Generating nonlinear load-displacement characteristics in mechanisms

by introducing misaligned over-constraints

Master thesis

Winkler van Grafhorst

Delft University of Technology

Generating nonlinear load-displacement characteristics in mechanisms

by introducing misaligned over-constraints

by

Winkler van Grafhorst

Winkler van Grafhorst
4459032

Supervisor:	G. Radaelli
Committee:	D. Farhadi
Committee:	W. van de Sande
Project Duration:	12, 2023 - 3, 2025
Faculty:	Mechanical Engineering, Delft
Department:	Precision & Microsystem Engineering

Cover:	Photograph of experimental setup of four-bar mechanism prototype with one compliant link and two misaligned over-constraints in a Zwick/Roell testing machine.
Style:	TU Delft Report Style

Preface

This work marks the conclusion of my academic journey at TU Delft. Over the years, I have gained a wealth of knowledge and developed a critical mindset that I will carry forward in my professional career. I am committed to applying my skills and expertise toward meaningful causes that contribute to society. This journey has been filled with challenges and successes, shaping me both as an engineer and as a person. Beyond my academic experience, I have been fortunate to be surrounded by inspiring mentors, colleagues, and friends. I would like to express my gratitude to all the TU Delft faculty members who have supported me throughout the years. A special thanks goes to Giuseppe Radaelli, whose confidence in my work and continuous encouragement have been invaluable. I will truly miss our bi-weekly meetings, which always motivated me to push forward with even greater determination. I would also like to extend my sincere appreciation to Patrick van Holst for his patience and support during the testing phase, and to the faculty workshop staff for their assistance in building the physical prototype. Finally, my gratitude goes to my family, girlfriend, and friends—their support has contributed to this work. It has been an incredible experience, and I look forward to the next chapter.

Winkler van Grafhorst
Delft, March 2025

Summary

The minimum number of bodies required to form a closed kinematic chain capable of motion is four. A planar four-bar mechanism has one degree of freedom (DOF), whereas in three-dimensional space, it becomes over-constrained, reducing its DOF to -2. While this kinematic idealization remains unaffected, its physical realization introduces conflicting constraints. This study deliberately incorporates misaligned over-constraints in four-bar mechanisms to achieve tailorable load-displacement characteristics by utilizing a compliant coupler link. Such tailorable characteristics could potentially eliminate the need for traditional springs and dampers in mechanical systems.

A positioning strategy for misaligned over-constraints is developed. An Euler-Bernoulli beam model with superposition is employed to analyze how misalignments influence the load-displacement response. Additionally, a numerical model using Simscape Multibody is implemented to verify the analytical results. To further validate the findings, a physical prototype is constructed and tested to compare real-world behavior with computational models. The numerical simulations effectively capture the response of the four-bar mechanism, considering elastic deformations in the compliant coupler link.

The results from all models exhibited strong agreement. Parameter relaxation is introduced to account for manufacturing tolerances and geometric imperfections. Furthermore, a parametric study is conducted to examine the influence of individual design parameters on the load-displacement characteristics. To facilitate design exploration, a graphical user interface (GUI) is developed, enabling users to tailor four-bar mechanisms based on specific load-displacement requirements. Two distinctive behaviors are presented: extended regions of nearly constant torque and sinusoidal load-displacement characteristics, both of which have potential applications in precision engineering and motion control.

Contents

Preface i

Summary ii

Nomenclature iv

1 Introduction 1

1.1 Background and motivation . . . 1

1.2 Research Objective . . . 1

1.3 Scope . . . 2

1.4 Structure of report . . . 2

2 Research Paper 3

3 Conclusion 20

A Appendix A - Euler-Bernoulli beam theory in MATLAB 21

B Appendix B - Simscape Multibody 30

C Appendix C - Graphical User Interface (GUI) in MATLAB App Designer 34

D Appendix D - Physical model 40

E Appendix E - Concepts 47

Nomenclature

Abbreviations

Abbreviation	Definition
FEA	Finite Element Analysis
GUI	Graphical User Interface
DOF	Degree of Freedom
ODE	Ordinary Differential Equation

Symbols

Symbol	Definition	Unit
G	Shear modulus	[GPa]
E	Young's modulus	[GPA]
I	Second moment of Area	[m^4]
J	Polar moment of Inertia	[m^4]
U	Energy	[J]
M	Bending moment	[Nm]
L_1	Ground link	[m]
L_2	Input link	[m]
L_3	(Compliant) coupler link	[m]
L_4	Output link	[m]
θ_1	Angle of ground link	[rad]
θ_2	Angle of input link	[rad]
θ_3	Angle of rigid coupler link	[rad]
θ_4	Angle of output link	[rad]
θ_a	Polar angle joint A	[rad]
$\theta'_{a,0}$	Initial angle of projected position vector wrt x_3	[rad]
θ'_a	Angle of projected position vector wrt x_3	[rad]
θ_b	Polar angle joint B	[rad]
$\theta'_{b,0}$	Initial angle of projected position vector wrt x'_3	[rad]
θ'_b	Angle of projected position vector wrt x'_3	[rad]
$\phi_{a,0}$	Initial azimuth angle joint A	[rad]
ϕ_a	Azimuth angle joint A	[rad]
$\phi_{b,0}$	Initial azimuth angle joint B	[rad]
ϕ_b	Azimuth angle joint B	[rad]
$\phi_{T,0}$	Initial angle of twist	[rad]
$\phi_{T,a,0}$	Initial angle of projected position vector joint A wrt x_3	[rad]
$\phi_{T,b,0}$	Initial angle of projected position vector joint B wrt x_3	[rad]
ϕ_T	Angle of twist resulting in torsion	[rad]
$\hat{\rho}_a$	Unit position vector joint A	$[\phi_a, \theta_a]$
$\hat{\rho}_b$	Unit position vector joint B	$[\phi_b, \theta_b]$
ρ'_a	Projection position vector joint A on $x_3 - x_2$-plane	$[x_2, x_3]$
ρ'_b	Projection position vector joint B on $x'_3 - x'_2$-plane	$[x'_2, x'_3]$
ρ''_a	Projection position vector joint A on $x_1 - x_3$-plane	$[x_2, x_3]$
ρ''_b	Projection position vector joint B on $x_1 - x_3$-plane	$[x_2, x_3]$

1
Introduction

1.1. Background and motivation

A four-bar linkage is the simplest closed-chain mechanism capable of motion, consisting of four rigid bodies connected by four revolute joints. From a planar perspective, such a mechanism has one degree of freedom (DOF). However, in the three-dimensional space, the same mechanism becomes over-constrained, resulting in a system with a theoretical DOF of –2. Although this does not affect the kinematic idealization, it introduces conflicting constraints when physically realized. These over-constraints can lead to internal stresses, reduced predictability, and, in some cases, buckling — especially when geometric imperfections arise from manufacturing tolerances or environmental influences such as thermal expansion.

While over-constraints are often considered undesirable, they also present unexplored potential. This research explores the deliberate use of misaligned over-constraints, in combination with a compliant coupler link, to tailor the load–displacement characteristics of a four-bar mechanism. By embedding compliance into one of the mechanism's links, it becomes possible to passively control force responses during motion, potentially reducing or eliminating the need for traditional springs, dampers, or actuators.

This approach not only aims to simplify mechanical systems but also to enable adaptive behavior through design. Tailored load–displacement responses have significant potential across various engineering applications, such as passive mass balancing, suspension systems, motion control, and singularity mitigation in linkages. Existing methods often require solutions like embedded springs, dampers or actuators, which can increase mass, cost, and design complexity. This work proposes an alternative: It deliberately introduces misaligned over-constraints in mechanisms to achieve tailorable load–displacement characteristics by leveraging a compliant link.

1.2. Research Objective

The primary objective of this research is to develop a methodology for designing four-bar mechanisms with misaligned over-constraints that produce nonlinear, tailorable load–displacement characteristics. The study combines theoretical analysis, numerical modeling, and physical prototyping to achieve the following goals:

- Analyze the influence of misaligned over-constraints using Euler–Bernoulli's classical beam theory and the principle of superposition.
- Validate analytical results through numerical simulations in Simscape Multibody.
- Experimentally verify the mechanical behavior using a physical prototype.
- Develop a graphical user interface (GUI) to support design exploration.
- Identify and characterize unique mechanical behaviors (e.g., significant regions of constant torque, sinusoidal-like responses).

1.3. Scope

The study focuses on a single degree-of-freedom four-bar mechanism in the three-dimensional space, in which only the coupler link is compliant. The joints are modeled as revolute joints, with two selected joints deliberately misaligned to introduce the misaligned over-constraints. These joints are located on either end of the compliant coupler link. The compliant link is straight and stress-free in the initial configuration of the four-bar, with a circular and constant cross-section. These assumptions are kept consistent to effectively compare and analyze the effects of misalignment on the mechanism's load–displacement characteristics.

1.4. Structure of report

This report consists of a research paper, which forms the core of the work, followed by several appendices that provide supplementary materials:

- **Chapter 2** presents the research paper, which details the methodology, modeling approaches, results, and conclusions. This chapter can be read independently.
- **Appendix A** contains MATLAB code used for analyzing the four-bar mechanism based on Euler–Bernoulli beam theory and the principle of superposition.
- **Appendix B** includes supplementary code for the Simscape Multibody model.
- **Appendix C** presents the MATLAB App Designer code for the GUI that allows interactive design exploration.
- **Appendix D** provides additional photographs of the experimental setup and prototype, along with technical drawings of the machined parts.
- **Appendix E** highlights initial concepts and design iterations that preceded the final design and may serve as a foundation for future research.

