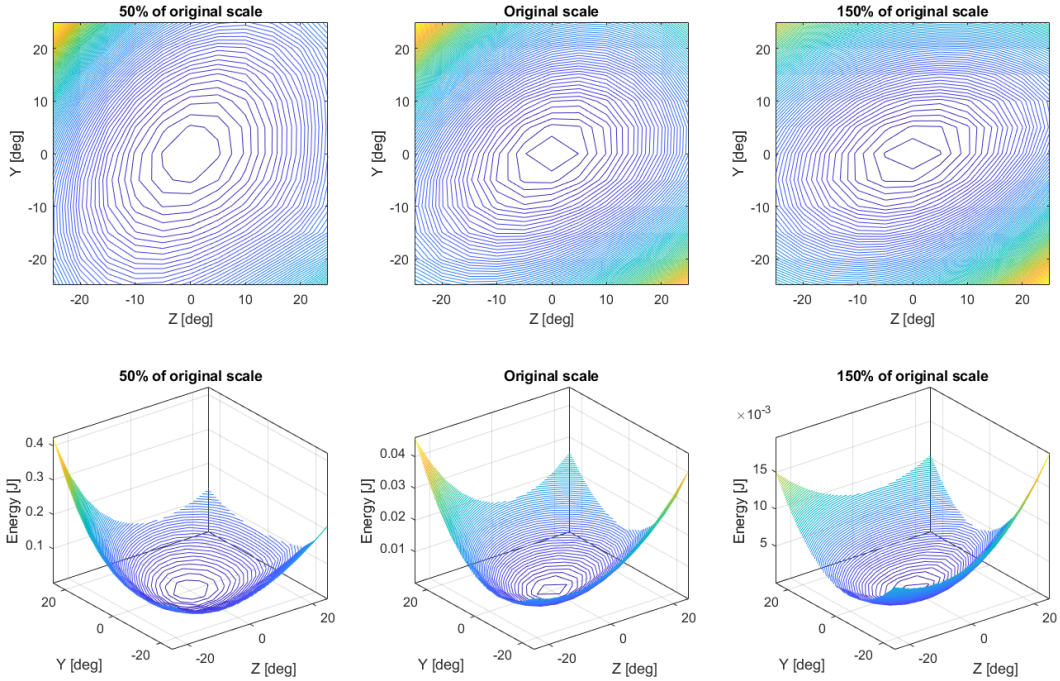


Figure B.2: Parametric sweep of parameter α

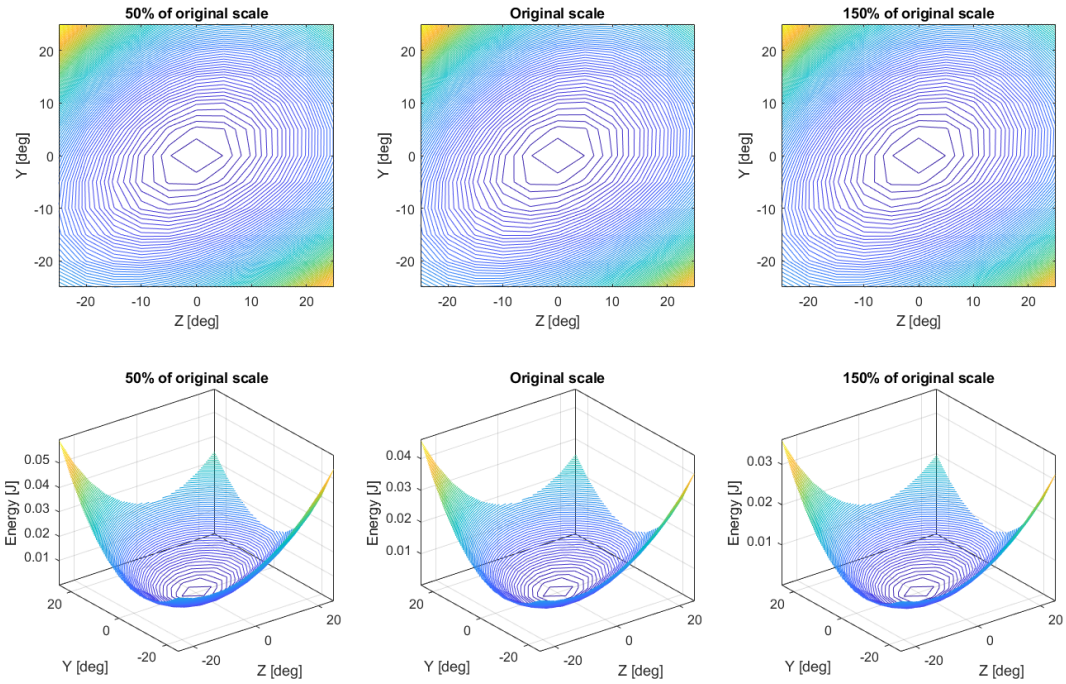


Figure B.3: Parametric sweep of parameter β

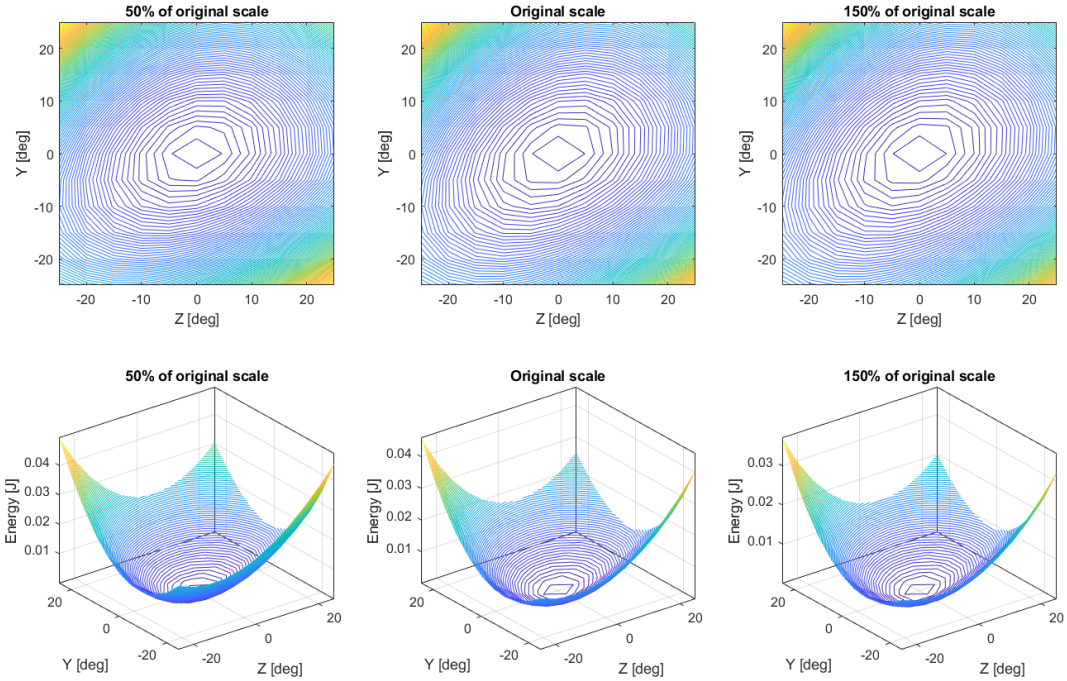


Figure B.4: Parametric sweep of parameter T

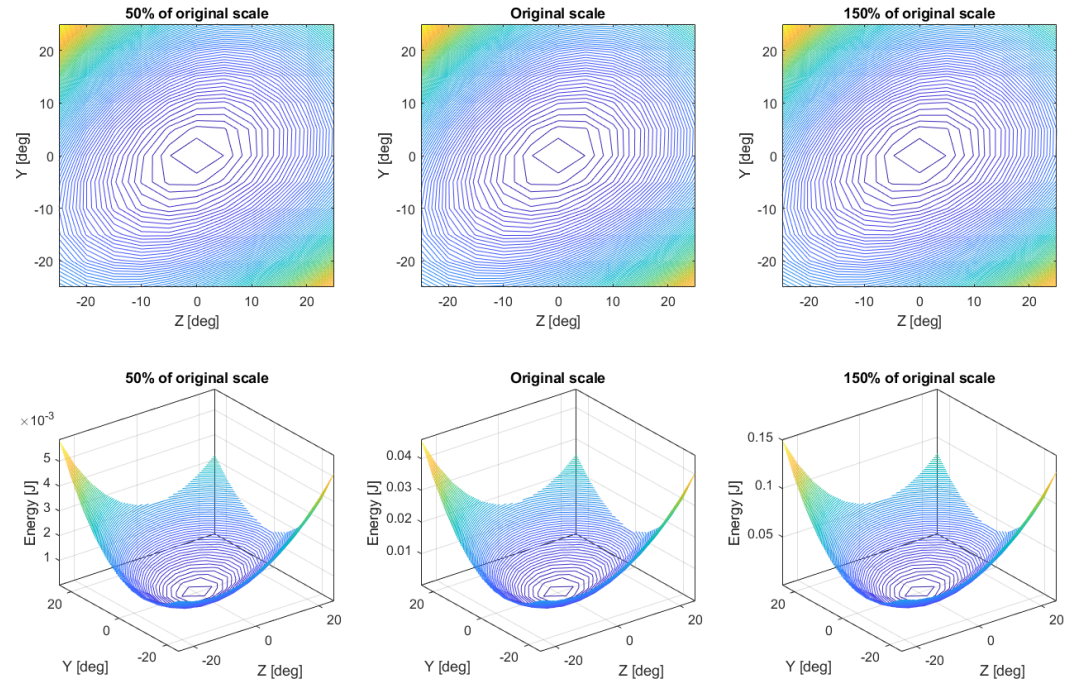


Figure B.5: Parametric sweep of parameter d

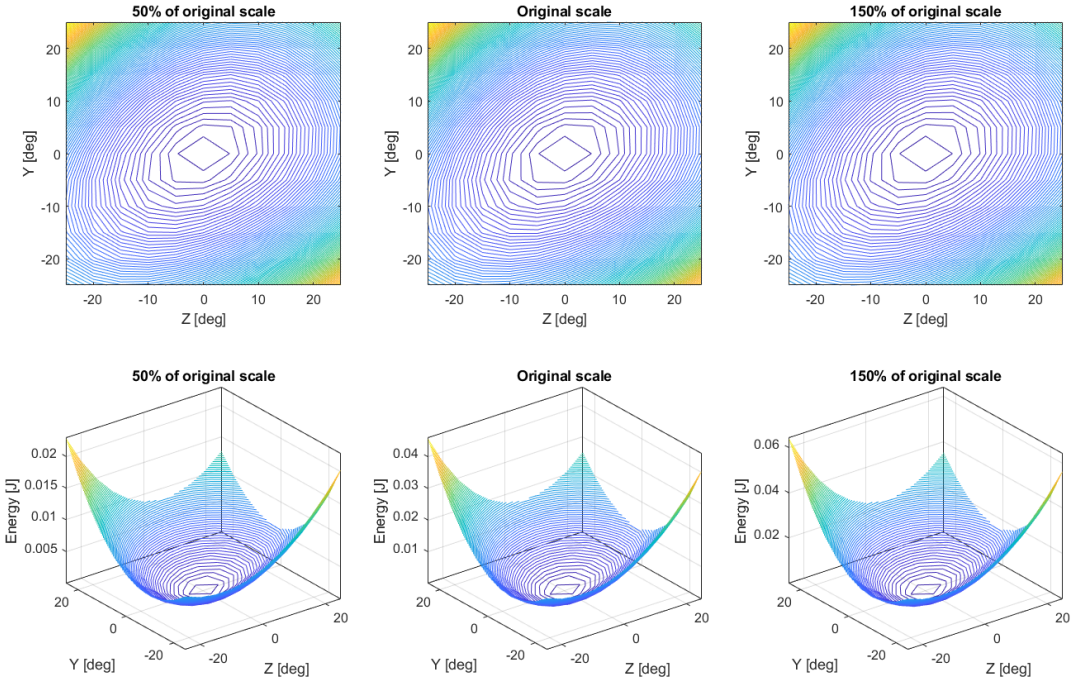


Figure B.6: Parametric sweep of parameter ΔR

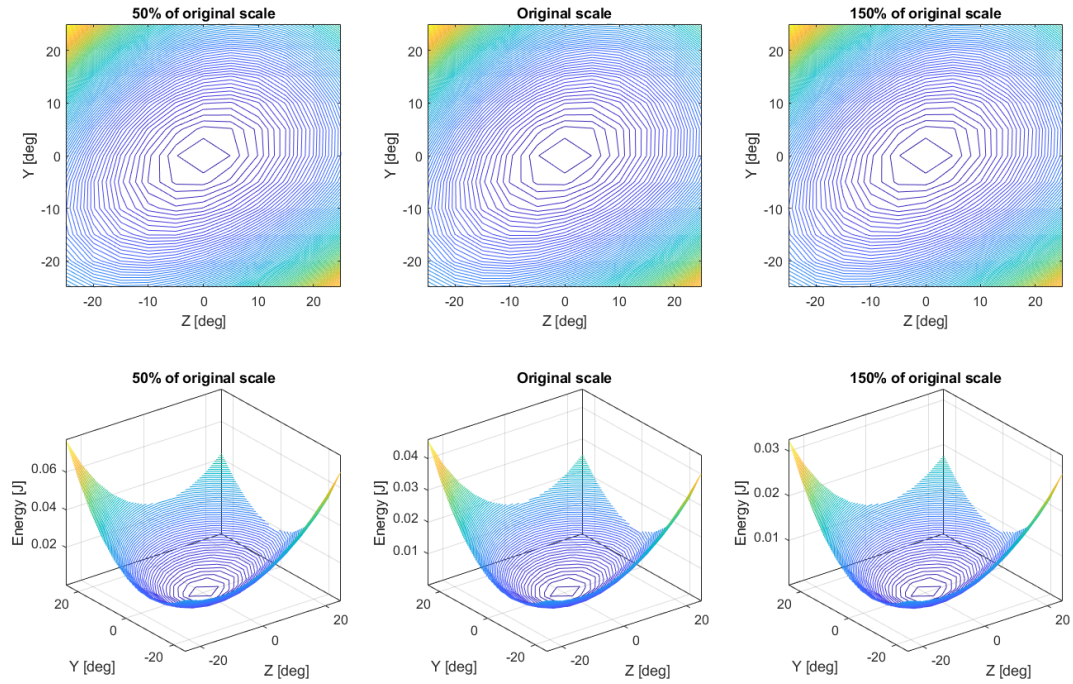


Figure B.7: Parametric sweep of parameter R_{in}

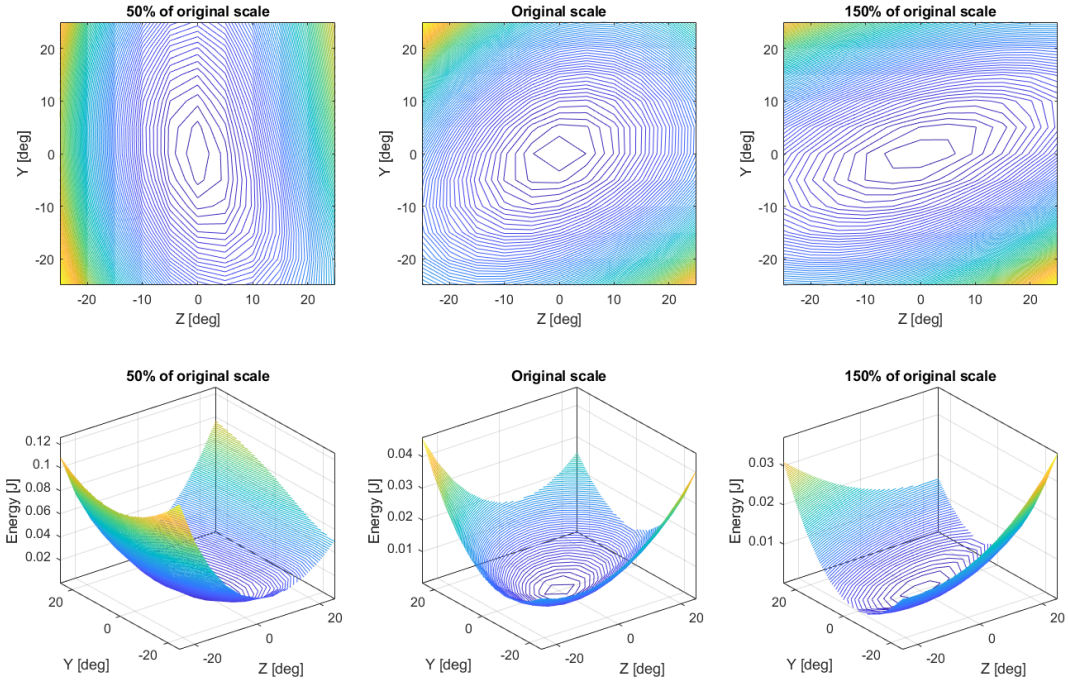


Figure B.8: Parametric sweep of parameter α_2

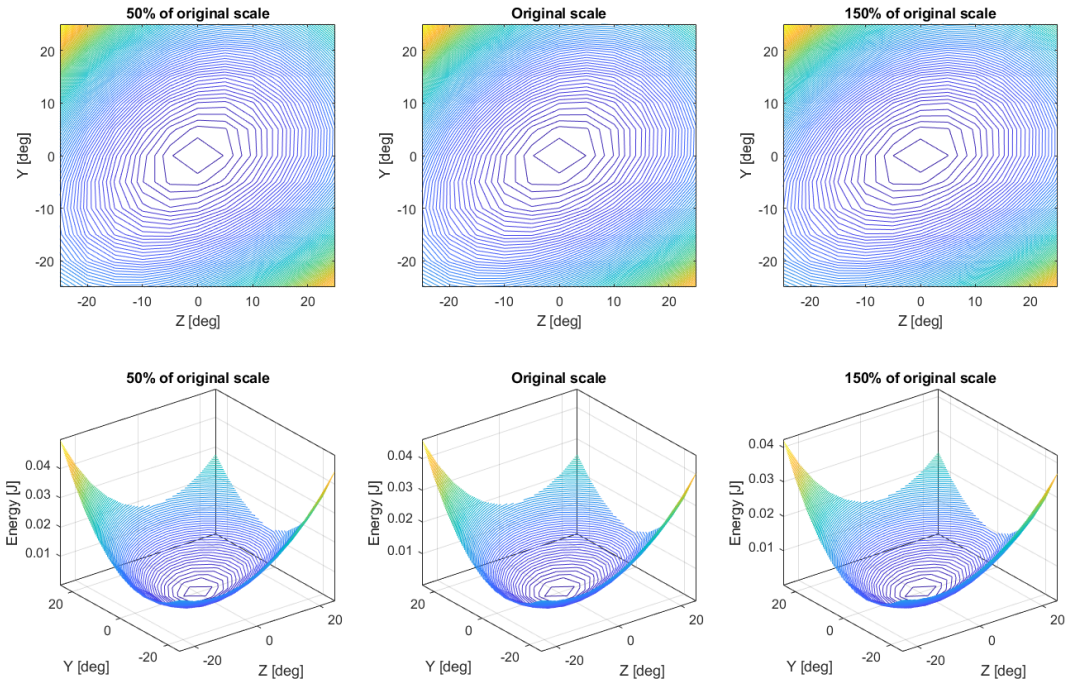


Figure B.9: Parametric sweep of parameter β_2

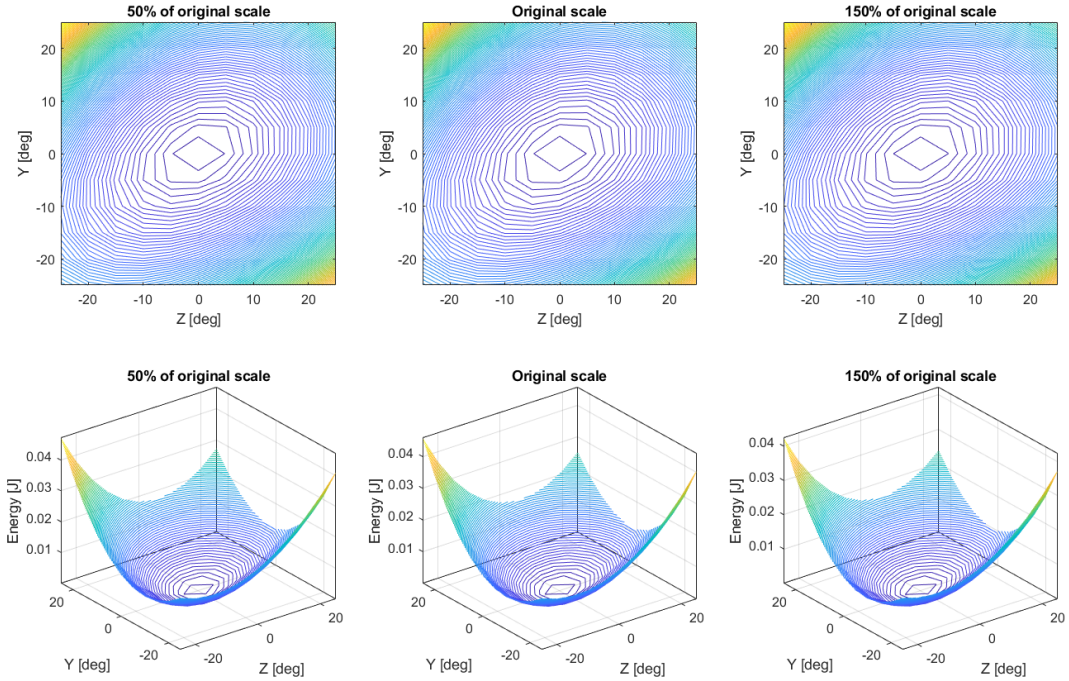


Figure B.10: Parametric sweep of parameter T_2

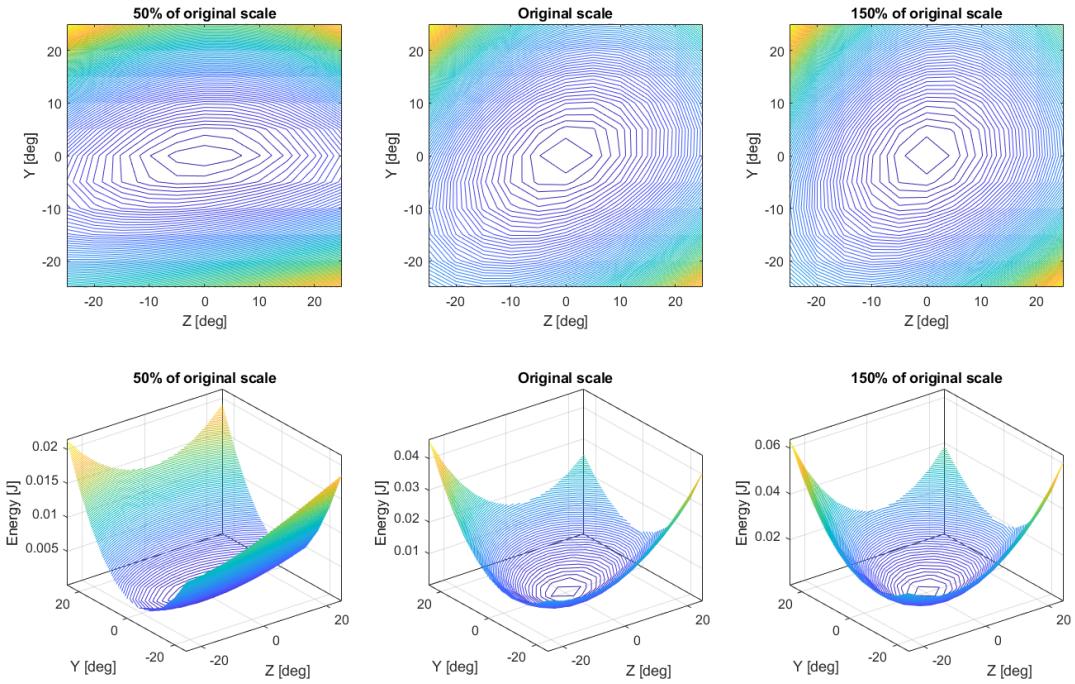


Figure B.11: Parametric sweep of parameter d_2

B.2. Complete sensitivity analysis

This section contains all sensitivity analyses, including the influential ones that have been used as optimization variables. Each figure shows the normalized RMS error of an axi-symmetrical energy field for a parameter variation of $\pm$50%.

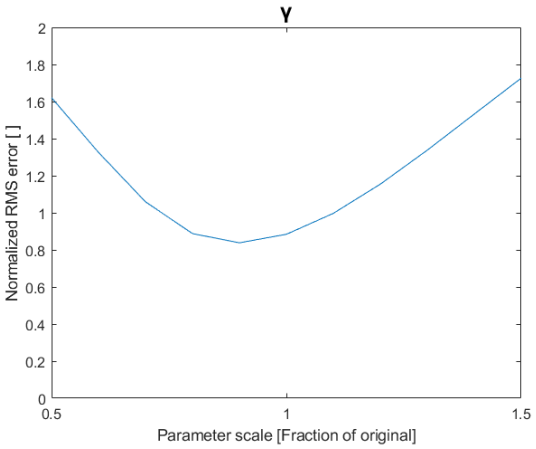


Figure B.12: Sensitivity of parameter γ

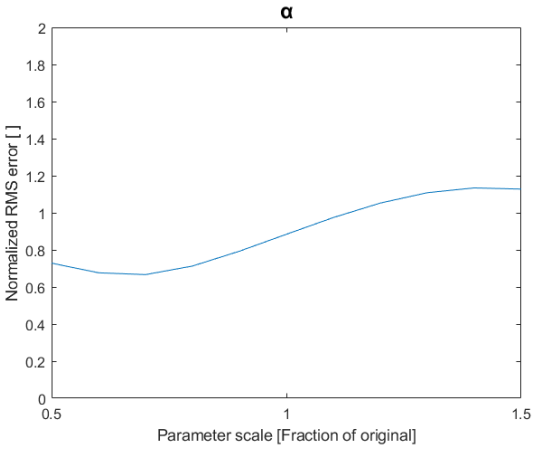