2

Research Paper

Generating nonlinear load-displacement characteristics in mechanisms by introducing misaligned over-contraints

Author: Winkler van Grafhorst, supervisor: Dr. ir. G. Radaelli

Abstract—**The minimum number of bodies required to form a closed kinematic chain capable of motion is four. A planar four-bar mechanism has one degree of freedom (DOF), whereas in three-dimensional space, it becomes over-constrained, reducing its DOF to negative two. While the kinematic idealization remains unaffected, its physical realization introduces conflicting constraints. This study deliberately incorporates misaligned over-constraints in four-bar mechanisms to achieve tailorable load-displacement characteristics by utilizing a compliant coupler link. Such tunable characteristics could potentially eliminate the need for traditional springs and dampers in mechanical systems.**

A positioning strategy for misaligned over-constraints is developed. An Euler-Bernoulli beam model with superposition is employed to analyze how misalignments influence the load-displacement response. Additionally, a numerical model using Simscape Multibody is implemented to verify the analytical results. To further validate the findings, a modular physical prototype is constructed and tested to compare real-world behavior with computational models. The numerical simulations effectively capture the response of the four-bar mechanism, considering elastic deformations in the compliant coupler link.

The results from all models exhibit strong agreement. Parameter relaxation is introduced to account for manufacturing tolerances and geometric imperfections. Furthermore, a parametric study is conducted to examine the influence of individual design parameters on the load-displacement characteristics. To facilitate design exploration, a graphical user interface (GUI) is developed, enabling users to tailor four-bar mechanisms based on specific load-displacement requirements. Two distinctive behaviors are presented: extended regions of nearly constant torque and sinusoidal load-displacement characteristics, both of which have potential applications in precision engineering and motion control.

Keywords: Four-bar mechanism, misalignment, over-constraint, nonlinear, load-displacement, constant torque, revolute joints, compliant mechanism.

I. Introduction

Degrees Of Freedom (DOF) determine a mechanism's mobility. Before any mechanism is designed, the number of DOF should be determined [1]. Various theories for calculating a mechanism's mobility have been proposed. The development of mobility criteria has been an area of research for over 150 years [2]. Many of these methods can be traced back to the same fundamental criterion, known as the Chebychev–Grübler–Kutzbach mobility criterion [3]. It is applicable for calculating the mobility of planar mechanisms, and can also be adapted for spatial mechanisms. The criterion for spatial mechanisms is provided by equation 1.

$$M = 6(n - j - 1) + \sum_{i=1}^{j} f_i \tag{1}$$

For a four-bar linkage in the three dimensional space - with $n = 4$ bodies, $j = 4$ kinematic pairs, and each ith kinematic pair allows one DOF - this equates to minus two DOF. Therefore the four-bar is considered over-constrained. Over-constrained mechanisms may be desirable due to their enhanced load-bearing capacity compared to exact constrained mechanisms. In addition, Eigenfrequencies are reduced, and a high degree of symmetry is often involved [4]. While the kinematic idealization of a mechanism is not affected by over-contraints, the physical realization is [5]. Geometric imperfections, arising from manufacturing tolerances or thermal expansion differences, effectively transform redundant constraints into conflicting ones. This can cause internal stresses, part deformation, and increased loads on bearings in mechanisms, potentially leading to nondeterministic behavior [6], or even complete immobility. In compliant mechanisms, misalignments can lead to undesirable static and dynamic behavior, including buckling, variations in support stiffness, and bifurcation [7]. However, these misalignments also present a potential opportunity that is often overlooked. By deliberately introducing extreme misaligned over-constraints in a compliant mechanism, it is possible to achieve tailorable load-displacement characteristics. In this approach, compliance is incorporated into one of the links, enabling a new method of controlling mechanical response. Traditionally, tailored load-displacement characteristics are achieved by incorporating actuators, springs, or dampers into mechanisms to perform specific functions. This alternative strategy leverages the inherent flexibility of compliant elements, potentially reducing system complexity while enhancing adaptability. Examples of traditional approaches to generate nonlinear load-displacement characteristics include the work of S. Liu et al., who embedded flexible joints within a metastructure to achieve nonlinear joint stiffness [8]. Furthermore, X. Jing et. al present a review on a method for nonlinear stiffness manipulation and employment in an X-shaped mechanism [9], [10]. Besides, spring elements are also used in the design of machinery and automated systems where gravity is a key factor that must be considered. Improper design can result in excessive force and actuator demands, which can be mitigated through passive mass balancing [11]. However, this approach increases system mass, leading to added weight and higher resource consumption. These challenges exemplify just a few of the many trade-offs in mechanism design. This research aims for another direction; not by adding springs, dampers and masses to the system, but by utilizing compliance of a part of a mechanism itself. By accurately positioning misaligned over-constraints, compliant elements will function as the loading and unloading of a spring as the mechanism moves. The effect of misaligned over-constraints in mechanisms have been studied before; however, it was applied to a mechanism where compliant joints absorbed the forces [4]. In contrast, this study implements compliance in the links rather than the joints. The misaligned over-constraints are revolute joints positioned under an angle, allowing for one DOF. The goal is to develop a methodology for the development of nonlinear load-displacement characteristics in mechanisms by utilizing misaligned over-constraints. Such mechanisms can have a wide range of applications; from mass-balancing systems, suspension systems, to smoothening of peak torques and forces in mechanisms. The use of (strong) actuators, springs and dampers could become redundant in some applications. Additionally, misaligned over-constraints can be strategically incorporated into mechanisms to mitigate the effects of singularities. For example, they enable the conversion of reciprocating actuation into rotary motion using linkages, ensuring efficient torque transmission even near kinematic singularities [12]. Since mechanisms vary widely in form and function, this research focuses on the mechanical response of misaligned over-constraints in four-bar mechanisms — the simplest closed-chain linkage that exhibits motion.

The structure of this paper is as follows. Section II, outlines the methodology that led to the results. It introduces the underlying concepts, including Euler-Bernoulli's beam theory and the principle of superposition, and presents the numerical Simscape model, the physical prototype and the experimental setup. Thereafter, in Section III, the results of the various models are compared and analyzed. A parameter study illustrates how different variables affect the load–displacement characteristics of the system. Furthermore, a Graphical User Interface (GUI) is presented, enabling users to quickly explore unique load–displacement behaviors. Two notable behaviors identified during this research are a plateau of constant torque and a sinusoidal torque response. In Section IV, potential improvements are proposed and future research directions are explored. Finally, the conclusion is presented in Section V.

II. Methodology

To generate nonlinear load-displacement characteristics, multiple steps are involved. These steps are covered in this section. First the concept will be explained in Section II-A. In section II-B, a simplified model will be presented: Euler-Bernoulli's beam theory and the principle of superposition. This simplified model serves as an effective tool for interactively designing and optimizing desired load–displacement characteristics. Next, the numerical model developed in Simulink Simscape is detailed in Section II-C. This model supports validation and refinement of the subsequent design stage. Finally, the physical model that was developed and tested is presented in Sections II-D and II-E, respectively.

A. Concept

Misaligned over-constraints in precision mechanisms might seem odd; however, this study incorporates misaligned over-constraints in combination with a compliant beam to deploy nonlinear load-displacement characteristics. The least amount of links required in a closed chain involving motion is four. A four-bar linkage has negative two degrees of freedom. Formulated differently: it has three over-constraints. The effect of misaligned over-constraints will be demonstrated on a four-bar linkage with four revolute joints. The concept is applicable to other over-constraint mechanisms, provided their kinematic chain is known, and their *aligned* counterpart does involve motion. Other types of joints, such as sliders, can also be utilized. The four-bar linkage consists of a ground link (L_1), input link (L_2), coupler link (L_3) and output link (L_4). Their absolute angles are indicated with θ_1, θ_2, θ_3 and θ_4 respectively, as shown in figure 1. The ground link is parallel to the x-axis, $\theta_1 = 0$.

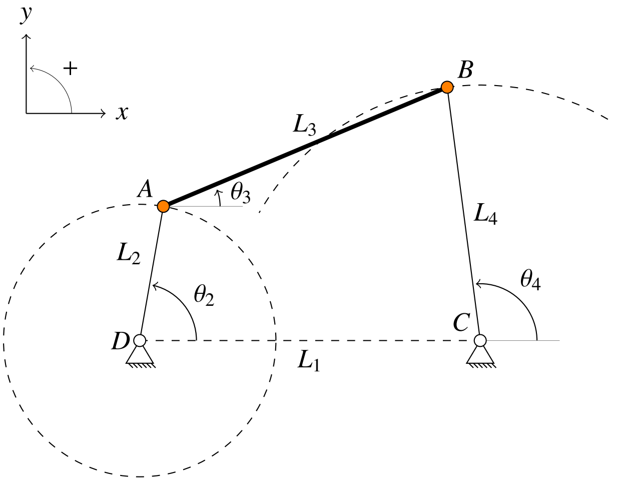


Fig. 1: Crank-Rocker four-bar mechanism, L_1 is the ground link (with $\theta_1 = 0$, not indicated), L_2 is the input link (θ_2), L_3 represents the coupler link (θ_3) and L_4 is the output link (θ_4). The misaligned over-constraints will be introduced to joint A and B, indicated in orange.

A subscript zero is used to indicate angles in the initial configuration, as in $\theta_{2,0}$. The coupler link is the compliant link, indicated by a **thick** line. This compliant link has a circular cross section with radius R. The joints that are misaligned with respect to a pure out of plane direction, are joints A and B, on either side of the compliant coupler link. Because the joints are nonparallel, the compliant link must deform to accommodate the kinematics. The deformation of the compliant link varies over the four-bar's range of motion and depends strongly on the design variables, including the misalignment parameters, resulting in complex load–displacement characteristics. The theory is demonstrated on a crank-rocker mechanism because it allows infinite motion of input angle θ_2. A crank-rocker mechanism is identified by the Grashof criterion [13]. All plots throughout this paper have θ_2^* on the x-axis, indicating

$\theta_2 - \theta_{2,0}$. As result, the x-axis of all plots ranges from 0 to 2π. In addition, the properties of the steel compliant link are as follows; E = 200 GPa, G = 75 GPa [14]; R = 1 mm. Finally, an established sign convention is used. The positive directions are as follows: load acts upwards; internal shear force causes clockwise rotation of the beam segment on which it acts; the internal moment causes compression in the top fibers; and clockwise rotation of angles.

B. Euler-Bernoulli Beam Theory and superposition

In this section the theory is split into two parts; the first part elaborates on the kinematics of the four-bar linkage and misaligned over-constraints. The second part focuses on how the misaligned over-contraints result in deformation of the compliant beam. Finally, the two parts are analyzed together to determine the mechanism's load–displacement characteristics in the absence of external loads. In the final step, external loading conditions are applied. Throughout this subsection, the configuration of the four-bar mechanism and the misalignment parameters remain unchanged.

1) ***Kinematics****:* The vector loop equation for a four-bar linkage with *aligned* over-constraints is formulated in Equation 2.

$$L_2 e^{\theta_2 i} + L_3 e^{\theta_3 i} - L_4 e^{\theta_4 i} - L_1 e^{\theta_1 i} = 0 \tag{2}$$

The position analysis of the four-bar mechanism is based on Freudenstein's equation [15] and solved numerically using fsolve in MATLAB over input angles ranging from $\theta_{2,0}$ to $\theta_{2,0} + 2\pi$. The solution returns θ_3 and θ_4 for all input angles of θ_2. For a crank-rocker mechanism, there are two possible solutions except at singularity position [16]. At singularity, the four-bar linkage gains or loses a DOF. The two branches are visualized in figure 2. The blue links showcase the cross-configuration, whereas the black links showcase the open configuration.

The initial position of the *aligned* four-bar linkage is set by $\theta_{2,0}, \theta_{3,0}$ and $\theta_{4,0}$. Assumed is that for a similar configuration, the location of joint A and B of the *aligned* four-bar, correspond exactly with the locations of A and B of the misaligned four-bar. This assumption is justified since the misalignments are small and therefore have a negligible effect on the length of the compliant beam once the compliant beam is deformed. To improve manufacturability, every initial configuration, at $\theta_{2,0}$, the four-bar is designed stress free, i.e., an undeformed compliant coupler link. Figure 4 illustrates the compliant coupler link in its initial, undeformed configuration, at $\theta_{2,0}$. Once θ_2 starts to rotate, the compliant coupler link will deform, as shown in figure 5. The axes of revolution of the *aligned* four-bar correspond to x_3 and x'_3 for joint A and B respectively. The axes of revolution of the misaligned revolute joints in A and B corresponds to the position vectors, $\hat{\rho}_a$ and $\hat{\rho}_a$ respectively. These position vectors are specified by the polar angle ($\theta \in [0, \frac{\pi}{2}]$)[1] and azimuth angle ($\phi \in [0, 2\pi]$).

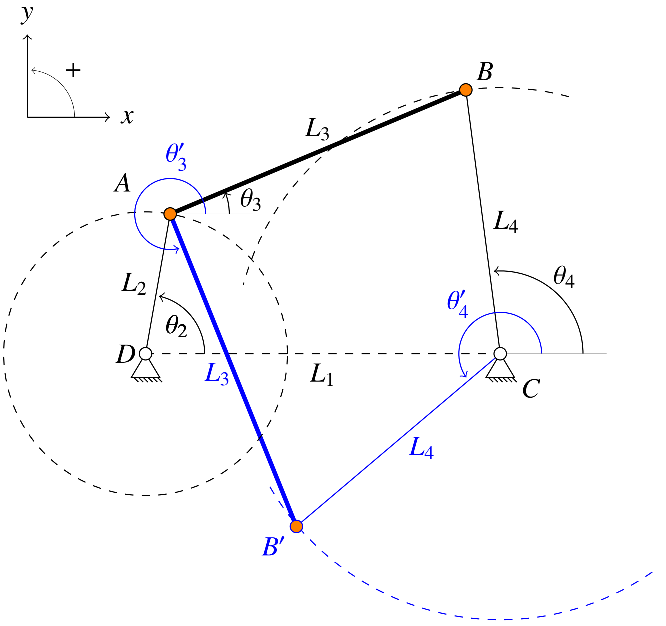


Fig. 2: Two branches of a Crank-Rocker four-bar mechanism: open and cross-configuration (the latter indicated in blue). Each four-bar mechanism has two solutions, resulting in different load-displacement characteristics once the misaligned over-constraints are introduced.

The polar angle represents the magnitude of the misalignment, and remains constant throughout the motion of the four-bar linkage. The azimuth angle defines the rotation of the radial line around the polar axis, and varies throughout the motion of the four-bar linkage. The azimuth in joint A, θ_a, starts at a selected value $\theta_{a,0}$ and varies throughout one revolution with the relative angle between the input link (L_2) and the coupler link (L_3), corresponding to $\theta_3 - \theta_2$. Similarly, the azimuth in joint B, θ_b, starts at a selected $\theta_{b,0}$ and varies with a relative angle corresponding to $\theta_4 - \theta_3$. For a given initial configuration, the azimuth angles in joint A and B are plotted in figure 3.

Next, the influence of position vectors on the boundary conditions at joints A and B will be presented. Over a full revolution of θ_2, the position vectors in joint A and B constrain the compliant coupler link. The slope of the compliant link at joint A equals the slope of the projection of the position vector of joint A on the x_3x_2-plane with respect to x_2, plus a shift. This shift ensures that for each selected initial position of the four-bar mechanism, the compliant coupler link remains in its undeformed state. The same approach applies to the slope of the compliant beam at joint B. This is visualized in figure 6. The projections of the position vectors are indicated with $\vec{\rho}'_a$ and $\vec{\rho}'_b$. Their angles are calculated using equations 3 and 4 respectively. Substituting the azimuth angles with the initial azimuth angles results in the shift that ensures a stress-free initial configuration of the four-bar mechanism.

$$\theta'_a = \arctan \frac{\sin \theta_a \sin \phi_a}{\cos \theta_a} \tag{3}$$

[1] In mathematics, the polar angle is defined over the interval $[0, \pi]$; however, in this case, once it exceeds $\frac{\pi}{2}$, the configuration may admit an equivalent description due to symmetry.

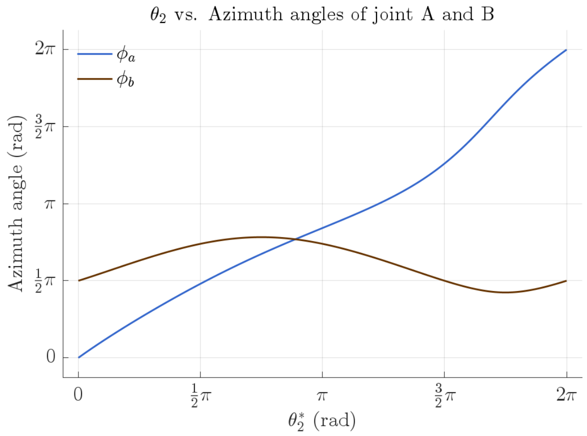


Fig. 3: The azimuth angle of joint A and B as function of θ_2^*. In this plot, the following configuration is used: $\phi_{a,0} = 0$ rad, $\phi_{b,0} = \frac{\pi}{2}$ rad and $\theta_2^* = \theta_2 - \theta_{2,0}$ rad. $\theta_{2,0} = \frac{\pi}{4}$. ϕ_a Completes one full revolution from 0 to 2π, whereas ϕ_b completes a partial revolution.

$$\theta_b' = \arctan \frac{\sin \theta_b \sin \phi_b}{\cos \theta_b} \tag{4}$$

Next to bending, the compliant coupler link is subjected to torsion. The angle of twist equals the relative angle between the projected position vectors of the spherical coordinate systems on the x_1x_3-plane, from figure 4 and 5. The angles that result in torsion are shown in figure 7. Similar to bending, a shift is added to ensure zero torsion at the selected initial position, $\theta_{2,0}$. The angle of the projection of the position vectors on the $x_3 - x_2$-plane with the x_3-axis from figure 4 and 5, is calculated using equation 5 and 6 for the misalignments in joint A and B respectively. The relative torsion angle (ϕ_T) equals $\phi_{T,b} - \phi_{T,a}$. For a given initial configuration, figure 8 shows a plot of the torsion angle and their relative angle as a function of θ_2^*.

$$\phi_{T,a} = \arctan \frac{\sin \theta_a \cos \phi_a}{\cos \theta_a} \tag{5}$$

$$\theta_{T,b} = \arctan \frac{\sin \theta_b \cos \phi_b}{\cos \theta_b} \tag{6}$$

*2) **Kinetics**:* With the boundary conditions of the compliant beam at joint A and B fully defined, the focus now shifts to the kinetics. The beam will be subjected to torsion and bending as a function of θ_2. Because the loading is linearly related to the stress that is to be determined; and the loading does not significantly change the beam's geometry, superposition can be applied [14]. From the elastic curvature of the compliant beam, the bending moments at both ends can be determined. First, the mechanism will be considered with zero external loading on the output link. A free body diagram of the compliant beam, together with the moment diagram and torsion diagram is shown in figure 9. The directions of all loadings is subjected to change as θ_2's angle changes. The compliant beam is subjected to two varying moments at either end (M_a and M_b). The moment function is given by equation 7. E is the Young's modulus of the compliant beam, and I is the polar moment of inertia. For a given θ_2, the torque remains constant along the length of the compliant beam due to its uniform cross-sectional area. Likewise, the shear force remains constant at each value of θ_2. The internal moment varies along the length of the beam, with M_a and M_b potentially differing in both magnitude and direction. Integration of the moment function with respect to x, results in the slope function, given by equation 8. Integrating the slope function, gives the deflection function, given by equation 9.