Figure B.13: Sensitivity of parameter α

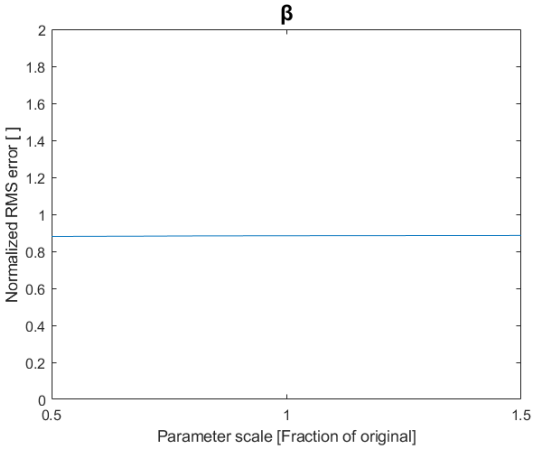


Figure B.14: Sensitivity of parameter β

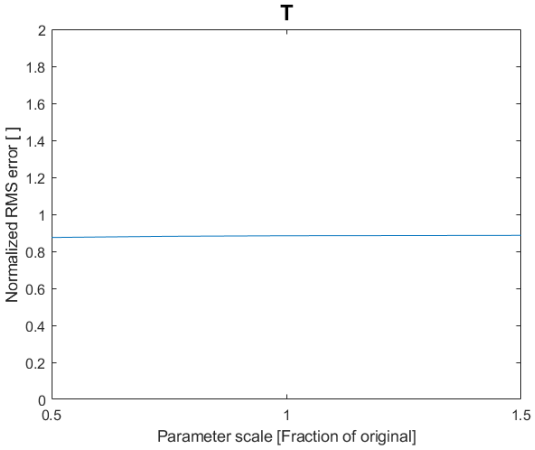


Figure B.15: Sensitivity of parameter T

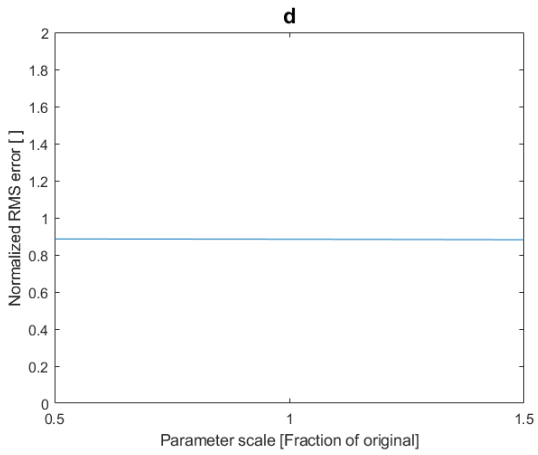


Figure B.16: Sensitivity of parameter d

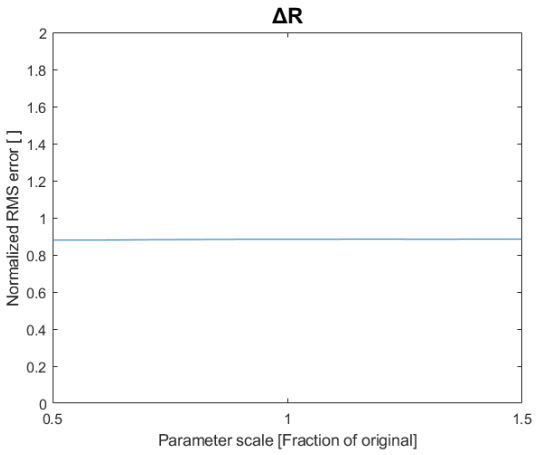


Figure B.17: Sensitivity of parameter ΔR

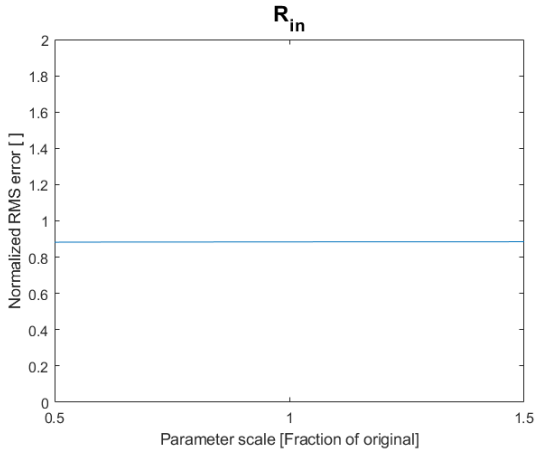


Figure B.18: Sensitivity of parameter R_{in}

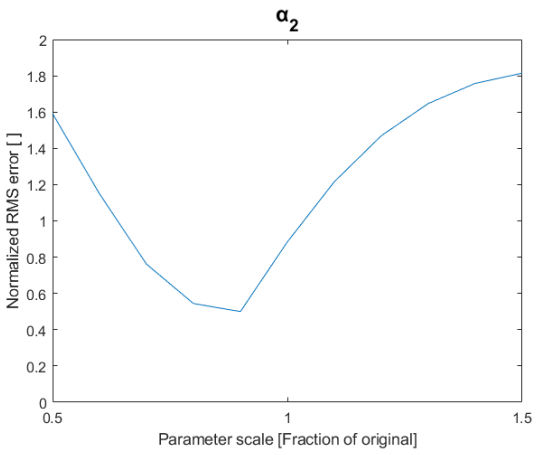


Figure B.19: Sensitivity of parameter α_2

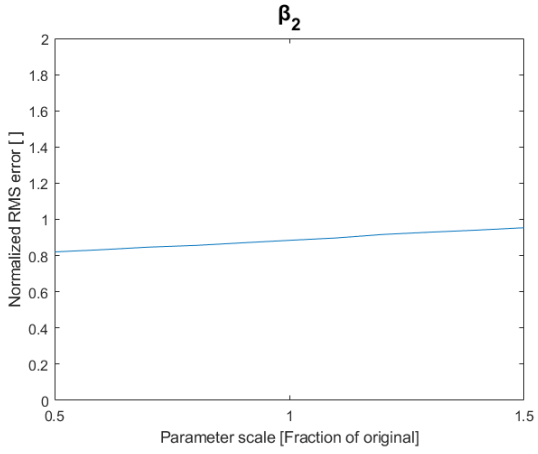


Figure B.20: Sensitivity of parameter β_2

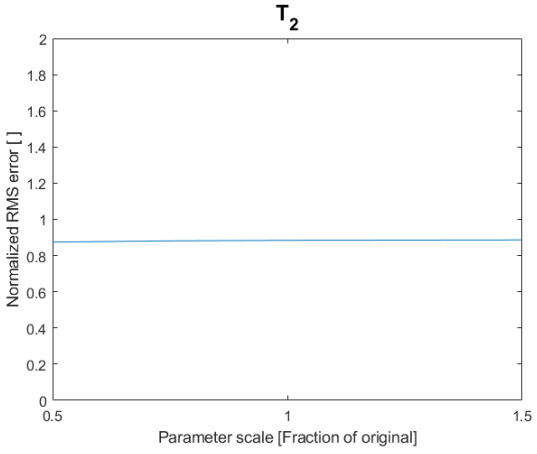


Figure B.21: Sensitivity of parameter T_2

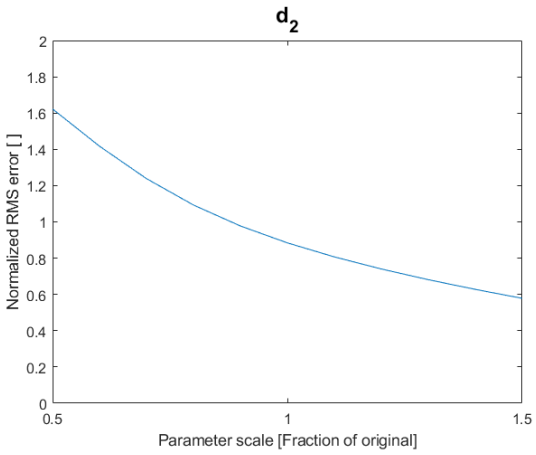


Figure B.22: Sensitivity of parameter d_2

B.3. Free spring length analysis

The effect of the free spring length L_0 on the desired energy field of the central spring, is analysed in this section. From the spring energy equation (eq. 8 in the paper), the error caused by L_0 might seem significant for values that are not close to zero. Without optimization of variables, this is indeed the case. However, after optimization of the spring stiffness k_s and spring attachment location x_0, it is possible to minimize and predict the error.