$$M(x) = EI\frac{d^2v}{dx^2} = \frac{M_b - M_a}{L_3}x + M_a \tag{7}$$

$$EI\frac{dv}{dx} = \int M(x) \cdot dx = \frac{1}{2}\frac{M_b - M_a}{L_3}x^2 + M_a x + C_1 \tag{8}$$

$$EIv(x) = \int EI\frac{dv}{dx}dx = \frac{1}{6}\frac{M_b - M_a}{L_3}x^3 + \frac{1}{2}M_a x^2 + C_1 x + C_2 \tag{9}$$

Solving the boundary conditions, given by the set of equations 10, yields the integration constants C_1 and C_2, as shown in equation 11.

$$\begin{aligned} EI\frac{dv}{dx}(x_2 = 0) &= \theta_a, \\ EI\frac{dv}{dx}(x_2 = L_3) &= \theta_b, \\ EIv(x_2 = 0) &= 0, \\ EIv(x_2 = L_3) &= 0 \end{aligned} \tag{10}$$

$$\begin{aligned} C_1 &= -\frac{M_b - M_a}{6}L_3 - \frac{1}{2}M_a L_3, \\ C_2 &= 0. \end{aligned} \tag{11}$$

Solving the system of equations result in the bending moments at position A and B of the compliant beam. The solution is given by Equation 12 and 13 respectively. For a given configuration, the moments in A and B are plotted versus θ_2^* in Figure 10. The varying bending moments in A and B over θ_2^* result in a varying elastic curve, plotted in Figure 11.

$$M_a = -\frac{2EI(2\theta_a - \theta_b)}{L_3} \tag{12}$$

$$M_b = \frac{2EI(\theta_a + 2\theta_b)}{L_3} \tag{13}$$

The torsion is calculated using Equation 14, where G is the shear modulus, J is the polar moment of inertia and ϕ_T is the angle of twist.

$$T = \phi_T \frac{JG}{L_3} \tag{14}$$

Similar to torsion, the shear force along the beam is constant. Since the compliant beam is slender, deformations due to shear are negligible compared to those caused by bending. Moreover,

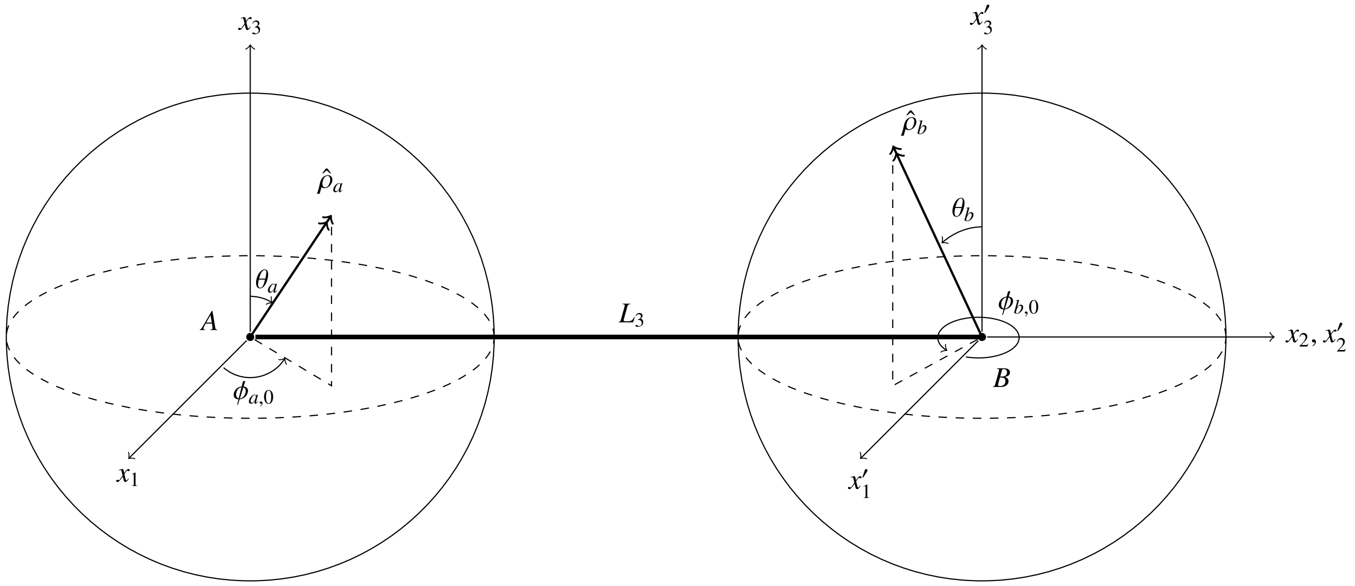


Fig. 4: The Compliant coupler link (L_3) in between two misaligned revolute joints. Input link and output link are not shown, x_1 and x_1' are always perpendicular to the coupler link of the *aligned* four-bar. The position vector of a spherical coordinate system represents the axis of revolution of the misaligned joint. This vector is described using the azimuth angle ($\phi \in [0, 2\pi]$) and the polar angle ($\theta \in [0, \frac{\pi}{4}]$). This figure shows the compliant coupler link (L_3) in its undeformed state at a selected initial configuration. x_1 And x_1' are always perpendicular to the *aligned* four-bar mechanism's coupler link. x_2 and x_2' are in line with the *aligned* four-bar mechanism's coupler link.

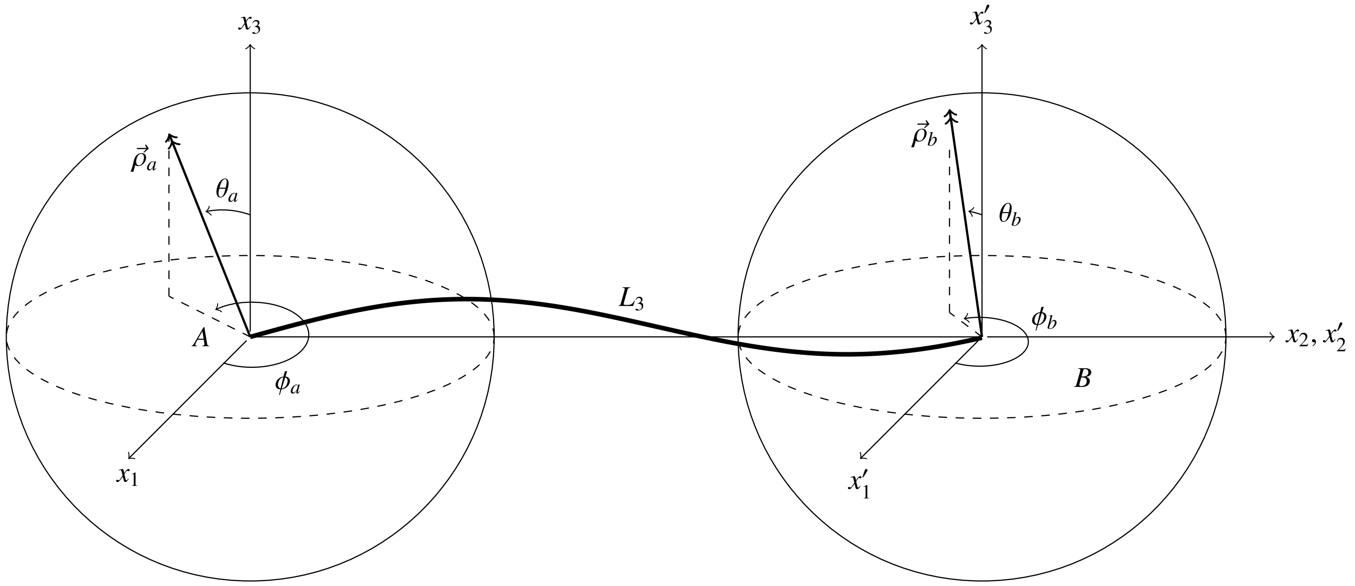


Fig. 5: Figure of deformed state of compliant coupler link at a given position of θ_2. ϕ_a and ϕ_b are functions of θ_2. Input and output link are not shown, x_1 and x_1' are always perpendicular to the *aligned* four-bar counterpart. The position vectors, $\hat{\rho}_a$ and $\hat{\rho}_b$ do rotate with azimuth angle, inducing bending and torsion on the compliant coupler link. The magnitude of the misalignments (polar angles) remain the same throughout a revolution of θ_2 and can only be changed between different configurations.

classical beam theory (Euler–Bernoulli) does not account for shear deformations.