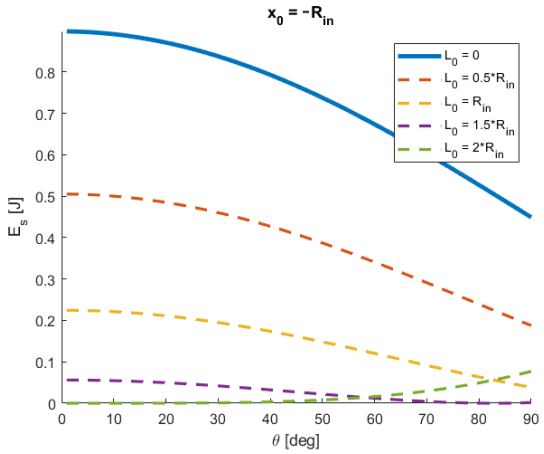


Figure B.23: Effect of L_0 on E_s, for $x_0 = -R_{in}$

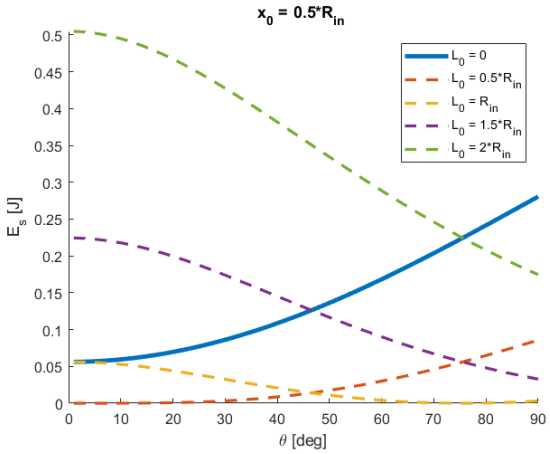


Figure B.24: Effect of L_0 on E_s, for $x_0 = 0.5R_{in}$

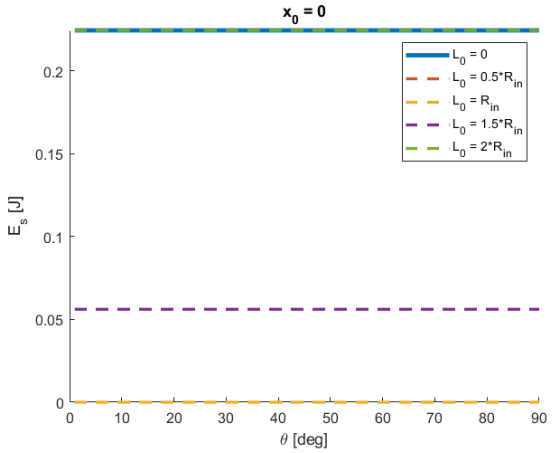


Figure B.25: Effect of L_0 on E_s, for $x_0 = 0$

First, to gain understanding of the situation, a number of configurations are discussed that show the effect of L_0 without optimized values. The figures in this section show the relation between the spring energy E_s and the rotation angle θ for various values of L_0 and spring attachment point x_0. The spring stiffness k_s is set at an arbitrary realistic value of 100 N/m in this analysis. Because the function shape is affected only by relative values, L_0 and x_0 are given as a function of inner joint radius R_{in}. A constant value of 0.067 m is attributed to R_{in}.

For negative values of x_0 (Fig. B.23), the spring becomes shorter with increasing angle and thus should be a tension spring to display a decreasing energy function. The graph corresponding to $L_0 = 0$ shows the ideal energy function without error. It can be seen that larger values of L_0 result in a lower and less steep energy function that deviates further from the desired shape. At a certain point, the function becomes increasing instead of decreasing. This is the point where L_0 becomes equal or larger than the spring length L. In that case, the spring behaves as a compression spring that becomes more

compressed with increasing angle, resulting in increasing energy. For positive values of x_0 (Fig. B.24), the spring elongates with increasing angle and should therefore be a compression spring to display a decreasing energy function. In this situation, a certain value of L_0 is required in order to let the spring behave as a compression spring. For too small values of L_0 the function is increasing instead of decreasing. This occurs at the point where $L = L_0$ and thus where the energy is zero. After that point, the spring continues to elongate while it is no longer in a loading mode of compression, but tension. For sufficiently high values of L_0, the point where the switch occurs is far enough outside of the range of motion such that the desired energy function can be approximated. The functions created from $x_0 = 0$ reveal that this is a special case where the energy is constant for every value of L_0 (Fig. B.25). This can be understood when considering that this value of x_0 coincides with the center of rotation. Thus, for any rotation, the spring will maintain the same length and therefore also constant energy.

When the parameters x_0 and k are optimized, the error is significantly smaller. Fig. B.26 displays the optimized error of both a compression and tension spring as the average deviation from the desired energy function in a range of motion of 30 degrees, expressed as a percentage. The error is relative to the energy function corresponding to the theoretical best optimized spring parameters from Table IV in the paper. Moreover, the functions are normalized before the error calculation, as to only calculate the error in function shape and not in function magnitude. As can be seen in Fig. B.26 below, the error in a compression spring is large for values of L_0/R_{in} below 1. This can be understood when looking at the function shape for low values of L_0/R_{in} in Fig. B.24, where it is increasing instead of decreasing. This analysis has shown that the error caused by L_0 can be negligible after optimization

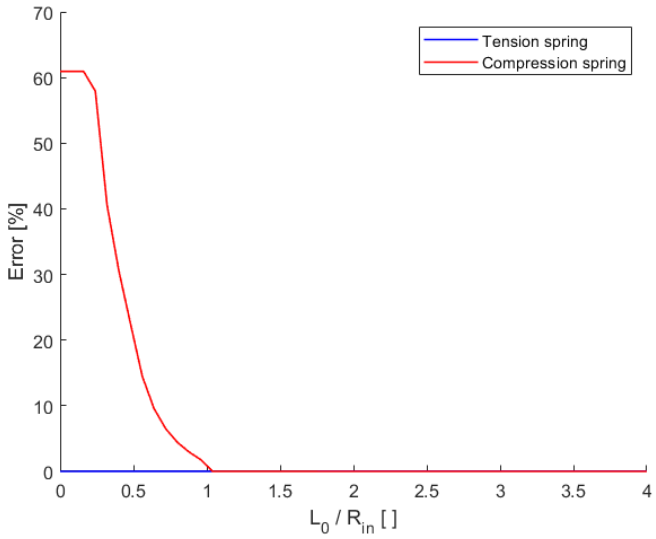


Figure B.26: The error of a tension and compression spring function relative to the desired shape, as a function of L_0/R_{in}.

B.4. Range of motion and neutral stability

The effect of the range of motion on the neutral stability of the optimized joint is analysed in this section. A measure of neutral stability is given in terms of energy reduction. This is calculated by dividing the RMSE of the neutrally stable energy field, by the reference energy field and converting that to a percentual difference. Here, the reference energy field is the optimized axi-symmetrical energy field of the joint. The RMSE is relative to a constant energy field over the range of motion. For each selected range of motion, a separate optimization has been carried out for the spring. No constraints were set on the spring optimization for this analysis, therefore practical considerations such as stress limits were not considered. The results can be seen in Fig. B.27, each red dot corresponds to a data point. The range of motion is given in terms of single-sided rotation. In other words, as the rotation in any direction from the neutral position.

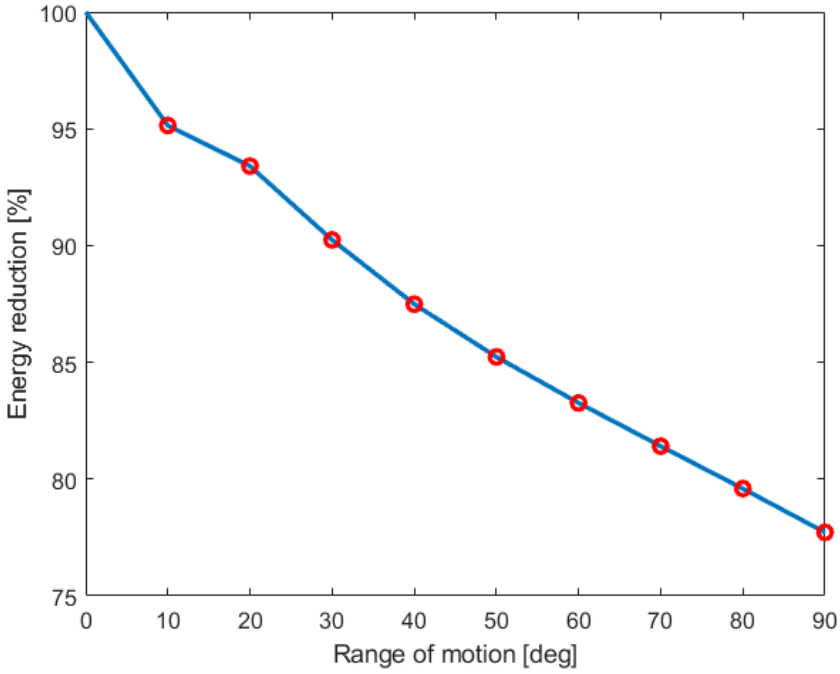


Figure B.27: Energy reduction vs range of motion of the joint

B.5. Simplified axi-symmetrical energy field analysis

This appendix is a more detailed attempt to understand the working principle behind the axi-symmetry of the energy field. A simplified version of the optimized joint is visualized in Fig. B.28 with the intention to achieve intuitive understanding of the working principle. The first arm is discretized into multiple sections of rigid parts with infinitesimal dimensions (red) and flexible parts (blue). The flexible parts have a relatively high torsional stiffness compared to the bending stiffness, just as in the real joint. In this illustration, let us look at the extreme case in which torsional stiffness is infinite. Each flexure has the same dimensions and is made out of the same material and therefore all flexures have equal stiffness k. Flexures 1 and 2 are angled 90 degrees relative to each other, just as flexures 3 and 4. The second arm of the joint is considered as rigid in this simplification, since the optimization resulted in a shell thickness of more than three times the shell thickness of the first arm. Furthermore, the arms are constrained by the tetrahedra to only move around the center of rotation (P), which must be taken into account in this simplified model. This is done by having the requirement that the green surfaces A, B and C must be tangential to the spherical path around P. These surfaces belong to the intermediate rigid body between flexures 2 and 3, the fixed joint support and the joint tip respectively.

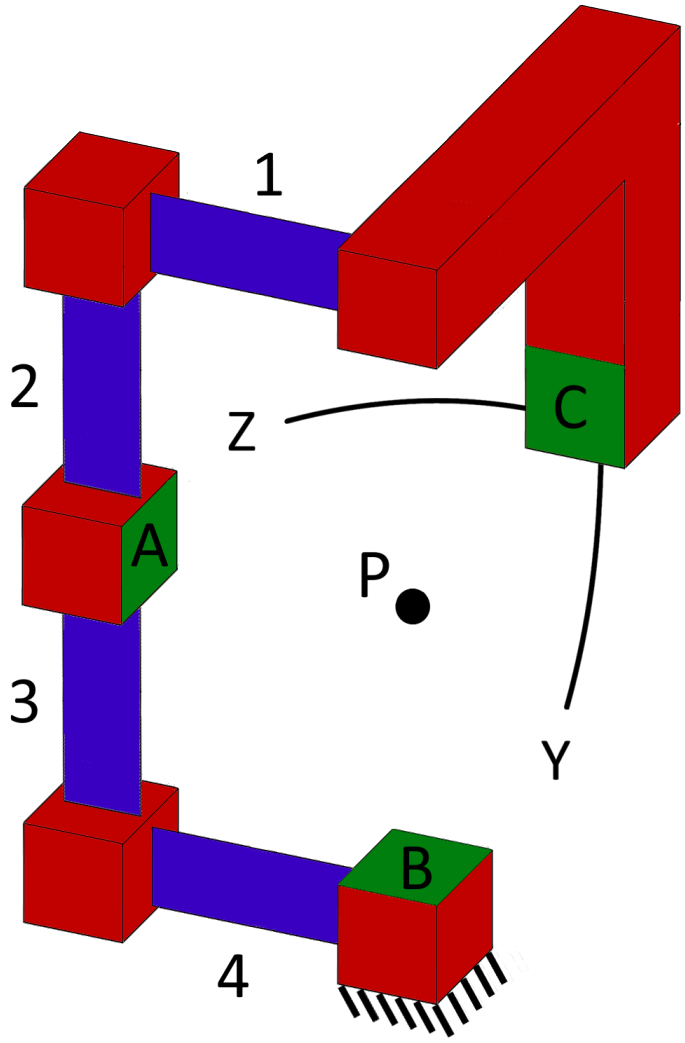


Figure B.28: The simplified model of the joint.

Now to illustrate the axi-symmetrical energy field of the model, a static analysis is performed for multiple deformation modes. Let us first consider a forced rotation θ purely in the spherical Z axis direction as illustrated in fig. B.29. This rotation will cause flexures 1 and 4 to bend by an angle ϕ. Because flexures 1 and 4 accomodate a single forced rotation and are equal flexures in series, they deflect by the same angle. In the deformed configuration, a straight line can be drawn from the base to tip of the flexure. This line is defined as L^* and will have a small deviation from the true undeformed length of the flexure L. The bending angle is also not constant over the length of the flexure, therefore an angle α is defined between the undeflected flexure and L^*. Flexures 2 and 3 will experience a bending moment as well and rotate by angle ψ and have an angle β between the undeflected flexure and L^*. Since surface A must remain in the spherical surface around P, it is forced to displace by a distance δ such that it is again aligned with the spherical surface.

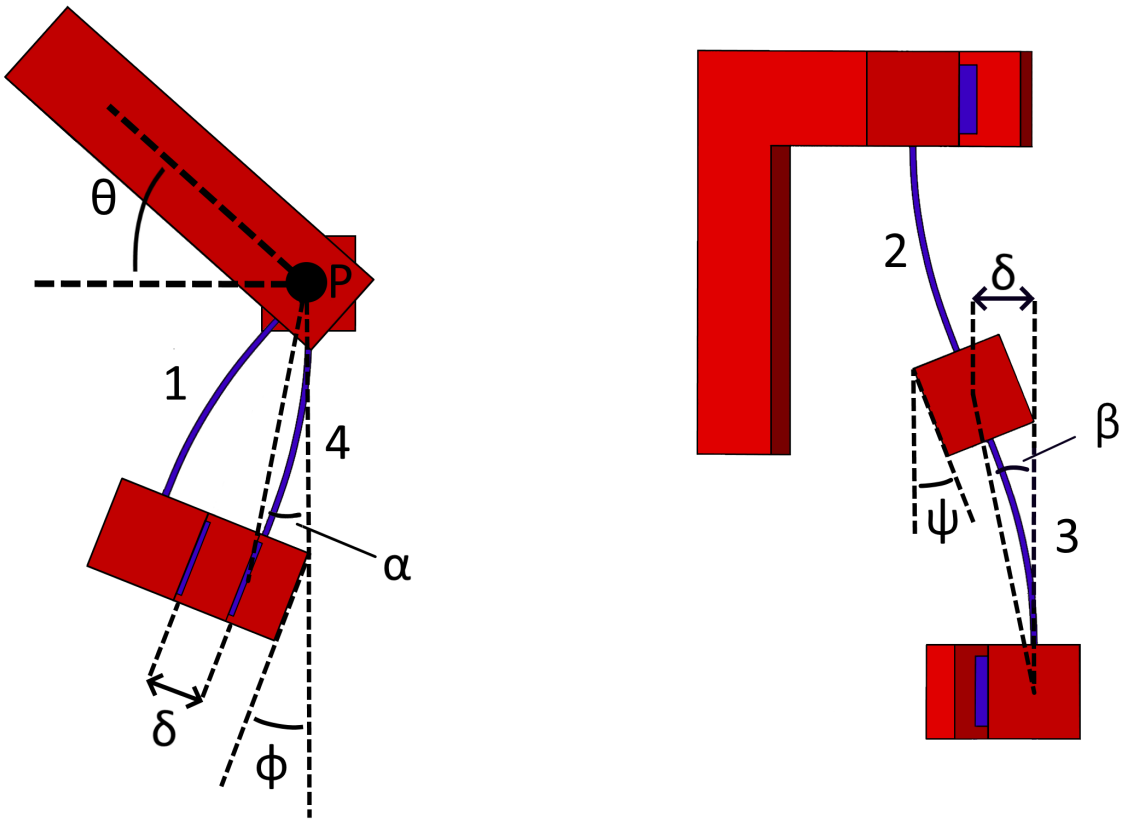


Figure B.29: The deformation caused by the loading mode where a rotation is forced in the spherical Z axis. Left: top view, right: front view.

Angles α and β can be related to the maximum bending angle by considering its non-linear trigonometric relationship. First, the nonlinear deflection in both longitudinal and normal axis of an arbitrary flexure under a moment load are given by equations B.1 and B.2 [7]. Here ξ represents the tip bending angle of the flexure.

$$\delta_y(\xi) = \frac{L}{2}\xi \tag{B.1}$$

$$\delta_x(\xi) = \frac{L}{2}(1 - \frac{\xi}{tan(\xi)}) \tag{B.2}$$

By using trigonometric relations and equations B.1 and B.2 for a range of motion of 30 degrees in either direction, the following can be formulated:

$$\alpha = tan^{-1}(\frac{\delta_y(\phi)}{L - \delta_x(\phi)}) \approx \frac{\phi}{2} \tag{B.3}$$

$$\beta = tan^{-1}(\frac{\delta_y(\psi)}{L - \delta_x(\psi)}) \approx \frac{\psi}{2} \tag{B.4}$$

From the trigonometric relations in Fig. B.29, the angles α and β can be related to each other (eq. B.5 & B.6).

$$sin(\frac{\delta}{L^*}) = \phi - \alpha \tag{B.5}$$

$$sin(\frac{\delta}{L^*}) = \beta \tag{B.6}$$

From equations B.3, B.4, B.5 and B.6 it is clear that $\phi = \psi$ on approximation. Only flexures 1 and 4 can bend in the plane of the forced rotation θ and therefore the sum of their rotations must result in the total rotation. Because these flexures deflect by the same angle ϕ and because it is known that $\phi = \psi$, the relation between the angles can be derived (eq. B.7).

$$\psi = \phi = \frac{\theta}{2} \tag{B.7}$$

From here the total non-linear strain energy stored in the flexures can be defined by eq. B.8.

$$E_Z = \sum_{i=1}^{N} k(1 - cos(\Theta(i))) = 2(k(1 - cos(\phi)) + 2(k(1 - cos(\psi)) = k(4 - 4cos(\frac{\theta}{2})) \tag{B.8}$$

Secondly, let us consider a pure rotation of θ along the spherical Y axis (Fig. B.30). This would result in a bending of flexures 2 and 3 by angle ϕ. The rotation causes a translational misalignment of surface A by δ with the spherical surface. To align surface A with the spherical surface again, it can either shift back with a translation or make a rotation. Simulations show that the latter happens. This rotation is facilitated by bending of flexure 1 and 4 by angle ψ, as seen from the top perspective. The same logic as before can be applied to the new configuration. Equations B.3, B.4, B.5 and B.6 still hold true and describe some of the angular relationships for the new configuration. Flexure 2 and 3 facilitate the total forced rotation θ. However, these flexures are rotated out of its undeformed plane by the angle ψ, while still being forced to rotate by θ in its original plane. Therefore, the absolute value of the rotation angle for these flexures is defined in eq. B.9.

$$\gamma = \frac{\theta}{cos(\psi)} \tag{B.9}$$

Then, by following the same argumentation as in eq. B.7, eq. B.10 can be formulated

$$\psi = \phi = \frac{\gamma}{2} \tag{B.10}$$

To keep the rigid arm aligned to its original plane, flexure 1 needs to rotate by an angle ψ as seen from the top view. However, since the flexure is rotated out of plane by an angle γ, the absolute rotation magnitude of flexure 1 (ζ) is larger, as seen in eq. B.11.

$$\zeta = \frac{\psi}{cos(\gamma)} \tag{B.11}$$

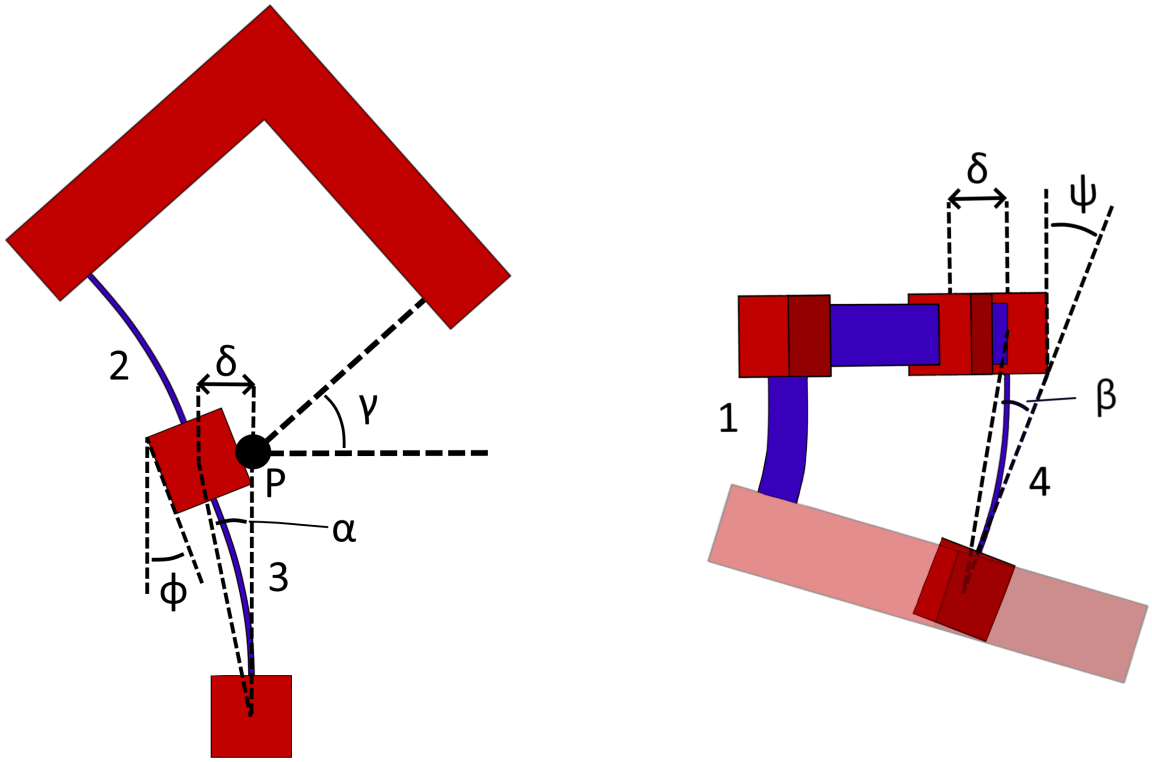


Figure B.30: The deformation caused by the loading mode where a rotation is forced in the spherical Y axis. Left: front view, right: top view.

Because flexures 1 and 4 are originally of equal length and are angled by ψ relative to the rigid arm and fixed world, the rigid arm is kinematically forced to rotate as well. This rotation is not in the same plane as ψ, since that would violate a pure rotation in Y-direction, but around the axis perpendicular to the joint tip (Surface C). This effect is lessened because flexure 1 has a larger rotation angle, and thus smaller effective length, relative to flexure 4. Thus, the length difference that has to be accomodated by the rigid arm through its rotation, is smaller. The total strain energy stored in the flexures can be described by eq. B.12.

$$E_Y = \sum_{i=1}^{N} k(1 - cos(\Theta(i))) = 2(k(1 - cos(\phi)) + k(1 - cos(\psi)) + k(1 - cos(\zeta)) \tag{B.12}$$

Substituting equations B.10 and B.11 and rewriting as a function of ϕ leads to eq. B.13.

$$E_Y = k(4 - 3cos(\phi) - cos(\frac{\phi}{cos(2\phi)})) \tag{B.13}$$

When comparing the strain energy of both load cases, it can be seen that the two scale dependent on ϕ and θ (eq. B.14). The exact value of ϕ can be found by substituting eq. B.9 in eq. B.10 and solving the resulting equation.

$$\frac{E_Y}{E_Z} = \frac{1 - \frac{3}{4}cos(\phi) - \frac{1}{4}cos(\frac{\phi}{cos(2\phi)})}{1 - cos(\frac{\theta}{2})} \tag{B.14}$$

Thus, according to eq. B.14 the deviation in axial symmetry between the two perpendicular directions is zero at a rotation angle of zero degrees and builds up to a maximum of 14.9% at a 30 degrees angle. Meaning that the energy in Y-direction at 30 degrees is 14.9% larger than the energy in Z-direction at 30 degrees. In the simulation this difference is 8.25% maximum. The simplified model does not account for a shift in the center of rotation or the bending energy of the tetrahedra. It is possible that these simplifications are the cause for the discrepancy in results. Interestingly, if the value of ϕ were to be taken as $\frac{\theta}{2}$ just as in the Z-direction loading mode, eq. B.14 would give a value of 8.27%.

C

Concept generation

The design progress and the concepts that were generated are shown in this appendix. Initially, a number of concepts were designed that are not based on the foundations of the Tetra I joint. These were not further developed than the basic working principle, concept sketch and the occasional prototype. Nevertheless, they are briefly described here as well. The concepts after that are based on the Tetra I joint and are more closely related to the final design. These will be elaborated upon in more detail.

C.1. Design progression

In this section a visualization of the concept generation is shown in the form of a design tree (Fig. C.1). It shows all design branches that were considered before choosing the final solution path. Furthermore, it is in approximately chronological order of the designs that were considered.

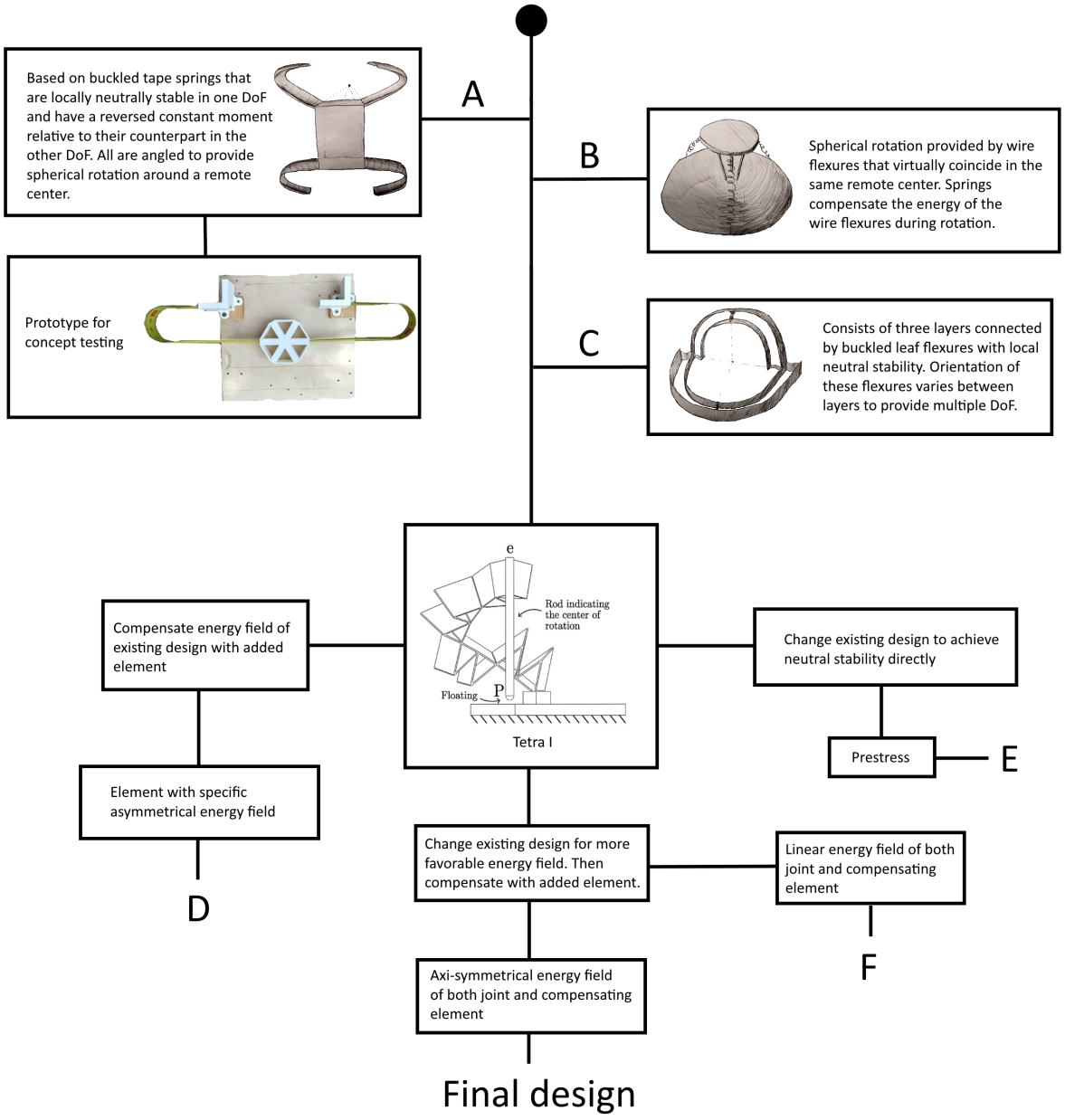


Figure C.1: Design tree that shows the considered design branches.

C.2. Concept A, B and C

These concepts were created before the concepts based on the existing Tetra I joint. A short description and early concept sketches can be found in Fig. C.1. All of these work in theory, but doubts arose about the implementation in practice and how well they would perform. A prototype was created to test the working principle for concept A, with partially successful results, but also many added difficulties that were unforeseen before the prototype. When the already well functioning Tetra I joint was found with the properties of compliant spherical motion and a remote center of rotation, the decision was made to attempt new concept designs based on that.

C.3. Concept D

Neutral stability can be achieved by compensating the energy field of the existing Tetra I joint with an added element. The joint originally has an asymmetrical energy field, therefore the element would need to have an asymmetrical energy field that is mirrored relative to that of the Tetra I. Such an element could be created by using a helical spring that is attached to the end-effector on one side, and to a dynamic attachment point on the other side. Because the attached point is not fixed, it influences the amount of spring elongation and thus the energy in the spring. The dynamic attachment point would have to displace differently with each rotation direction, in order to create an asymetrical energy field. This is visualized in Fig. C.2. The dynamic displacement can be achieved by a compliant element with direction-dependent stiffness. The element itself also requires energy to deform and thus affects the energy field. An example of such a specific compliant element with direction-dependent stiffness is a wire flexure with an oval cross-section. This concept was not pursued further, because the final design of the central spring has a similar working principle, but easier implementation.