At this point, the internal bending moment as a function of x_2 and implicitly of θ_2, as well as the torsion as an implicit function of θ_2, are known. The bending strain energy and torsional strain energy can then be computed using Equations 15 and 16 respectively.

$$U_{bending} = \int_0^{L_3} \frac{M(x)^2}{2EI} dx \tag{15}$$

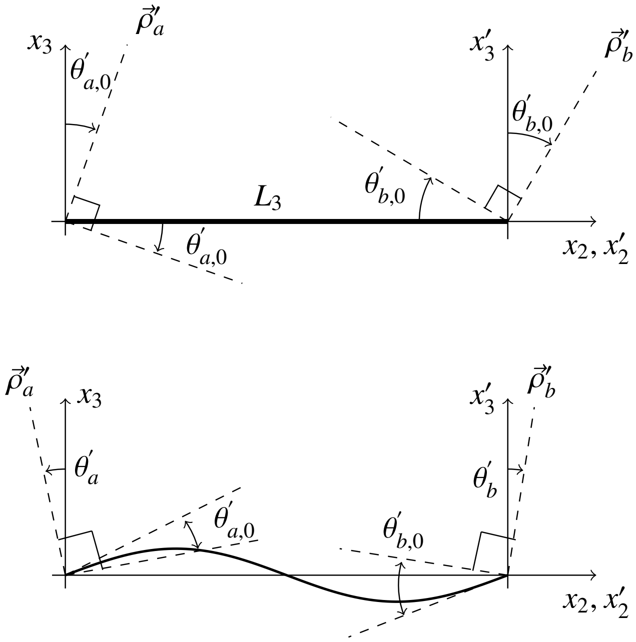


Fig. 6: Top: undeformed compliant coupler link at the selected initial position ($\theta_{2,0}$). The shift of $\theta_{a,0}$ and $\theta_{b,0}$ ensures no deformation. Below: Bending mode of compliant coupler link at a given position of θ_2. Shifts are added to the angle of the projected position vectors. The projected position vectors $\vec{\rho}'_a$, $\vec{\rho}'_b$ are projections of $\vec{\rho}_a$ and $\vec{\rho}_b$ as shown in figure 4 on the $x_3x_2/x'_3x'_2$-plane.

$$U_{torsion} = \frac{T^2 L_3}{2GJ} \tag{16}$$

The superimposed strain energy in the compliant beam is given by equation 17. Figure 12 shows how torsion strain energy and bending strain energy contribute to the total strain energy for a given initial configuration.

$$U_{tot} = U_{bending} + U_{torsion} \tag{17}$$

To calculate the torque of the input link as a function of θ_2, the total strain energy has to be differentiated with respect to θ_2, shown by equation 18. This results in the angular displacement-torque characteristics of a four-bar with misaligned over-constraints, as shown in figure 13.

$$\tau = \frac{dU_{tot}}{d\theta_2} \tag{18}$$

Thus far, the mechanism was considered without external loading on the output link. In general, a four-bar mechanism transfers motion, force or energy from a source to an output. Because the compliant beam is in the kinetic chain of the mechanism, the beam will be subjected to compressive or tensile loading, depending on its configuration. As the compliant link deforms due to bending during a rotation of θ_2, axial compressive and axial tensile strength of the beam is affected by the initial curvature of a member's axis [17], [18]. For a perfectly straight beam, its behavior remains deterministic as long as the applied load does not exceed the critical buckling load, defined by Equation 19.

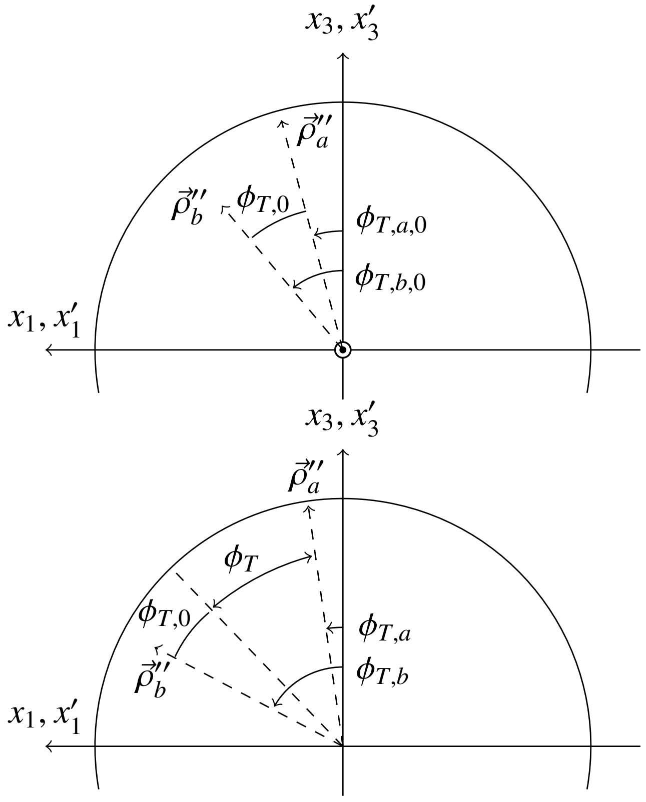


Fig. 7: Projections of $\vec{\rho}_a$ and $\vec{\rho}_b$ on x_3x_1-plane. Relative angle of the two projections equals the angle of twist, plus a shift. This results in torsion of the compliant coupler link throughout a revolution of θ_2.

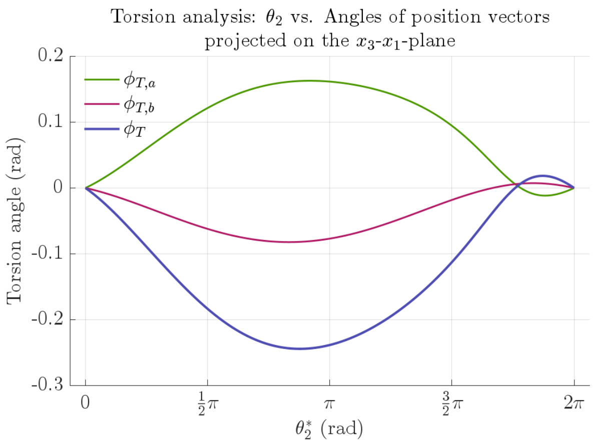


Fig. 8: Torsion angle of joint A and joint B and their relative angle, resulting in the angle of twist of the compliant beam. $\theta_2^* = \theta_2 - \theta_{2,0}$. $\theta_{2,0} = 30°$. For this plot, the azimuth angles are $\phi_{a,0} = 210°$, $\phi_{b,0} = 48°$, the misalignments are $\theta_a = \theta_b = 5°$ and beam lengths are $L_1 = 0.20$ m, $L_2 = 0.08$ m, $L_3 = 0.23$ m and $L_4 = 0.15$ m.

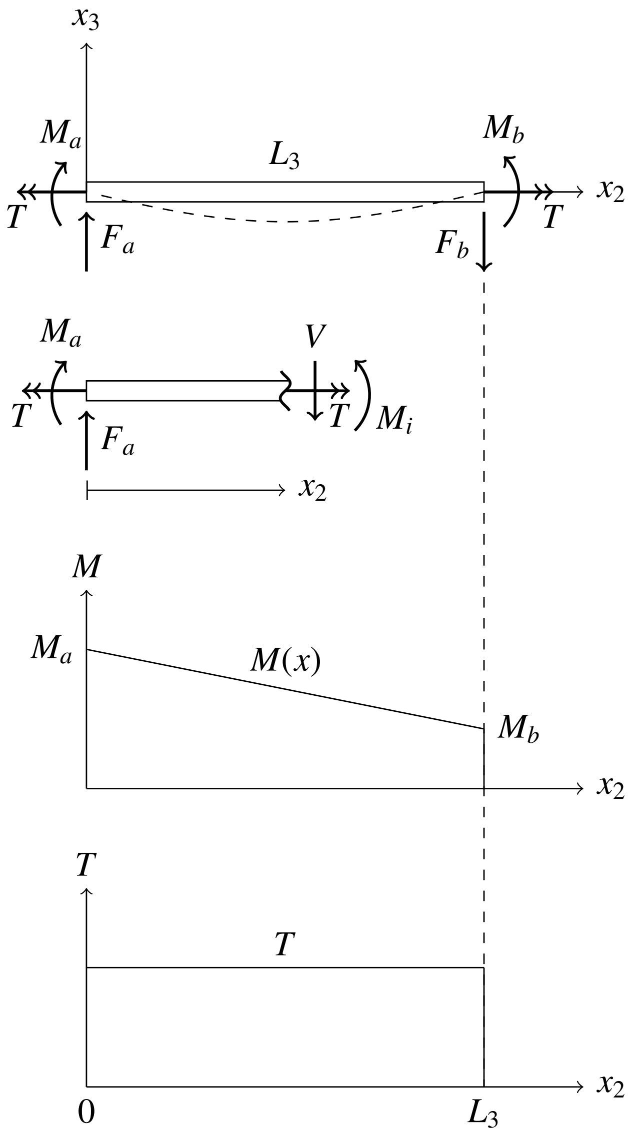


Fig. 9: Compliant coupler link with two moments on either end and torsion. Estimated deformation mode is indicated with a dashed line. Underneath, a section cut is made at distance x_2 to analyze internal bending moments and shear force. Internal bending moment equals $M_i = M_a + F_a x_2$, and shear force equals $V = -F_a$. The bottom two diagrams show moment and torsion as function of x_2.

$$P_{cr} = \frac{\pi^2 EI}{L_3^2} \tag{19}$$

The beam is perfectly straight when $\theta_2^* = 0 \vee \theta_2^* = 2\pi$, for other values, the beam is in a deformed state as shown in Figure 11. Therefore, for deterministic behavior the beam should at least not exceed the critical load in the initial position. Once the beam is bent, the axial strength of the beam is influenced by shape of the elastic curve. FEA would efficiently automate the solution process. This will be discussed in Section II-C.

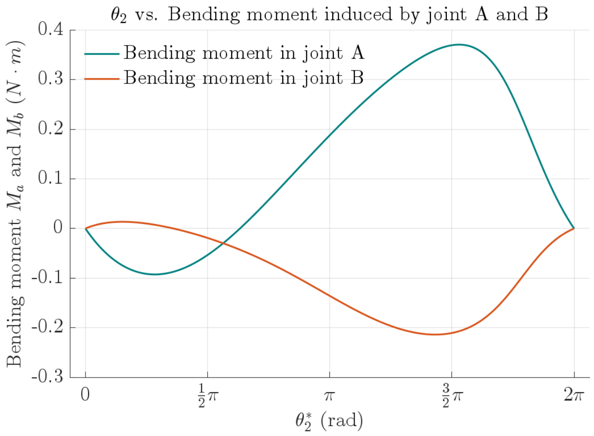


Fig. 10: Bending moments in joint A and B as function of $\theta_2^* = \theta_2 - \theta_{2,0}$. For this plot, the azimuth angles are $\phi_{a,0} = 210°$, $\phi_{b,0} = 48°$, the misalignments are $\theta_a = \theta_b = 5°$ and beam lengths are $L_1 = 0.20$ m, $L_2 = 0.08$ m, $L_3 = 0.23$ m and $L_4 = 0.15$ m.

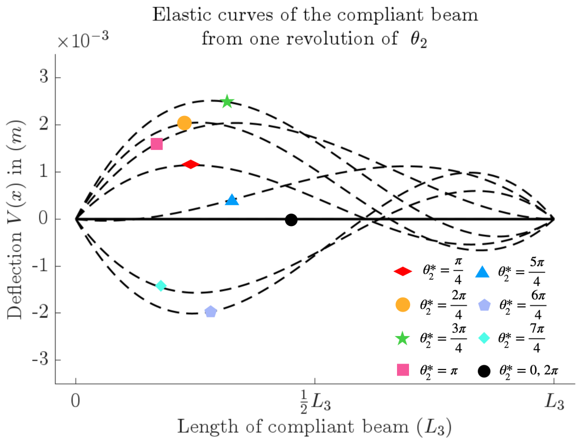


Fig. 11: Elastic curves of compliant coupler link (L_3) over one revolution of θ_2, at a given initial configuration $[\theta_{2,0}\ \theta_a\ \theta_b\ \phi_{a,0}\ \phi_{b,0}] = [30°\ 5°\ 5°\ 210°\ 48°]$. Beam lengths are $L_1 = 0.20$ m, $L_2 = 0.08$ m, $L_3 = 0.23$ m and $L_4 = 0.15$ m.

C. Numerical model

To validate the approach from the previous section, a numerical model in Simulink Simscape is made. Part of the block diagram is shown in figure 14. Note that this figure shows an applied constant torque of 0.5 Nm to the output link, this value remains zero for the unloaded situation. The blocks are obtained from the Simulink and Simscape library. The misaligned over-constraints are revolute joint blocks, positioned with rigid transforms. The input, output and ground link are solid blocks and the compliant coupler link is a general flexible beam with circular cross-section. The general modeling of the flexible beam can be viewed as a transient dynamic analysis of a rigid body coupled with integrated finite element analysis (FEA). This general flexible beam is capable of elastic deformations. A ramp signal is used to provide a

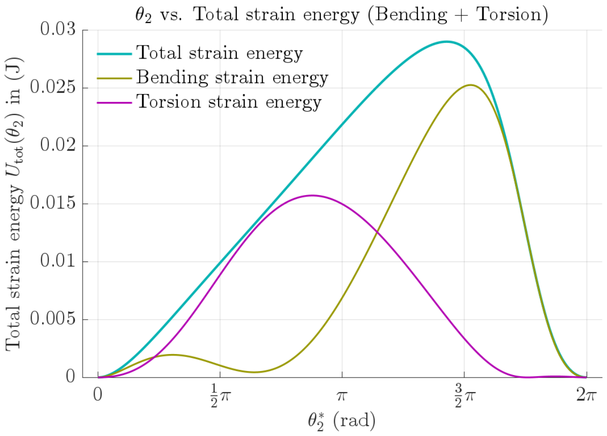


Fig. 12: Bending strain energy, torsion strain energy and their superimposed strain energy. Constant slope results in a region of constant torque. $\theta_2^* = \theta_2 - \theta_{2,0}$ where $\theta_{2,0} = 30°$. For this plot, the azimuth angles are $\phi_{a,0} = 210°$, $\phi_{b,0} = 48°$, the misalignments are $\theta_a = \theta_b = 5°$ and beam lengths are $L_1 = 0.20$ m, $L_2 = 0.08$ m, $L_3 = 0.23$ m and $L_4 = 0.15$ m.

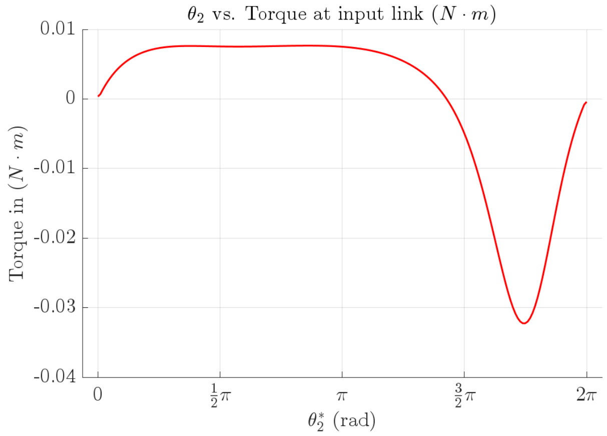


Fig. 13: Derivative of total strain energy with respect to θ_2 results in the torque profile of the input link as function of θ_2^*.

constant angular velocity to the input link. Since a mechanical system with constraints is analyzed, a stiff, variable-step solver (ODE23t) is used [19]. To minimize inertial effects, the mass of the solids is reduced, and the constant angular velocity is maintained moderate at $2\pi/10$ rad/s. The positioning of the misalignments are with respect to the compliant beam, and are parametrized using the same spherical coordinates as discussed in the previous section. Gravity is set to zero. The torque is measured from joint A as result of a constant angular velocity input signal. The general flexible beam is discretized into three elements, as it showed to be the optimal trade-off between simulation time and a detailed result. For a given initial configuration, similar to the four-bar configuration of the previous section, output data (angle and torque) are plotted and shown in figure 17.

In addition to the unloaded four-bar, a loaded four-bar is modeled; however, Simscape does not inherently account for axial compressive or tensile strength changes due to beam deflection in a way that captures geometric stiffening or buckling effects. It does consider length variations due to deformations.

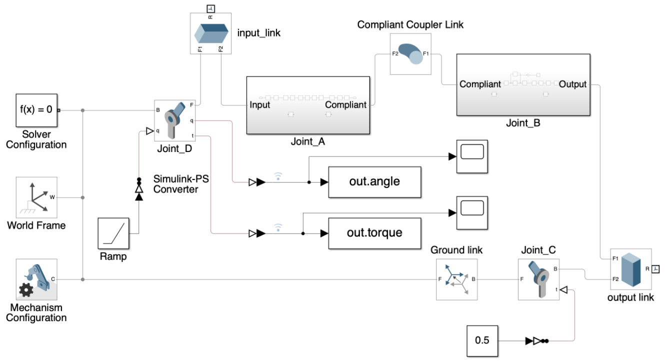


Fig. 14: Simscape Multibody model of four-bar linkage. Joint D is driven with a constant angular velocity. At joint D, the torque is automatically computed and sent to the workspace. The same applies to the angular displacement. Joint A and B are located in two subsystems. The misalignments are located in the subsystems and modeled as rigid transform blocks involving rotation. compliant beam is modeled as general flexible beam with $n = 3$ elements. All other body are modeled as solid bricks. Joints are modeled as revolute joints with zero stiffness and damping. In this figure a constant torque of 0.5 Nm is applied to the output link.

D. Physical model

A physical model is made to validate theoretical predictions and to test real-world behavior that the simulations might overlook. A render of the model is shown in figure 15. To obtain a high degree of reliability and repeatability, the compliant beam was made out of a steel alloy material. The diameter of the compliant beam equals 2 mm. The input and output link are machined out of a 6000-series aluminium. To reduce friction between the links, a pair of ball bearings is used for every joint. The bearings were press-fitted into their position. The mechanism was designed to change as many variables as possible without sacrificing reliability. As a result, the length of all links can be changed, the azimuth angles can be varied and the initial starting position of the input link with a stress free compliant beam can be chosen; in addition, the misalignments can be varied by producing additional cylinders with other polar angles. Laser engravings make sure the reading of azimuth angles is within a margin of five degrees. The joints were made with 5° polar angles. The axles of all joints have a diameter of 4 mm. The cylinders that hold the bearings are hold in place using set screws.

E. Experiment set-up

The experiment was executed at room temperature using a Zwick/Roell universal testing machine with a static torque

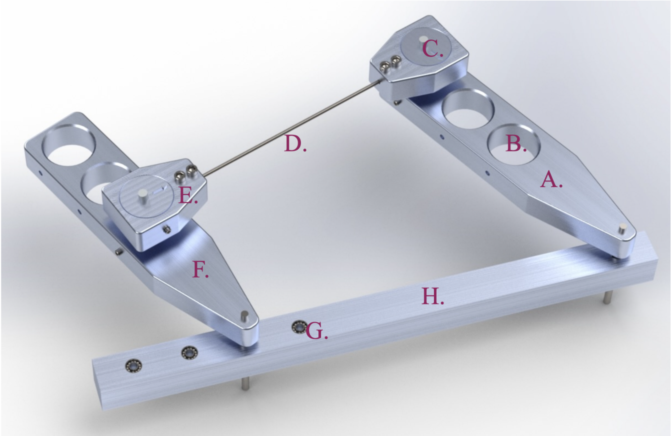


Fig. 15: Render of the physical model. Most system parameters are interchangeable. (A) Output link; (B) Mounting position; (C) Adjustable misaligned over-constraint cylinder with a pair of ball bearings; (D) Compliant coupler link; (E) Connector between coupler link and misaligned joint cylinder; (F) Input link; (G) Alternative mounting position for the input link with ball bearings; (H) Ground link.

sensor. The four-bar mechanism is fixed upside down to a Thorlabs frame structure. This structure connects the frame to the torque sensor without limiting the range of motion of the mechanism, see Figure 16. The input link is driven with a constant angular velocity of 2π rad/min. This speed was determined by comparing experimental results with lower and higher angular velocities. To reduce gravitational effects, the mechanism's motion was set perpendicular to gravity. In addition to experiments with misaligned joints, experiments with *aligned* revolute joint cylinders were conducted. These were subtracted from the misaligned results to isolate and mitigate the effects of a shifting center of mass, which could introduce imbalance in the mechanism and influence the load–displacement characteristics. Simultaneously, general bearing friction is eliminated by extracting these results. Finally, every measurement consisted out of four revolutions in positive direction, followed by four reverse rotations. Figure 16 shows the test set-up with parts labeled.