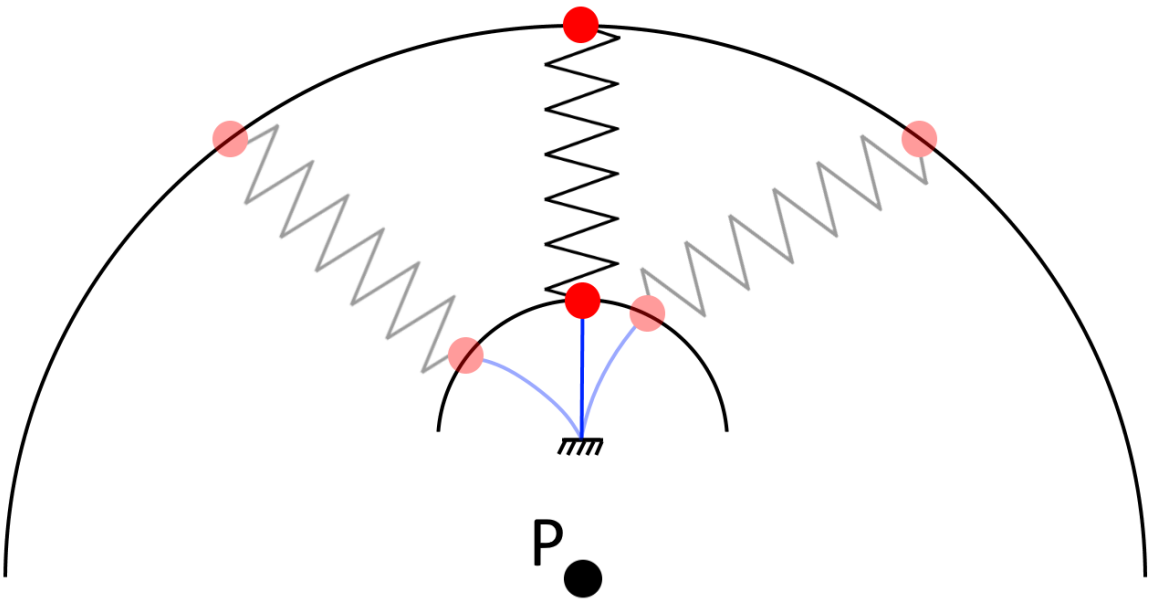


Figure C.2: 2D simplified view of concept D. The spring is attached at the red dots and the semi-circles show the path of the end-effector and compliant element (blue). A few example rotations are given to show the difference in displacement of the compliant element and its attachment point.

C.4. Concept E

By prestressing the end-effector of the original Tetra I in its stiff direction, it should be possible to lower or remove the stiffness in the other directions. The stiffest direction of motion in the joint is towards, or away from, the center of rotation. An attempt was made to show this behaviour with a version of the Tetra I that is adapted for 3D printing (Fig. C.3). While attempting to apply sufficient force, the prototype broke quickly however. After this, an attempt was made to simulate the behaviour with ANSYS. A force was applied in the direction toward the center of rotation, after which the rotation of the joint was simulated. Multiple magnitudes of the force and reversed directions were attempted as well. For lower forces, minor changes could be seen to the energy field, most notably an unwanted shift in the neutral position, but no major flattening of the energy field. Higher applied forces caused immediate convergence issues and stresses in the joint that were higher than the yield stress, as well as significant center shifts. Because of the low feasibility and lack of promising results, this concept was not further explored.

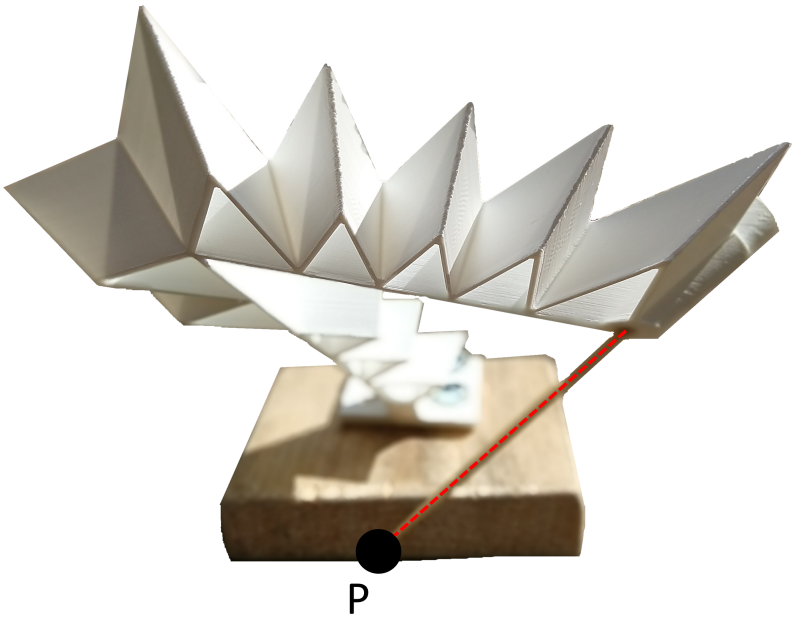


Figure C.3: Tetra I adapted for 3D printing. Tested as concept E with an applied force along the red line.

C.5. Concept F

This concept is based on acquiring a local linear energy field of the joint. If that is achieved, two of those identical joints can be attached to each other at the end-effector in order to add their energy fields. With proper initial deformation before attachment, the reversed linear parts of the energy fields overlap, creating local neutral stability (Fig. C.4). Although promising in theory, no immediate method was found to create a joint with a locally linear energy field, especially in two rotational directions. Therefore, the development of this concept was put on hold indefinitely.

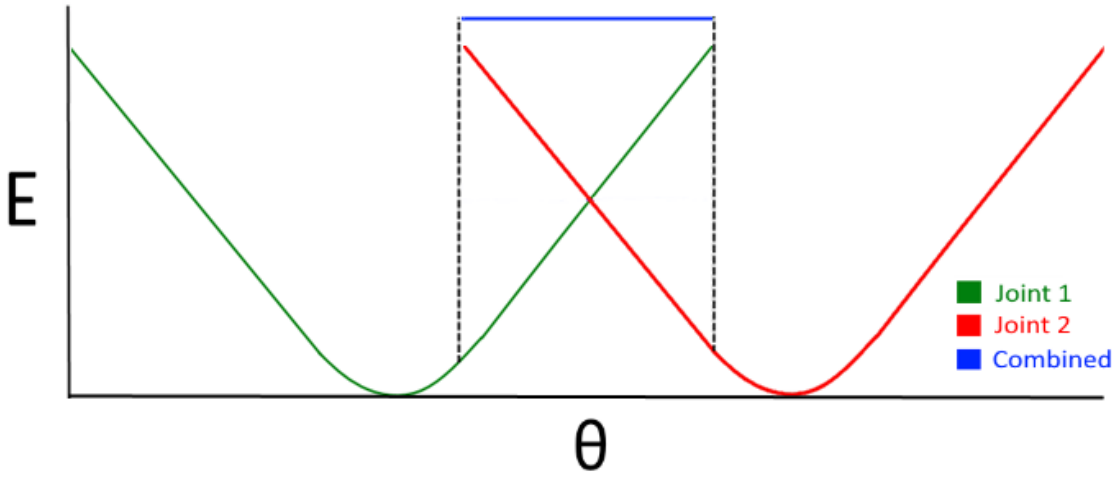


Figure C.4: 2D view of the overlapping energy fields of concept F. The combined part in the center is neutrally stable.

D

Programming and code

In this appendix, all code that has been used during the thesis is provided. Moreover, a brief description of the programming is provided at the start of each section for context and explanations.

D.1. ANSYS APDL model

This section of code is used to simulate the joint and its rotation. It is highly versatile and free to parametric input in a wide range. Multiple simulation options are included for different goals: a single rotation can be simulated, all rotations in an energy field can be simulated as well and simulations for the energy fields of multiple parametric variations are included too. More options include the addition of the spring and the effect of gravity.

```
!   Dion Hogervorst

!!!!    GENERAL INFO

    *DIM, Dir_Log, STRING, 200
    *DIM, Dir_Data, STRING, 200
    *DIM, Dir_Input, STRING, 200
    Dir_Log(1) = JOIN(Dir(1), 'Logs')
    Dir_Data(1) = JOIN(Dir(1), 'ANSYS_DATA')
    Dir_Input(1) = JOIN(Dir(1), 'ANSYS_INPUT')

    /CWD, Dir_Log(1)

FINISH

    /title,  Spherical_model
    /FILNAME, Spherical_model, 1
    /CONFIG, NRES, 1000000
    /units, SI

!!!!    PARAMETERS

    Pi = ACOS(-1)       !pi, 3.14...

    !Material parameters
    !Performance grade Titanium alloy (Ti6Al4V)
    Density = 4390
    Emodulus = 110e9

    !Geometry parameters first arm
    Alpha = 169.3797        !Total angle/length of single arm around point
    of rotation  (Original = ~75)
    Beta = 18               !Angle between arm and peak of
    tetrahedra                   (Original = ~18)
    Gamma = 89.0442         !Angle between the two
    arms                                (Original = ~70)
    T = 4                   !Number of tetrahedra per
    arm                           (Original = 4)
    d = 0.0005              !Shell
    thickness                                          (Original = 0.0005)
    Delta_R = 0.035         !Difference between inner and outer
    radius                  (Original = 0.035)
    R_in = 0.067            !inner radius around point of
    rotation                     (Original = 0.067)
    R_out = R_in + Delta_R  !Outer radius around point of rotation

    !Geometry parameters second arm
    Alpha_2 = 91.2225               !Total angle/length of single arm
    around point of rotation  (Original = ~75)
    Beta_2 = Beta                   !Angle between arm and peak of
    tetrahedra                  (Original = ~18)
    T_2 = T/2                       !Number of tetrahedra per
    arm                               (Original = 4)
    d_2 = 0.0017                    !Shell
    thickness                                          (Original = 0.0005)
    !The 3 parameters below should currently not be changed, it displaces
    keypoints and prevents the arms from merging into 1 structure
    Delta_R_2 = Delta_R             !Difference between inner and outer
    radius                  (Original = 0.035)
    R_in_2 = R_in                   !inner radius around point of
```

```
        rotation                        (Original = 0.067)
48      R_out_2 = R_in_2 + Delta_R_2    !Outer radius around point of rotation
49
50
51      !Spring parameters
52      k = 105.9               !spring stiffness
53      L0 =  0.000000000001    !force free length of the spring
54      x0 = -0.3738            !attachment location of the spring
55
56      !Simulation parameters
57      Sim_type = 0        !If 0, it does not loop. If 1, it only loops the
        rotation to reach all locations. If 2, it loops the parameter variation
        as well.
58      Gravity = 0         !If 0, gravity is nonexistent. If 1, gravity is on.
59      Spring = 0          !If 0, spring is not included. If 1, it is included.
60      Disp_angle = 30     !Total one-sided displacement angle
61      Disp_step  = 10     !Displacement step (=< Disp_angle &
        Disp_angle/Disp_step = integer)
62
63      !Sim_type 0 only
64      ROTATE_Z = 0        !Z Rotation in deg (Sim_type 0 only)
65      ROTATE_Y = 30       !Y Rotation in deg (Sim_type 0 only)
66
67      !Data storage parameters
68      Array_length = 2*Disp_angle/Disp_step+1
69      *DIM, Energy_curve, ARRAY, Array_length, Array_length   !Creates table
        to store final energy data in
70      *DIM, M_Z_plane, ARRAY, Array_length, Array_length
71
72  !!!!    START
73
74  *get,twallbefore,active,,time,wall  !saves the wall time before solution
75
76  Variation = 1
77  Par_name = 'test'
78
79  /PREP7
80
81  !!!!    GEOMETRY
82
83      CSYS, 2     !Activates spherical coordinates (Radius, Rot_Z_deg,
        Rot_Y_deg)
84
85      !FIRST ARM
86
87      !Tetrahedron keypoints
88      K,  1,  R_in + Delta_R,     0,              0
89      K,  2,  R_in + Delta_R,     Alpha/(T),      0
90      K,  3,  R_in + Delta_R,     1/2*Alpha/(T),  Beta
91      K,  4,  R_in,               Alpha/(T),      0
92      K,  5,  R_in,               0,              0
93      K,  6,  R_in,               1/2*Alpha/(T),  Beta
94
95      !Surfaces of a single tetrahedron
96      A, 1, 5, 6, 3
97      A, 2, 4, 6, 3
98      A, 1, 2, 4, 5
99
100     !Generating tetrahedra
101     AGEN, T, 1, 3, , 0, Alpha/T, 0
```

```
102
103     *GET, Max_area_arm_1, AREA, , NUM, MAX      !Maximum areanumber in
        first arm
104
105     !SECOND ARM
106
107     CLOCAL, 11, 2, 0, 0, 0, Alpha, Gamma,  0
108
109     !Tetrahedron keypoints
110     K,  101,    R_in_2 + Delta_R_2,     0,                  0
111     K,  102,    R_in_2 + Delta_R_2,     -Alpha_2/(T_2),     0
112     K,  103,    R_in_2 + Delta_R_2,     -1/2*Alpha_2/(T_2), -Beta_2
113     K,  104,    R_in_2,                 -Alpha_2/(T_2),     0
114     K,  105,    R_in_2,                 0,                  0
115     K,  106,    R_in_2,                 -1/2*Alpha_2/(T_2), -Beta_2
116
117     !Surfaces of a single tetrahedron
118     ASEL, NONE                          !Deselecting previous area for area
        generation below
119
120     A, 101, 105, 106, 103
121     A, 102, 104, 106, 103
122     A, 101, 102, 104, 105
123
124     !Generating tetrahedra
125     AGEN, T_2, ALL, , , 0, -Alpha_2/T_2, 0
126
127     *GET, Max_area_arm_2, AREA, , NUM, MAX      !Maximum areanumber in
        second arm (and thus total joint)
128
129     ASEL, ALL
130
131     *GET, Max_Lines, LINE, 0, count                     !Shows total number
        of lines
132
133     !Coordinate systems
134     CSYS, 0
135     CLOCAL, 14, 0, 0, 0, 0, Alpha, Gamma,  0
136     CLOCAL, 15, 2, 0, 0, 0, -ALpha_2, 0, 0              !Aligned with
        undeformed end point of joint (Spherical)
137     CLOCAL, 16, 0, 0, 0, 0, 0, 0, 0                     !Aligned with
        undeformed end point of joint (Cartesian)
138
139     LSEL, ALL
140     KSEL, ALL
141     ASEL, ALL
142
143     NUMMRG, ALL                                         !Attach all geometry
144
145 !!!!    ELEMENTS & MESHING
146
147     !SHELL ARM 1
148     ET, 1, SHELL181             !Shell element type for entire real joint
149     MP, DENS,   1,  Density     !Defines a linear material property as a
        constant or a function of temperature.
150     MP, EX,     1,  Emodulus
151     MP, NUXY,   1,  Poisson
152     SECTYPE, 1, SHELL           !Section definition to shell
153     SECDATA, d, 1, 0, 3         !Shell specs
154
```

```
155     ASEL, S, AREA, , 1, Max_area_arm_1
156     AATT, 1, , 1, 16, 1          !Linking material to areas, NOTE: Pay
        attention to the associated coordinate system
157     AMESH, ALL                   !Meshing all selected areas
158
159     !SHELL ARM 2
160     ET, 4, SHELL181              !Shell element type for entire real joint
161     MP, DENS,   4,  Density      !Defines a linear material property as a
        constant or a function of temperature.
162     MP, EX,     4,  Emodulus
163     MP, NUXY,   4,  Poisson
164     SECTYPE, 4, SHELL            !Section definition to shell
165     SECDATA, d_2, 4, 0, 3        !Shell specs
166
167     ASEL, S, AREA, , Max_area_arm_1+1, Max_area_arm_2
168     AATT, 4, , 4, 16, 4          !Linking material to areas, NOTE: Pay
        attention to the associated coordinate system
169     AMESH, ALL                   !Meshing all selected areas
170
171     ASEL, ALL
172
173     !REVOLUTE DOF OF END-EFFECTOR
174     ET, 5, MPC184, 6, , , 0      !Rigid constraint element
175     MP, DENS,   5,  0            !Defines a linear material property as a
        constant or a function of temperature.
176     ESYS, 16
177     MAT, 5
178     TYPE, 5
179     SECNUM, 5
180     SECTYPE, 5, JOINT, REVO
181     SECJOINT, LSYS, 16, 16
182
183     LSEL,   S,  LINE,   ,   Max_Lines-4,    ,   , 1     !Select relevant
        line and its nodes etc. for endpoint nodes
184     *VGET, N_tip, NODE, NoNodes , NLIST, ,,, 4          !Shows selected
        endpoint node numbers
185     *GET, Max_N_tip, NODE, 0, COUNT                     !Shows the maximum
        number of endpoint nodes
186
187     N, 5000, R_in, 0, 0, 0, 0, 0
188     N, 5001, R_in + Delta_R, 0, 0, 0, 0, 0
189     E, N_tip(2), 5000
190     E, N_tip(1), 5001
191
192     !RIGID BEAMS CONSTRAINT
193     ET, 3, MPC184, 1, 1     !Rigid constraint element
194     MP, DENS,   3,  0       !Defines a linear material property as a
        constant or a function of temperature.
195     ESYS, 16
196     MAT, 3
197     TYPE, 3
198
199     N, 10000, 0, 0, 0, 0, 0 ,0  !CoR node
200     N_CoR = 10000
201
202     E, N_CoR, 5000
203     E, N_CoR, 5001
204
205     *GET, Rigid_elems_max, ELEM, 0, NUM, MAX    !Shows the maximum number
        of endpoint nodes
```

```
206       Rigid_elem1 = Rigid_elems_max - 3
207       Rigid_elem2 = Rigid_elems_max - 2
208       Rigid_elem3 = Rigid_elems_max - 1
209       Rigid_elem4 = Rigid_elems_max
210
211       *IF, Spring, EQ, 1, THEN
212       !SPRING
213       ET, 2, COMBIN14, , , 0
214       MP, DENS,   2,  0       !Defines a linear material property as a
          constant or a function of temperature.
215       R, 2, k, 0, 0, , , L0,
216       REAL, 2
217       MAT, 2
218       TYPE, 2
219
220       N, 10001, x0, 0, 0, 0, 0, 0
221       N_spring = 10001
222
223       E, N_spring, N_tip(2)
224       *ENDIF
225
226       *GET, Max_Elements, ELEM, 0, NUM, MAXD                        !Shows
          total number of lines
227
228       CSYS, 16
229       NSEL, ALL
230       NROTAT, ALL                                                   !Rotates nodal
          coordinate system to align with active coordinate system
231
232       LSEL,   S,  LINE,   ,   1,  ,   ,1                           !Select relevant
          line and its nodes etc. for support nodes, used later
233       *VGET, N_support, NODE, NoNodes , NLIST, ,,, 4                !Shows selected
          support node numbers
234       *GET, Max_N_support, NODE, 0, COUNT                           !Shows the maximum
          number of support nodes
235
236       LSEL, ALL
237       ESEL, ALL
238       KSEL, ALL
239       NSEL, ALL
240
241   !!!! SET ROTATION VARIABLES
242
243       ROT_abs = sqrt(ROTATE_Z**2 + ROTATE_Y**2)       !Absolute value of the
          rotation step
244
245       *IF, ROTATE_Y, NE, 0, THEN                      !Defines rotation angle
          around X from which to actuate the rotation step, accurate in most cases
246           ROT_angle = atan(ROTATE_Z/ROTATE_Y)*180/Pi
247       *ELSEIF, ROTATE_Y, EQ, 0, AND, ROTATE_Z, GT, 0, THEN    !Special case
          where rotate_z is not different if positive or negative, this adjusts
          for that
248           ROT_angle = 90
249       *ELSEIF, ROTATE_Y, EQ, 0, AND, ROTATE_Z, LT, 0, THEN    !Special case
          where rotate_z is not different if positive or negative, this adjusts
          for that
250           ROT_angle = -90
251       *ENDIF
252
253       *IF, ROTATE_Y, LT, 0, THEN                      !Due to the above
```

```
        definitions, negative values for Y are mirrored onto positive plane.
        Here it is mirrored back to achieve deformation in negative y direction
254         ROT_angle = ROT_angle + 180
255     *ENDIF
256
257     CSYS, 16
258     CLOCAL, 17, 0, 0, 0, 0, 0, ROT_angle, 0         !Rotate by the rotation
        angle
259     NROTAT, ALL                                     !Change nodal CS as
        well
260
261 FINISH
262
263 /SOLU
264
265     antype, STATIC, NEW     !static analysis
266     autots,ON
267     nlgeom,ON
268     pstres,ON
269     arclen,OFF
270     PRED,ON
271
272     OUTRES,ALL,ALL
273
274     NSUBST, 15, 20, 10      !Manually set substep numbers if convergence
        problems arise
275
276 !!!!    CONSTRAINTS & DISPLACEMENT
277
278     !Apply constraints on the support nodes and spring node
279     D, N_support(1), UX, , , N_support(Max_N_support), , UY, UZ, ROTX,
        ROTY, ROTZ
280
281     *IF, Spring, EQ, 1, THEN
282         D, N_spring, UX, , , , , UY, UZ
283     *ENDIf
284
285     !Apply gravity
286     *IF, Gravity, EQ, 1, THEN
287         ACEL, 0, 0, -9.81       !In global frame, because that is the only
            option
288     *ENDIF
289
290     *IF, Sim_type, EQ, 0, THEN
291
292         ESEL, ALL
293
294         !Apply constraints on the CoR node
295         D, N_CoR, ROTY, , , , , ROTX    !This prevents the N_CoR node from
            rotating in unwanted directions.
296
297         !Apply displacement to the CoR node
298         D, N_CoR, ROTZ, ROT_abs*Pi/180
299
300         SOLVE
301
302         /POST1
303
304         !Retrieve total energy data
305         ESEL, U, ELEM, , Rigid_elem1, Rigid_elem4
```

```
306         SET, LAST                    !Sets the database from which it pulls
            data
307         ETABLE, SENE, SENE           !Retrieve strain energy per element
308         SSUM                         !Sum strain energy from ETABLE
309         *GET, Energy_point, SSUM, 0 , ITEM, SENE !Gets energy data
310
311         !Exporting sim 0 data
312         /CWD, Dir_Data(1)                                  !Change
            directory to Data
313         *CFOPEN, Energy_point_%ROTATE_Z%_%ROTATE_Y%, csv    !Sets the
            vwrite output file from here on
314         *VWRITE, Energy_point
315         %G
316         *CFCLOS
317         /CWD, Dir_Log(1)
318
319         /POST26
320
321         /CWD,   Dir_Data(1)                              !Change directory
            to Data
322
323         *CFOPEN, M&F_point, csv !Sets the vwrite output file from here on
324
325         /CWD,   Dir_Log(1)                               !Change directory
            to Data
326
327         RFORCE, 2, N_CoR, M, X
328         RFORCE, 3, N_CoR, M, Y
329         RFORCE, 4, N_CoR, M, Z
330         RFORCE, 5, N_CoR, F, X
331         RFORCE, 6, N_CoR, F, Y
332         RFORCE, 7, N_CoR, F, Z
333
334         STORE, MERGE
335
336         VGET, M_X, 2
337         VGET, M_Y, 3
338         VGET, M_Z, 4
339         VGET, F_X, 5
340         VGET, F_Y, 6
341         VGET, F_Z, 7
342
343         *GET, M_Size, PARM, M_Z, DIM, X
344
345         *VWRITE, M_X(M_Size,1), M_Y(M_Size,1), M_Z(M_Size,1),
            F_X(M_Size,1), F_Y(M_Size,1), F_Z(M_Size,1)
346         %G, %G, %G, %G, %G, %G
347
348         *CFCLOS
349
350         FINISH
351
352   *ELSEIF, Sim_type, NE, 0, THEN
353
354       j = 1
355       *DO, ROTATE_Z, -Disp_angle, Disp_angle, Disp_step       !Loop that
          rotates in Z direction to multiple distances
356           *DO, ROTATE_Y, -Disp_angle, Disp_angle, Disp_step       !Loop that
              rotates in Y direction to multiple distances
357
```

```
ESEL, ALL

/PREP7                                              !Needs to go back to PREP7 commands

ROT_abs = sqrt(ROTATE_Z**2 + ROTATE_Y**2)           !Absolute value of the rotation step

*IF, ROTATE_Y, NE, 0, THEN                          !Defines rotation angle around X from which to actuate the rotation step, accurate in most cases
    ROT_angle = atan(ROTATE_Z/ROTATE_Y)*180/Pi
*ELSEIF, ROTATE_Y, EQ, 0, AND, ROTATE_Z, GT, 0, THEN
!Special case where rotate_z is not different if positive or negative, this adjusts for that
    ROT_angle = 90
*ELSEIF, ROTATE_Y, EQ, 0, AND, ROTATE_Z, LT, 0, THEN
!Special case where rotate_z is not different if positive or negative, this adjusts for that
    ROT_angle = -90
*ENDIF

*IF, ROTATE_Y, LT, 0, THEN                          !Due to the above definitions, negative values for Y are mirrored onto positive plane. Here it is mirrored back to achieve deformation in negative y direction
    ROT_angle = ROT_angle + 180
*ENDIF

CSYS, 16
CLOCAL, 17, 0, 0, 0, 0, 0, ROT_angle, 0             !Rotate by the rotation angle
NROTAT, ALL                                         !Change nodal CS as well

/SOLU                                               !Because it needs to go back to solution commands here

!Apply constraints on the CoR node
D, N_CoR, ROTY, , , , , ROTX                        !This prevents the N_CoR node from rotating in unwanted directions.

!Apply displacement to the CoR node
D, N_CoR, ROTZ, ROT_abs*Pi/180

SOLVE

/POST1

!Retrieve total energy data
ESEL, U, ELEM, , Rigid_elem1, Rigid_elem4
SET, LAST                   !Sets the database from which it pulls data
ETABLE, SENE, SENE          !Retrieve strain energy per element
SSUM                        !Sum strain energy from ETABLE
*GET, Energy_curve((-ROTATE_Y+Disp_angle)/Disp_step+1, (ROTATE_Z+Disp_angle)/Disp_step+1), SSUM, 0 , ITEM, SENE
!Gets energy data and stores it in 'Energy_curve' at the right location (Top and right taken as positive)