III. Results

In this section, the results obtained from the three models — Euler-Bernoulli, Simscape and the physical model — are presented and analyzed. First, a comparative analysis of the model's performance and accuracy is conducted. Subsequently, experimental results are shown. Followed by a parametric study, that is carried out to investigate the influence of key parameters on the system's behavior. Lastly, two distinct load-displacement characteristics are examined to highlight unique response features under specific conditions.

A. Comparative analysis

Each model has its own purpose; the Simscape model is an accurate model that does take into account variations in compliant beam length as result of deflection. The higher accuracy comes with a cost; higher computation times can be expected. This is where the Euler-Bernoulli model comes into place; it needs a fraction of the computation time to compute the kinematic solution of the four-bar mechanism, from there results are instantly displayed. Thereby some assumptions have been made; Euler-Bernoulli assumes a beam's neutral plane does not involve compression or tension; and the beam's length is fixed regardless of bending. Finally, the physical model validates the aforementioned models while also taking into account real-world behavior: i.e., damping, friction, manufacturing tolerances and other geometrical imperfections.

For all explored configurations, the Simscape model and Euler-Bernoulli model are in close agreement, as shown in figure 17. The residual plot, confirms the finding and shows deviations that are orders of magnitude smaller than the absolute torques obtained, shown in figure 18. The deviations are attributed to the varying beam length of the compliant coupler link that is not considered in the Euler-Bernoulli model. Besides, shear strain energy is ignored, because the compliant beam is a slender beam.

Experiments have been conducted for two given configurations. Figure 19 shows measurements from four consecutive revolutions in positive direction, followed by four reverse

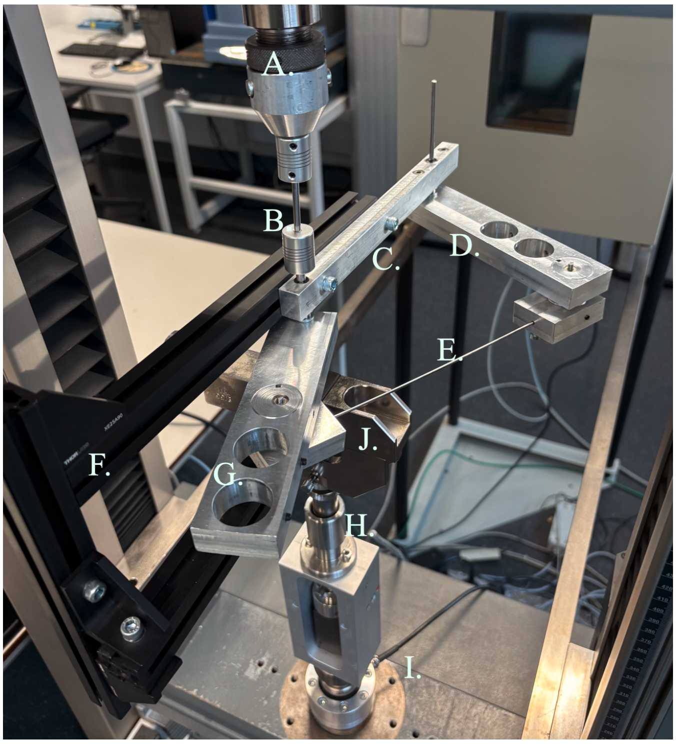


Fig. 16: Photo of experimental set-up. A. Constant angular velocity actuator; B. Coupler link; C. Ground link; D. Output link; E. Compliant coupler link; F. Thorlabs frame structure around mechanism; G. Input link; H. Ball joint that prevents jamming; I. Torque sensor; J. Clamping mechanism (small working bench).

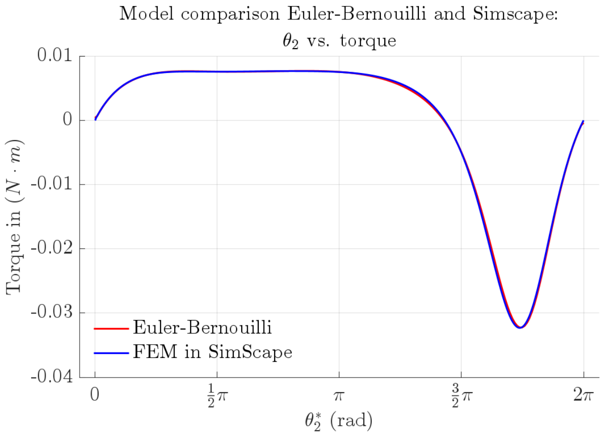


Fig. 17: Model comparison between Simscape and Euler-Bernoulli. For the given configuration, both models parameters are set at $\theta_2^* = \theta_2 - \theta_{2,0}$, where $\theta_{2,0} = 30°$. The azimuth angles are $\phi_{a,0} = 210°$, $\phi_{b,0} = 48°$, the misalignments are $\theta_a = \theta_b = 5°$ and beam lengths are $L_1 = 0.20$ m, $L_2 = 0.08$ m, $L_3 = 0.23$ m and $L_4 = 0.15$ m.

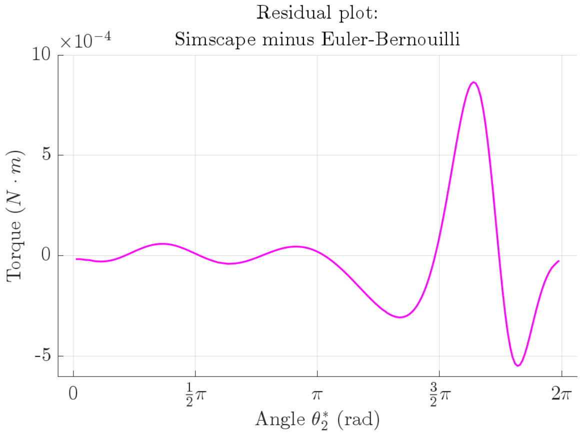


Fig. 18: Residual plot of Simscape torque values minus Euler-Bernoulli torque values over one revolution of θ_2. This residual plot corresponds to the given configuration from figure 17.

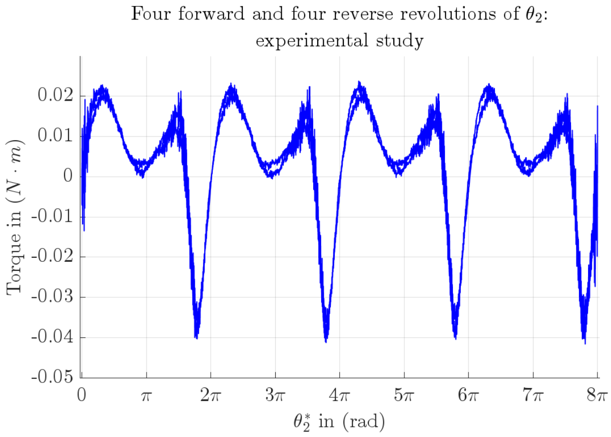


Fig. 19: Experimental results; four revolutions in the positive direction and four revolutions in the negative direction, showcasing hysteresis.

revolutions. The blue line corresponds to the data obtained by the torque sensor. The purpose of measuring four revolutions was to eliminate the effects of acceleration during the startup and stopping phases of the constant velocity driver. Hysteresis can be observed. The noise can be attributed to the excitation of Eigenfrequencies of the four-bar mechanism by the driver, and to sensor noise.

The measurements of the second positive revolution are isolated and plotted on top of the Simscape and Euler-Bernoulli model, shown in Figure 20. To account for fabrication tolerances, deviations in material properties and measurement inaccuracies, parameter relaxation is applied to the Euler-Bernoulli model. This is done using a Monte-Carlo simulation [20]. Relaxation of all parameters was recorder in 50 runs. The region between the upper and lower bounds of the graph represents the bandwidth.

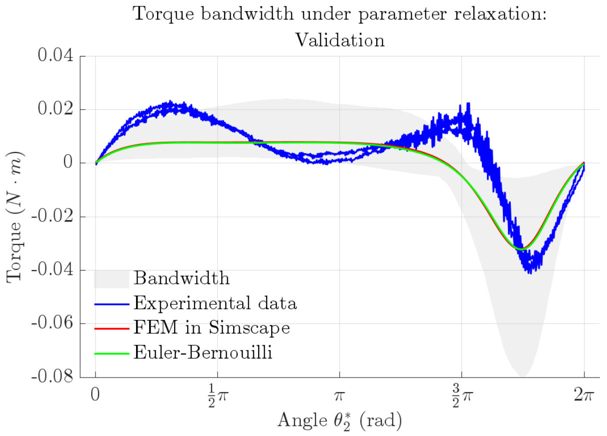


Fig. 20: Parameter relaxation. Young's modulus of 200 ± 5% GPa, Shear modulus of 75 ± 5% GPa, Misalignment of 5° ± 2°, azimuth joint A of 210° ± 10°, azimuth in joint B of 48° ± 10° and $\theta_{2,0}$ of 30° ± 5°. Blue line corresponds to physical experiment including one positive and one negative revolution.