```

```
399             /POST26
400
401             RFORCE, 4, N_CoR, M, Z
402
403             STORE, MERGE
404
405             VGET, M_Z_%j%, 4
406
407             *GET, M_Size, PARM, M_Z_%j%, DIM, X
408
409             M_Z_plane((-ROTATE_Y+Disp_angle)/Disp_step+1,
                (ROTATE_Z+Disp_angle)/Disp_step+1) = M_Z_%j%(M_Size,1)
410
411             *IF, ROTATE_Z, EQ, -Disp_angle, AND, ROTATE_Y, EQ, -Disp_angle,
                THEN    !Saves an image only in this specific configuration for
                visual confirmation (best view)
412
413             /POST1
414                 !View settings
415                 *Get, View_X, NODE, N_tip(1), LOC, X
416                 *Get, View_Y, NODE, N_tip(1), LOC, Y
417                 *Get, View_Z, NODE, N_tip(1), LOC, Z
418                 /VIEW, All, View_X, View_Y, View_Z
419                 /DIST, ALL, 0.15, 1
420                 /DSCALE, ALL ,1.0
421                 /ESHAPE, 1
422                 DSYS, 16
423
424                 /CWD, Dir_Input(1)
425                 *USE, Def_path
426
427                 /CWD, Dir_Data(1)
428                 /IMAGE, SAVE, Energy_%Par_name%_%Variation%     !Saves an
                    image
429                 /CWD, Dir_Log(1)
430
431             *ENDIF
432             j = j + 1
433         *ENDDO
434     *ENDDO
435
436
437 !!!!    SAVING RESULT
438
439     !Save to file
440     /CWD,   Dir_Data(1)                                 !Change directory to
        Data
441     *CFOPEN, Energy_%Par_name%_%Variation%, csv !Sets the vwrite output
        file from here on
442
443     !Simply exporting the energy data...
444     *IF, Array_length, EQ, 3, THEN
445         *VWRITE, Energy_curve(1,1), Energy_curve(1,2), Energy_curve(1,3)
446         %G, %G, %G
447     *ELSEIF, Array_length, EQ, 5, THEN
448         *VWRITE, Energy_curve(1,1), Energy_curve(1,2), Energy_curve(1,3),
            Energy_curve(1,4), Energy_curve(1,5)
449         %G, %G, %G, %G, %G
450     *ELSEIF, Array_length, EQ, 7, THEN
451         *VWRITE, Energy_curve(1,1), Energy_curve(1,2), Energy_curve(1,3),
```

```
            Energy_curve(1,4), Energy_curve(1,5), Energy_curve(1,6),
            Energy_curve(1,7)
452         %G, %G, %G, %G, %G, %G, %G
453     *ELSEIF, Array_length, EQ, 9, THEN
454         *VWRITE, Energy_curve(1,1), Energy_curve(1,2), Energy_curve(1,3),
            Energy_curve(1,4), Energy_curve(1,5), Energy_curve(1,6),
            Energy_curve(1,7), Energy_curve(1,8), Energy_curve(1,9)
455         %G, %G, %G, %G, %G, %G, %G, %G, %G,
456     *ELSEIF, Array_length, EQ, 11, THEN
457         *VWRITE, Energy_curve(1,1), Energy_curve(1,2), Energy_curve(1,3),
            Energy_curve(1,4), Energy_curve(1,5), Energy_curve(1,6),
            Energy_curve(1,7), Energy_curve(1,8), Energy_curve(1,9),
            Energy_curve(1,10), Energy_curve(1,11)
458         %G, %G, %G, %G, %G, %G, %G, %G, %G, %G, %G
459     *ELSEIF, Array_length, EQ, 13, THEN
460         *VWRITE, Energy_curve(1,1), Energy_curve(1,2), Energy_curve(1,3),
            Energy_curve(1,4), Energy_curve(1,5), Energy_curve(1,6),
            Energy_curve(1,7), Energy_curve(1,8), Energy_curve(1,9),
            Energy_curve(1,10), Energy_curve(1,11), Energy_curve(1,12),
            Energy_curve(1,13)
461         %G, %G, %G, %G, %G, %G, %G, %G, %G, %G, %G, %G, %G
462     *ENDIF
463
464     *CFCLOS
465
466     *CFOPEN, M_Z_plane, csv !Sets the vwrite output file from here on
467
468     *IF, Array_length, EQ, 3, THEN
469         *VWRITE, M_Z_plane(1,1), M_Z_plane(1,2), M_Z_plane(1,3)
470         %G, %G, %G
471     *ELSEIF, Array_length, EQ, 5, THEN
472         *VWRITE, M_Z_plane(1,1), M_Z_plane(1,2), M_Z_plane(1,3),
            M_Z_plane(1,4), M_Z_plane(1,5)
473         %G, %G, %G, %G, %G
474     *ELSEIF, Array_length, EQ, 7, THEN
475         *VWRITE, M_Z_plane(1,1), M_Z_plane(1,2), M_Z_plane(1,3),
            M_Z_plane(1,4), M_Z_plane(1,5), M_Z_plane(1,6), M_Z_plane(1,7)
476         %G, %G, %G, %G, %G, %G, %G
477     *ELSEIF, Array_length, EQ, 9, THEN
478         *VWRITE, M_Z_plane(1,1), M_Z_plane(1,2), M_Z_plane(1,3),
            M_Z_plane(1,4), M_Z_plane(1,5), M_Z_plane(1,6), M_Z_plane(1,7),
            M_Z_plane(1,8), M_Z_plane(1,9)
479         %G, %G, %G, %G, %G, %G, %G, %G, %G,
480     *ELSEIF, Array_length, EQ, 11, THEN
481         *VWRITE, M_Z_plane(1,1), M_Z_plane(1,2), M_Z_plane(1,3),
            M_Z_plane(1,4), M_Z_plane(1,5), M_Z_plane(1,6), M_Z_plane(1,7),
            M_Z_plane(1,8), M_Z_plane(1,9), M_Z_plane(1,10), M_Z_plane(1,11)
482         %G, %G, %G, %G, %G, %G, %G, %G, %G, %G, %G
483     *ELSEIF, Array_length, EQ, 13, THEN
484         *VWRITE, M_Z_plane(1,1), M_Z_plane(1,2), M_Z_plane(1,3),
            M_Z_plane(1,4), M_Z_plane(1,5), M_Z_plane(1,6), M_Z_plane(1,7),
            M_Z_plane(1,8), M_Z_plane(1,9), M_Z_plane(1,10), M_Z_plane(1,11),
            M_Z_plane(1,12), M_Z_plane(1,13)
485         %G, %G, %G, %G, %G, %G, %G, %G, %G, %G, %G, %G, %G
486     *ENDIF
487
488     *CFCLOS
489
490 *ENDIF
491
```

```
492       /CWD,    Dir_Log(1)          !Change directory back to log
493
494       !Defines an abbreviation, makes them appear in the tool box bar
495       *Abbr, Keypoints, kplot          ! Displays keypoints
496       *Abbr, Nodes, nplot              ! Displays nodes
497       *Abbr, Lines, lplot              ! Displays lines
498       *Abbr, Elements, eplot           ! Displays elements
499       *Abbr, Deformed, PLDISP, 2       ! Displays the displaced structure
500       *Abbr, Strain_energy, PLNSOL, SEND, ELASTIC
501       *Abbr, Stress, PLNSOL, S, EQV
502       *Abbr, Csys_global, DSYS, 0      ! Displays global cartesian coordinate
          system
503       *Abbr, Csys_actuation, DSYS, 17 ! Displays global spherical coordinate
          system
504       *Abbr, Csys_endpoint, DSYS, 16   ! Displays coordinate system aligned
          with undeformed endpoint of the joint
505
506       *get,twallafter,active,,time,wall           !saves the wall time after
          solution
507       Sim_time_h = (twallafter-twallbefore)       !Simulation time in seconds
508
509       *IF, Sim_time_h, GT, 1, THEN                !Loop that determines if
          simulation time is shown in seconds, minutes or hours
510           Sim_time = Sim_time_h
511           Time_unit = 'hours'
512       *ELSEIF, Sim_time_h, LT, 1, AND, Sim_time_h, GT, 1/60, THEN
513           Sim_time = Sim_time_h*60
514           Time_unit = 'minutes'
515       *ELSEIF, Sim_time_h, LT, 1/60, THEN
516           Sim_time = Sim_time_h*3600
517           Time_unit = 'seconds'
518       *ENDIF
519
520       *MSG, UI, %Sim_time%, Time_unit         !Displays the final simulation
          time
521       Simulation time = %G %C
522
523   FINISH
524
525   /POST1
526       !View settings
527       *Get, View_X, NODE, N_tip(1), LOC, X
528       *Get, View_Y, NODE, N_tip(1), LOC, Y
529       *Get, View_Z, NODE, N_tip(1), LOC, Z
530       /VIEW, All, View_X, View_Y, View_Z
531       /DIST, ALL, 0.15, 1
532       /DSCALE, ALL ,1.0
533       /ESHAPE, 1
534       DSYS, 16
535       PLDISP, 2
```

D.2. Joint optimization code

This section contains all files that are needed to perform the joint optimization. Before each file, a short description is given about the important principles used in the code.

D.2.1. Main optimization

The main Matlab code used in the joint optimization is shown here. Before calling the optimization, it removes a number of log and startup files, if they exist. After that, they are recreated as empty folders. This action is performed in order to prevent ANSYS run errors during the optimization. The objective function is called with a $fminsearch$ command.

```
clear all
close all
clc
%% Info

% REQUIRED MANUAL ACTIONS
    %1. OPEN MATLAB AS ADMINISTRATOR?
    %2. CONNECT TO TU DELFT VPN
    %3. SET DESIRED DISP_ANGLE AND DISP_STEP IN THE ANSYS INPUT FILE
    %4. SELECT DESIRED X0 AND MATCH NUMBER OF VARIABLES IN THE OBJ.M FILE

% Fundamental steps of this optimization
    %1. Run ANSYS with specific parameters
    %2. Import ANSYS iteration results to this matlab script
    %3. Analyze ANSYS data and adjust for next optimization iteration
    %4. Send new parameters to ANSYS
    %5. Repeat until optimization has completed

%% Clean up the folders from earlier runs

if exist('Logs')
    rmdir('Logs', 's');
    mkdir('Logs')
end

if exist('Startup')
     rmdir('Startup', 's');
     mkdir('Startup')
end

copyfile run_ansys.bat Startup

%% Optimization

%Starting parameters
x0 = [0.7*75; 0.9*70; 0.9*75; 0.0005*1.5];      %[Alpha; Gamma; Alpha_2; d_2] approximate best location according to initial correlation figures

options = optimset('Display','iter');

tic
[x,fval] = fminsearch(@obj, x0, options);
toc

%% Result display

disp("Starting parameters: ")
disp(x0')

disp("Optimized parameters: ")
disp(x')

disp("fval = " + fval)
```

D.2.2. Objective function

After being called by the main optimization code, the file starts by setting the optimization variables to those of the current iteration, determined by $fminsearch$. With these values, an ANSYS input file is created that delivers the new parameter values to ANSYS. Furthermore, a messenger file is created just before ANSYS is called. ANSYS can use this messenger file to communicate to Matlab that it has completed its run. The file that actually calls ANSYS in batch mode is ran after which the code waits. This file contains the following:

```
SET ANS_CONSEC=YES
SET ANSYS_LOCK=OFF
SET KMP_STACKSIZE=2048k
"C:\Program Files\ANSYS Inc\v212\ANSYS\bin\winx64\MAPDL.exe" -b -i "C:\Users\
Desktop\Thesis\ANSYS_APDL\Optimization\Spherical_model_optimization.ans" -o
"C:\Users\Desktop\Thesis\ANSYS_APDL\Optimization\Startup\file.out"
```

When ANSYS has completed this run, the messenger file is deleted and Matlab is signaled to continue the optimization. Finally, the new energy field file of the latest run is retrieved and analysed according to the axi-symmetrical energy field metric that forms the new objective function value.

```
function f = obj(x)

%% Export parameters to ANSYS

% MANUALLY MATCH PARAMS BELOW TO THE OPTIMIZATION PARAMETERS
% Geometry parameters
PARAM.Alpha = x(1);     %Sets the geometry parameters to match parameters chosen by fminsearch for next iteration
PARAM.Gamma = x(2);
PARAM.Alpha_2 = x(3);
PARAM.d_2 = x(4);

%Export new params to matlab

fid = fopen('ANSYS_input.txt','wt');
%Create ansys parameter input file
fprintf(fid,  'Alpha = %d \n',        PARAM.Alpha       );
fprintf(fid,  'Gamma = %d \n',        PARAM.Gamma       );
fprintf(fid,  'Alpha_2 = %d \n',      PARAM.Alpha_2     );
fprintf(fid,  'd_2 = %d \n',          PARAM.d_2         );
fclose(fid);

% Create a file that ansys deletes once it is done running
fid2 = fopen('Matlab_ready.txt','wt');
fprintf(fid2,  'Matlab_ready = 0');
fclose(fid2);

%% Start Ansys run
! run_ansys.bat > NUL

%% Waiting for ANSYS to finish run

%While the matlab_ready file exists, wait. ANSYS deletes the file when done with iteration.
count = 0;

while isfile('Matlab_ready.txt')
    pause(1)
     count = 1 + count;

    %If the counter reaches a certain timespan, it is likely something went wrong. If it is a temporary license connection error, a second request to run ansys will continue the loop
    if count == 300
        cd 'C:\Users\Dion-\Desktop\Thesis\ANSYS_APDL\Optimization\Startup'
        ! run_ansys.bat > NUL
        disp("Backup ANSYS run requested")
        count = 0;
    end
end

%% Processing new iteration data into metric scalar data

Energy = readmatrix("Energy_file.csv");         %Read energy field of current run
E_loc_all = [1, sqrt(2), 2, sqrt(5), sqrt(8), 3]; %All E_loc possibilities within manageable range (- & + 3 data points)
Neut_pos = length(Energy)/2 + 0.5;  %Neutral position of the joint, selected in the datamatrix

```

```
53          for m = 1:(Neut_pos-1)*(Neut_pos)/2
54
55              E_loc = E_loc_all(m); %Matrix distance of selected datapoints
                from neutral position. Calculated by pythagoras theorem, so can
                be integer or a square root
56              clear("E_point") %Removes E_point from previous loop
57              l = 1;  %Loop variable, is reset here
58
59              for j = [-length(Energy)/2 + 0.5: length(Energy)/2 - 0.5]
                %Loop that selects every datapoint of 1 axis
60                  for k = [-length(Energy)/2 + 0.5: length(Energy)/2 - 0.5]
                    %Loop that selects every datapoint of the other axis
61                      if  sqrt(k^2 + j^2) == E_loc   %Checks if the selected
                        points correspond to selected location
62
63                          E_point(l) = Energy(Neut_pos + j, Neut_pos + k);
                            %Saves energy of that point
64                          l = l + 1;  %updates location in E_point
65
66                      end
67                  end
68              end
69
70              E_average = sum(E_point)/length(E_point);    %Averages the
                energy of selected points
71              E_RMS(m) = sqrt(sum((E_average - E_point).^2)/length(E_point));
                %RMS error of the points relative to the average per E_loc
72
73          end
74
75  E_RMS_total = sum(E_RMS);   %Sum of all E_loc RMS errors for the specific
    variation
76  E_RMS_scaled = E_RMS_total/(sum(sum(Energy))/length(Energy)^2); %Scaled
    version of RMS by average total energy to account for magnitude differences
    without shape differences
77
78  f = E_RMS_scaled;
79
80  end
```

D.2.3. ANSYS APDL optimization

The ANSYS APDL optimization file is similar to the ANSYS APDL model that has been shown previously. Since this file is only used in the optimization, all non-essential code sections have been removed. Moreover, parameter values of the optimization variables are imported from matlab before each run. Furthermore, an adaptation has been made that uses the messenger file to signal Matlab that it has completed its run.

```
!   Dion Hogervorst

!!!!    GENERAL INFO

    /CLEAR, NOSTART

    *DIM, Dir_Log, STRING, 200
    *DIM, Dir_Data, STRING, 200
    *DIM, Dir_Input, STRING, 200

    Dir_Log(1) = JOIN(Dir(1), 'Logs')
    Dir_Data(1) = JOIN(Dir(1), 'ANSYS_DATA')
    Dir_Input(1) = JOIN(Dir(1), 'ANSYS_INPUT')

    /CWD, Dir_Log(1)

FINISH

    /title,  Spherical_model_optimization
    /FILNAME, Spherical_model_optimization, 1
    /CONFIG, NRES, 1000000
    /units, SI

!!!!    PARAMETERS

    Pi = ACOS(-1)       !pi, 3.14...

    !Material parameters
    !Performance grade Titanium alloy (Ti6Al4V)
    Density = 4390
    Emodulus = 110e9

    !Geometry parameters first arm
    Alpha = 75              !Total angle/length of single arm around point of rotation  (Original = ~75)
    Beta = 18               !Angle between arm and peak of tetrahedra                   (Original = ~18)
    Gamma = 70              !Angle between the two arms                                 (Original = ~70)
    T = 4                   !Number of tetrahedra per arm                               (Original = 4)
    d = 0.0005              !Shell thickness                                            (Original = 0.0005)
    Delta_R = 0.035         !Difference between inner and outer radius                  (Original = 0.035)
    R_in = 0.067            !inner radius around point of rotation                      (Original = 0.067)
    R_out = R_in + Delta_R  !Outer radius around point of rotation

    !Geometry parameters second arm
    Alpha_2 = Alpha                 !Total angle/length of single arm around point of rotation  (Original = ~75)
    Beta_2 = Beta                   !Angle between arm and peak of tetrahedra                   (Original = ~18)
    T_2 = T                         !Number of tetrahedra per arm                               (Original = 4)
    d_2 = d                         !Shell thickness                                            (Original = 0.0005)
    !The 3 parameters below should currently not be changed, it displaces keypoints and prevents the arms from merging into 1 structure
```

```
49      Delta_R_2 = Delta_R             !Difference between inner and outer
        radius                    (Original = 0.035)
50      R_in_2 = R_in                   !inner radius around point of
        rotation                        (Original = 0.067)
51      R_out_2 = R_in_2 + Delta_R_2    !Outer radius around point of rotation
52
53      !Simulation parameters
54      Disp_angle = 30     !Total one-sided displacement angle
55      Disp_step  = 10     !Displacement step (=< Disp_angle &
        Disp_angle/Disp_step = integer)
56
57      !Data storage parameters
58      Array_length = 2*Disp_angle/Disp_step+1
59      *DIM, Energy_curve, ARRAY, Array_length, Array_length   !Creates table
        to store final energy data in
60
61      !Overwrite by external variables
62      /INPUT,'ANSYS_input','txt', Dir_Input(1)
63
64  !!!!    START
65
66  /PREP7
67
68  !!!!    GEOMETRY
69
70      CSYS, 2     !Activates spherical coordinates (Radius, Rot_Z_deg,
        Rot_Y_deg)
71
72      !FIRST ARM
73
74      !Tetrahedron keypoints
75      K,  1,  R_in + Delta_R,     0,              0
76      K,  2,  R_in + Delta_R,     Alpha/(T),      0
77      K,  3,  R_in + Delta_R,     1/2*Alpha/(T),  Beta
78      K,  4,  R_in,               Alpha/(T),      0
79      K,  5,  R_in,               0,              0
80      K,  6,  R_in,               1/2*Alpha/(T),  Beta
81
82      !Surfaces of a single tetrahedron
83      A, 1, 5, 6, 3
84      A, 2, 4, 6, 3
85      A, 1, 2, 4, 5
86
87      !Generating tetrahedra
88      AGEN, T, 1, 3, , 0, Alpha/T, 0
89
90      *GET, Max_area_arm_1, AREA, , NUM, MAX      !Maximum areanumber in
        first arm
91
92      !SECOND ARM
93
94      CLOCAL, 11, 2, 0, 0, 0, Alpha, Gamma,  0
95
96      !Tetrahedron keypoints
97      K,  101,    R_in_2 + Delta_R_2,     0,                  0
98      K,  102,    R_in_2 + Delta_R_2,     -Alpha_2/(T_2),     0
99      K,  103,    R_in_2 + Delta_R_2,     -1/2*Alpha_2/(T_2), -Beta_2
100     K,  104,    R_in_2,                 -Alpha_2/(T_2),     0
101     K,  105,    R_in_2,                 0,                  0
102     K,  106,    R_in_2,                 -1/2*Alpha_2/(T_2), -Beta_2
```

```
103
104       !Surfaces of a single tetrahedron
105       ASEL, NONE                          !Deselecting previous area for area
          generation below
106
107       A, 101, 105, 106, 103
108       A, 102, 104, 106, 103
109       A, 101, 102, 104, 105
110
111       !Generating tetrahedra
112       AGEN, T_2, ALL, , , 0, -Alpha_2/T_2, 0
113
114       *GET, Max_area_arm_2, AREA, , NUM, MAX      !Maximum areanumber in
          second arm (and thus total joint)
115
116       ASEL, ALL
117
118       *GET, Max_Lines, LINE, 0, count                    !Shows total number
          of lines
119
120       !Coordinate systems
121       CSYS, 0
122       CLOCAL, 14, 0, 0, 0, 0, Alpha, Gamma,  0
123       CLOCAL, 15, 2, 0, 0, 0, -ALpha_2, 0, 0             !Aligned with
          undeformed end point of joint (Spherical)
124       CLOCAL, 16, 0, 0, 0, 0, 0, 0, 0                    !Aligned with
          undeformed end point of joint (Cartesian)
125
126       LSEL, ALL
127       KSEL, ALL
128       ASEL, ALL
129
130       NUMMRG, ALL                                        !Attach all geometry
131
132   !!!!    ELEMENTS & MESHING
133
134       !SHELL ARM 1
135       ET, 1, SHELL181             !Shell element type for entire real joint
136       MP, DENS,   1,  Density     !Defines a linear material property as a
          constant or a function of temperature.
137       MP, EX,     1,  Emodulus
138       MP, NUXY,   1,  Poisson
139       SECTYPE, 1, SHELL           !Section definition to shell
140       SECDATA, d, 1, 0, 3         !Shell specs
141
142       ASEL, S, AREA, , 1, Max_area_arm_1
143       AATT, 1, , 1, 16, 1         !Linking material to areas, NOTE: Pay
          attention to the associated coordinate system
144       AMESH, ALL                  !Meshing all selected areas
145
146       !SHELL ARM 2
147       ET, 4, SHELL181             !Shell element type for entire real joint
148       MP, DENS,   4,  Density     !Defines a linear material property as a
          constant or a function of temperature.
149       MP, EX,     4,  Emodulus
150       MP, NUXY,   4,  Poisson
151       SECTYPE, 4, SHELL           !Section definition to shell
152       SECDATA, d_2, 4, 0, 3           !Shell specs
153
154       ASEL, S, AREA, , Max_area_arm_1+1, Max_area_arm_2
```

```
155     AATT, 4, , 4, 16, 4          !Linking material to areas, NOTE: Pay
        attention to the associated coordinate system
156     AMESH, ALL                   !Meshing all selected areas
157
158     ASEL, ALL
159
160     !REVOLUTE DOF OF JOINT TIP
161     ET, 5, MPC184, 6, , , 0          !Rigid constraint element
162     MP, DENS,   5,  0                !Defines a linear material property as
        a constant or a function of temperature.
163     ESYS, 16
164     MAT, 5
165     TYPE, 5
166     SECNUM, 5
167     SECTYPE, 5, JOINT, REVO
168     SECJOINT, LSYS, 16, 16
169
170     LSEL,   S,  LINE,   ,   Max_Lines-4,    ,   , 1      !Select relevant
        line and its nodes etc. for endpoint nodes
171     *VGET, N_tip, NODE, NoNodes , NLIST, ,,, 4           !Shows selected
        endpoint node numbers
172     *GET, Max_N_tip, NODE, 0, COUNT                      !Shows the maximum
        number of endpoint nodes
173
174     N, 5000, R_in, 0, 0, 0, 0, 0
175     N, 5001, R_in + Delta_R, 0, 0, 0, 0, 0
176     E, N_tip(2), 5000
177     E, N_tip(1), 5001
178
179     !RIGID BEAMS CONSTRAINT
180     ET, 3, MPC184, 1, 1      !Rigid constraint element
181     MP, DENS,   3,  0        !Defines a linear material property as a
        constant or a function of temperature.
182     ESYS, 16
183     MAT, 3
184     TYPE, 3
185
186     N, 10000, 0, 0, 0, 0, 0 ,0              !CoR node
187     N_CoR = 10000
188
189     E, N_CoR, 5000
190     E, N_CoR, 5001
191
192     *GET, Rigid_elems_max, ELEM, 0, NUM, MAX    !Shows the maximum number
        of endpoint nodes
193     Rigid_elem1 = Rigid_elems_max - 3
194     Rigid_elem2 = Rigid_elems_max - 2
195     Rigid_elem3 = Rigid_elems_max - 1
196     Rigid_elem4 = Rigid_elems_max
197
198     *GET, Max_Elements, ELEM, 0, NUM, MAXD      !Shows total number of lines
199
200     CSYS, 16
201     NSEL, ALL
202     NROTAT, ALL                                      !Rotates nodal
        coordinate system to align with active coordinate system
203
204     LSEL,   S,  LINE,   ,   1,  ,   ,1                  !Select relevant
        line and its nodes etc. for support nodes, used later
205     *VGET, N_support, NODE, NoNodes , NLIST, ,,, 4       !Shows selected
```

```
        support node numbers
206     *GET, Max_N_support, NODE, 0, COUNT                  !Shows the maximum
        number of support nodes
207
208     LSEL, ALL
209     ESEL, ALL
210     KSEL, ALL
211     NSEL, ALL
212
213 FINISH
214
215 /SOLU
216
217     antype, STATIC, NEW     !static analysis
218     autots,ON
219     nlgeom,ON
220     pstres,ON
221     arclen,OFF
222     PRED,ON
223
224     OUTRES,ALL,ALL
225
226 !!!!    CONSTRAINTS & DISPLACEMENT
227
228     !Apply constraints on the support nodes
229     D, N_support(1), UX, , , N_support(Max_N_support), , UY, UZ, ROTX,
        ROTY, ROTZ
230
231     *DO, ROTATE_Z, -Disp_angle, Disp_angle, Disp_step       !Loop that
        rotates around Z to multiple distances
232         *DO, ROTATE_Y, -Disp_angle, Disp_angle, Disp_step       !Loop that
            rotates around Y to multiple distances
233
234             /PREP7                                          !Needs to go
                back to PREP7 commands
235
236             ROT_abs = sqrt(ROTATE_Z**2 + ROTATE_Y**2)       !Absolute value
                of the rotation step
237
238             *IF, ROTATE_Y, NE, 0, THEN                      !Defines
                rotation angle around X from which to actuate the rotation
                step, accurate in most cases
239                 ROT_angle = atan(ROTATE_Z/ROTATE_Y)*180/Pi
240             *ELSEIF, ROTATE_Y, EQ, 0, AND, ROTATE_Z, GT, 0, THEN
                !Special case where rotate_z is not different if positive or
                negative, this adjusts for that
241                 ROT_angle = 90
242             *ELSEIF, ROTATE_Y, EQ, 0, AND, ROTATE_Z, LT, 0, THEN
                !Special case where rotate_z is not different if positive or
                negative, this adjusts for that
243                 ROT_angle = -90
244             *ENDIF
245
246             *IF, ROTATE_Y, LT, 0, THEN                      !Due to the
                above definitions, negative values for Y are mirrored onto
                positive plane. Here it is mirrored back to achieve deformation
                in negative y direction
247                 ROT_angle = ROT_angle + 180
248             *ENDIF
249
```

```
250             CSYS, 16
251             CLOCAL, 17, 0, 0, 0, 0, 0, ROT_angle, 0         !Rotate by the
                rotation angle
252             NROTAT, ALL                                     !Change nodal
                CS as well
253
254             /SOLU                                           !Because it
                needs to go back to solution commands here
255
256             !Apply constraints on the CoR node
257             D, N_CoR, ROTY, , , , , , ROTX                  !This prevents
                the N_CoR node from rotating in unwanted directions. It leaves
                end connection between rigid beam and joint tip free since
                MPC184 has 6 DoF at EACH node
258
259             !Apply displacement to the CoR node
260             D, N_CoR, ROTZ, ROT_abs*Pi/180
261
262             SOLVE
263
264 FINISH
265
266             /POST1
267
268             !Retrieve total energy data
269             SET, LAST                   !Sets the database from which it
                pulls data
270             ETABLE, SENE, SENE          !Retrieve strain energy per element
271             SSUM                        !Sum strain energy from ETABLE
272             *GET, Energy_curve((-ROTATE_Y+Disp_angle)/Disp_step+1,
                (ROTATE_Z+Disp_angle)/Disp_step+1), SSUM, 0 , ITEM, SENE
                !Gets energy data and stores it in 'Energy_curve' at the right
                location (Top and right taken as positive)
273
274         *ENDDO
275     *ENDDO
276
277 !!!!    SAVING RESULT
278
279
280     !Save to file
281     /CWD,   Dir_Data(1)                             !Change directory to
        Data
282     *CFOPEN, Energy_file, csv   !Sets the vwrite output file from here on
283
284     !Simply exporting the energy data...
285     *IF, Array_length, EQ, 3, THEN
286         *VWRITE, Energy_curve(1,1), Energy_curve(1,2), Energy_curve(1,3)
287         %G, %G, %G
288     *ELSEIF, Array_length, EQ, 5, THEN
289         *VWRITE, Energy_curve(1,1), Energy_curve(1,2), Energy_curve(1,3),
            Energy_curve(1,4), Energy_curve(1,5)
290         %G, %G, %G, %G, %G
291     *ELSEIF, Array_length, EQ, 7, THEN
292         *VWRITE, Energy_curve(1,1), Energy_curve(1,2), Energy_curve(1,3),
            Energy_curve(1,4), Energy_curve(1,5), Energy_curve(1,6),
            Energy_curve(1,7)
293         %G, %G, %G, %G, %G, %G, %G
294     *ELSEIF, Array_length, EQ, 9, THEN
295         *VWRITE, Energy_curve(1,1), Energy_curve(1,2), Energy_curve(1,3),
```

```
            Energy_curve(1,4), Energy_curve(1,5), Energy_curve(1,6),
            Energy_curve(1,7), Energy_curve(1,8), Energy_curve(1,9)
296         %G, %G, %G, %G, %G, %G, %G, %G, %G,
297     *ELSEIF, Array_length, EQ, 11, THEN
298         *VWRITE, Energy_curve(1,1), Energy_curve(1,2), Energy_curve(1,3),
            Energy_curve(1,4), Energy_curve(1,5), Energy_curve(1,6),
            Energy_curve(1,7), Energy_curve(1,8), Energy_curve(1,9),
            Energy_curve(1,10), Energy_curve(1,11)
299         %G, %G, %G, %G, %G, %G, %G, %G, %G, %G, %G
300     *ELSEIF, Array_length, EQ, 13, THEN
301         *VWRITE, Energy_curve(1,1), Energy_curve(1,2), Energy_curve(1,3),
            Energy_curve(1,4), Energy_curve(1,5), Energy_curve(1,6),
            Energy_curve(1,7), Energy_curve(1,8), Energy_curve(1,9),
            Energy_curve(1,10), Energy_curve(1,11), Energy_curve(1,12),
            Energy_curve(1,13)
302         %G, %G, %G, %G, %G, %G, %G, %G, %G, %G, %G, %G, %G
303     *ENDIF
304
305     *CFCLOS
306
307 FINISH
308
309 !!!!        RESTART
310
311 /DELETE,
    'C:\Users\Desktop\Thesis\ANSYS_APDL\Optimization\ANSYS_INPUT\Matlab_ready',
    'txt'
```

D.3. Spring optimization code

This section contains all files that are needed to perform the spring optimization. Before each file, a short description is given about the important principles used in the code.

D.3.1. Main optimization

The main optimization is contained in this file. First, a number of parameters are set that are used in the optimization. The energy field of the optimized joint is loaded as well. $MultiStart$ is used to complete the actual optimization, with the algorithm being $interior - point$, which is called through $fmincon$. Initially, random points where chosen to be used in the $MultiStart$. However, this had a downside. The stochastic nature of this method leads to a different result every time, which prevents repeatability of results. This also indicates that the result is sensitive to the starting point and that there are many local minima. In the final version, a fixed starting point set is chosen that starts in a relatively realistic range, but is not bound within it. What is shown here is the version for the theoretical best optimization result, therefore no constraint equations are called. The main optimization file for tension and compression springs is largely similar, with the only differences being the bounds, number of variables and the addition of constraint functions.

```
clear all
close all
clc

%% File wide params

%Multistart range
ms_min = [-0.3; 0; 0];
ms_max = [0.3; 0.3; 1000];

%Lower and upper optimization bounds
lb = [-inf; 0; 0];%ms_min;
ub = [inf; inf; inf];%ms_max;

%Other params
disp_angle = 30;    %Set to match ansys disp_angle
disp_step = disp_angle/3;     %Set to match ansys disp_step

R = 0.067;

%% Obtain energy curve to compensate
E_joint = readmatrix("Energy_tita_originalsize.csv"); %Original energy
curve to compensate

%% Optimization
step = (500)^(1/length(ub)); %Determines stepsize for 500 multistarts

n = 1;
for x0_1 = ms_min(1):(ms_max(1)-ms_min(1))/step:ms_max(1)
    for x0_2 = ms_min(2):(ms_max(2)-ms_min(2))/step:ms_max(2)
        for x0_3 = ms_min(3):(ms_max(3)-ms_min(3))/step:ms_max(3)

            ptmatrix(n,:) = [x0_1, x0_2, x0_3];

            n = n + 1;
        end
    end
end

ms = MultiStart;
tpoints = CustomStartPointSet(ptmatrix);
opts = optimoptions(@fmincon,'Algorithm','interior-point');
problem =
createOptimProblem('fmincon','objective',@(x)Spring_obj_new(x,R,E_joint,disp_
angle,disp_step),'x0',[0,0,0],'lb',lb,'ub',ub,'options',opts);
[x,fval] = run(ms,problem,tpoints)
```

D.3.2. Objective function

The objective function of the spring is called in the main optimization and requires a number of inputs. The metric itself is determined by a loop that covers all locations of the energy field. For the points that are within a 30 degrees range of motion, the energy within the spring is calculated. After that, the RMSE is calculated and used as the objective function.

```
function f = Spring_obj_new(x, R, E_com, disp_angle, disp_step)
%x = optimization parameters
%R = attachment radius of spring
%E_com = to be compensated energy curve
%disp_angle = displacement angle of the simulation
%disp_step = displacement step of the simulation

%% set params
x_0 = x(1);          %X coordinate spring origin
L0 = x(2);           %Neutral spring length
k = x(3);            %Spring stiffness

m = 1;               %Loop counter
%% Mirrored energy curve metric

for n = 1:length(E_com)

    Z(n) = -disp_angle + (n-1)*disp_step;   %Z rotation in equal steps as
    number of datapoints available

    for i = 1:length(E_com)

        Y(i) = -disp_angle + (i-1)*disp_step; %Y rotation in equal steps as
        number of datapoints available

        Rot_abs(n,i) = sqrt(Z(n)^2 + Y(i)^2);       %Absolute rotation
        angle, same as defined in ansys

%       To exclude data outside of the round range of motion
        if Rot_abs(n,i) <= disp_angle

            x1(n,i) = cosd(Rot_abs(n,i))*R;    %x (vertical height)
            coordinate of the joint tip
            w1(n,i) = sind(Rot_abs(n,i))*R;    %horizontal coordinate of
            the joint tip. Can be either Z or Y, axisymmetricality makes
            them the same, hence it is called w1
            L(n,i) = sqrt((x1(n,i) - x_0)^2 + w1(n,i)^2);      %Length of
            spring


            E(n,i) = 0.5*k*(L(n,i) - L0)^2;         %Energy of general spring

            E_total(n,i) = E_com(n,i) + E(n,i);

            E_relevant(m) = E_total(n,i);
            m = m + 1;
        end

    end
end

E_relevant_avg = sum(E_relevant)/length(E_relevant);

E_relevant_RMS = sqrt(sum((E_relevant -
E_relevant_avg).^2)/length(E_relevant));

f = E_relevant_RMS/(sum(sum(E_com))/length(E_com)^2);
end
```

D.3.3. Compression constraints

This section of code shows the compression spring constraints that were used during the optimization of a realistic compression spring. The constraints are called in the main optimization file of the compression spring optimization. The definition of the constraints themselves have been discussed more extensively in the paper.

```
function [g, g_eq] = Spring_cons_comp_new(x, R)

%Params
x_0 = x(1);
L0 = x(2);
d = x(3);
D = x(4);
n = x(5);

E = 190e9;
G = E/2.6;
k = (d^4*G)/(64*n*(D/2)^3);

%Maximum spring length calc (in neutral position)
x1_center = cosd(0)*R;    %x (vertical height) coordinate of the joint tip
w1_center = sind(0)*R;    %horizontal coordinate of the joint tip. Can be either Z or Y, axisymmetricality makes them the same, hence it is called w1
L_center = sqrt((x1_center - x_0)^2 + w1_center^2);      %Length of spring

%Force and stress
F = k*(L_center - L0);
tau_eqv = 16*abs(F)*(0.5*D)/(pi*d^3) + 4*abs(F)/(pi*d^2);
sigma_ultimate = 1720e6;
F_ultimate = 308.2;

%If eqv stress is larger than half the ultimate stress.
g(1) =  tau_eqv/(sigma_ultimate/2) - 1;

%If the force is larger than 308.2N, the joint itself has the problem of exceeding the ultimate tensile stress (900 Mpa for tita)
g(2) = abs(F)/(F_ultimate/2) - 1;

%If minimum length is smaller (because dl = L0-L) than critical buckling length, create barrier
g(3) = (-0.812*(1 - sqrt(1 - 6.87*(D/L0)^2))*L0 + L0)/(L_center) - 1;

%Prevent the true length from being smaller than the minimum length when loops are touching
g(4) = d*n/L_center - 1;

g_eq = [];

end


```

D.3.4. Tension constraints

This section of code shows the tension spring constraints that were used during the optimization of a realistic tension spring. The constraints are called in the main optimization file of the tension spring optimization. The definition of the constraints themselves have been discussed more extensively in the paper.

```
function [g, g_eq] = Spring_cons_ten_new(x, R)

%Params
x_0 = x(1);
L0 = x(2);
d = x(3);
D = x(4);
n = x(5);

E = 190e9;
G = E/2.6;
k = (d^4*G)/(64*n*(D/2)^3);

%Maximum spring length calc (in neutral position)
x1_center = cosd(0)*R;    %x (vertical height) coordinate of the joint tip
w1_center = sind(0)*R;    %horizontal coordinate of the joint tip. Can be either Z or Y, axisymmetricality makes them the same, hence it is called w1
L_center = sqrt((x1_center - x_0)^2 + w1_center^2);      %Length of spring

%Force and stress
F = k*(L_center - L0);
tau_eqv = 16*abs(F)*(0.5*D)/(pi*d^3) + 4*abs(F)/(pi*d^2);
sigma_ultimate = 1720e6;
F_ultimate = 308.2;

%If eqv stress is larger than half the ultimate stress.
g(1) =  tau_eqv/(sigma_ultimate/2) - 1;

%If the force is larger than 308.2N, the joint itself has the problem of exceeding the ultimate tensile stress (900 Mpa for tita)
g(2) = abs(F)/(F_ultimate/2) - 1;

%If L0 is smaller than smallest L0 (This excludes the possibility for further pretension), create barrier
g(3) = d*n/L0 - 1;

g_eq = [];

end


```

Bibliography

[1] Michael Sussman. “Duchenne muscular dystrophy”. In: *JAAOS-Journal of the American Academy of Orthopaedic Surgeons* 10.2 (2002), pp. 138–151.

[2] Yumen Bionics. URL: `https://yumenbionics.com/`.

[3] Larry L Howell. “Compliant mechanisms”. In: *21st century kinematics*. Springer, 2013, pp. 189–216.

[4] Juan A Gallego and Just L Herder. “Criteria for the static balancing of compliant mechanisms”. In: *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Vol. 44106. 2010, pp. 465–473.

[5] Mark Schenk and Simon D Guest. “On zero stiffness”. In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 228.10 (2014), pp. 1701–1714.

[6] Surena Namdari et al. “Defining functional shoulder range of motion for activities of daily living”. In: *Journal of shoulder and elbow surgery* 21.9 (2012), pp. 1177–1183.

[7] Tat Joo Teo et al. “A generic approximation model for analyzing large nonlinear deflection of beam-based flexure joints”. In: *Precision Engineering* 34.3 (2010), pp. 607–618.