Another given configuration is plotted in figure 21. Again, the Simscape and Euler-Bernoulli model are in close agreement. The experiment has a similar curve; it lies within the bandwidth. Four consecutive rotations of θ_2 where measured, out of which the second revolution is considered.

Up to this point, the four-bar mechanism is considered in an unloaded situation. The compliant coupler link is part of the kinetic chain; forces will be transferred through the compliant beam. As result, the beam will be subjected to compressive or tensile loadings, depending on the configuration. In reality, the beam's tensile strength and compressive strength, are reduced by bending as result of the misalignments. The effect of a loaded output link is modeled in Simscape. By applying a positive torque to the output link, the compliant beam will be subjected to compressive forces. The effect of load is plotted

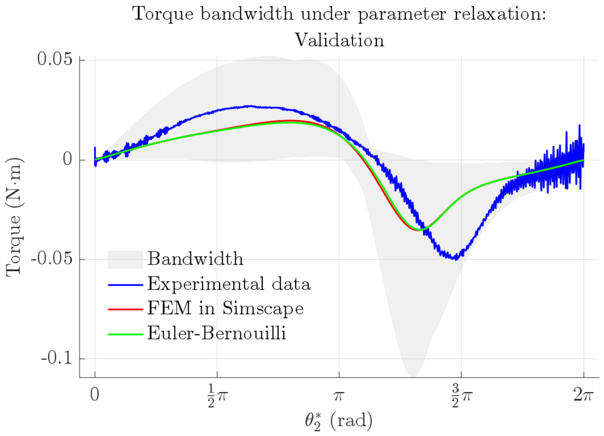


Fig. 21: Parameter relaxation. Young's modulus of 200 ± 5% GPa, Shear modulus of 75 ± 5% GPa, Misalignment of 5° ± 2°, azimuth joint A of 210° ± 10°, azimuth in joint B of 48° ± 10° and $\theta_{2,0}$ of 30° ± 5°. Blue line corresponds to physical experiment including one positive revolution.

in figure 22. The discrepancy between the loaded misaligned model, and the superimposed unloaded aligned and unloaded misaligned models could be attributed to two factors. First, variation in length of the coupler link due to bending, results in a deviated mechanical advantage of the four-bar. This confirms the orange line is positioned above the purple line in figure 23. In addition, due to second-order bending as result of the pre-bent compliant beam, a reduction in axial stiffness is observed [21]. This geometrical nonlinearity is not taken into account by Simscape.

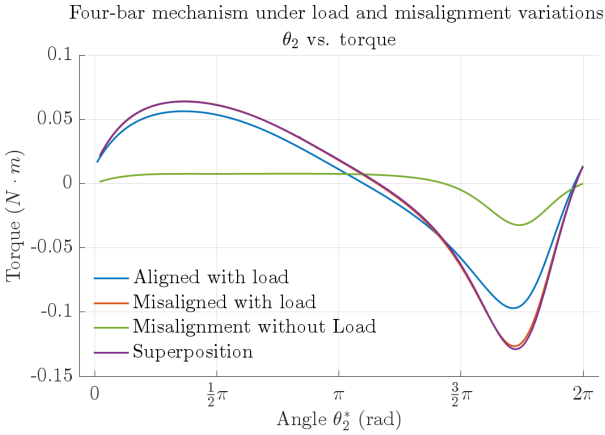


Fig. 22: Loaded and unloaded configuration modeled in Simscape. The blue line represents the four-bar with *aligned* over-constraints and a positive loading of 0.4 Nm. The orange line represents misaligned configuration, with a 0.4 Nm positive loading to the output link. The green line represents misaligned unloaded four-bar mechanism. Purple line is the superposition of the green line with the Blue line.

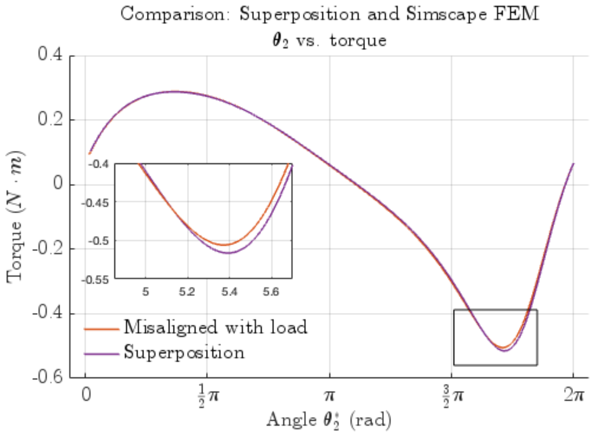


Fig. 23: Simscape comparison between superimposed model (orange) and integrated misaligned and loaded configuration (purple).

B. Parameter study

In this section, the influence of different parameters of the misaligned over-constraint four-bar mechanism on the load-displacement characteristics will be presented. Parameters with a significant influence are; beam lengths and length ratios, compliant beam properties (flexural and torsional rigidity), azimuth and polar angles of both joints, initial stress-free angle of $\theta_{2,0}$ and open versus cross-configuration of the four-bar mechanism. A continuous combination of varying parameters can be chosen. As a result, infinitely many different load-displacement curves can be obtained. To systematically assess the impact of each parameter, its effect on the load-displacement characteristics will be analyzed at multiple discrete values while keeping all other parameters constant. This is done for all parameters except for beam length ratios. Due to the wide variety of possible configurations, a limited number of plots would not be sufficient to comprehensively capture all cases and is therefore disregarded. It can be concluded that the relationship between link lengths determine whether a mechanism functions as a crank-rocker, rocker-crank, double crank or double rocker. Furthermore, it influences a mechanism's mechanical advantage, motion, stability and singularity positions [22]. To investigate the influence of beam length ratios in relation to over-constraint misalignments, several key insights can be derived. In a double crank mechanism, both azimuth angles of the misaligned joints, make a full revolution. In a crank-rocker mechanism, the azimuth angle of the rocker completes only a partial revolution. When the input link reaches zero, the load-displacement behavior is influenced solely by the misalignment in joint A.

*1) **Azimuth angles in joint A and B:*** The azimuth angle represents the horizontal angle of a misalignment in a spherical coordinate system. For both azimuth angles, the values lie within a 2π domain. The azimuth angles in the initial stress-free configuration of joint A and B can be interchanged. These selected values are independent of the selected initial configuration at $\theta_{2,0}$, and do not influence the stress-free initial condition. For five distinct values, evenly spaced over the defined range, the load-displacement characteristics have been plotted. An azimuth of 2π rad equals an azimuth of zero. For misalignment A, the results are shown in figure 24. The azimuth of joint A completes a full revolution, exerting a more significant influence on the load-displacement characteristics compared to the azimuth of joint B, shown in figure 25. The azimuth at joint B does not alter the overall shape of the load-displacement curve, as the number of peaks and valleys remains unchanged. However, their positions along the θ_2^*-axis exhibit slight shifts, and their magnitudes are affected. The valley in between $\frac{3\pi}{2}$ and 2π is almost unaffected.

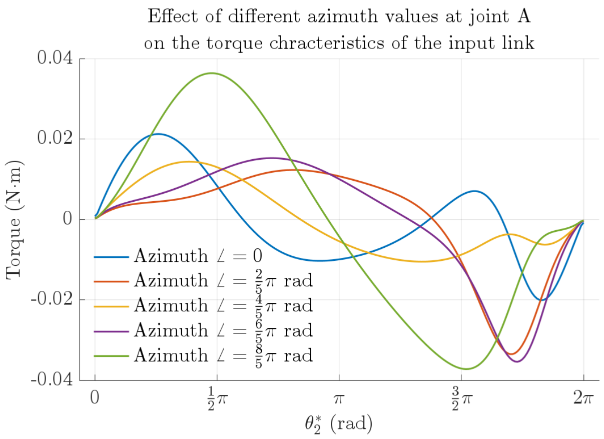


Fig. 24: Azimuth angle of joint A from zero to $\frac{8}{5}\pi$ rad. 2π rad corresponds to 0 rad. Azimuth b is fixed at 0 rad, $\theta_{2,0} = \frac{1}{2}$rad, $L_1 = 0.20$ m, $L_2 = 0.08$ m, $L_3 = 0.23$ m, $L_4 = 0.15$ m

.

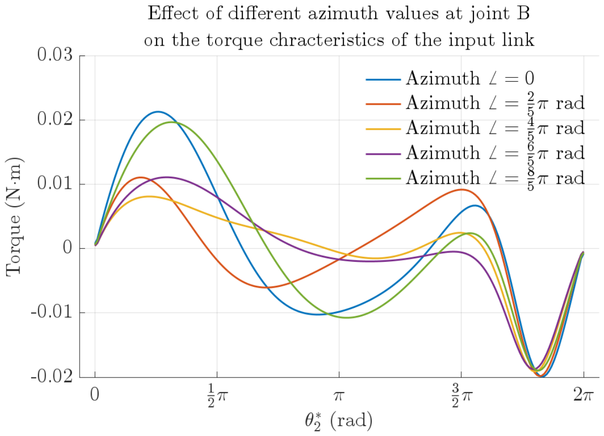


Fig. 25: Azimuth b from zero to $\frac{8}{5}\pi$ rad in 5 steps. Azimuth a = 0 rad, $\theta_{2,0} = \frac{1}{2}$rad, $L_1 = 0.20$ m, $L_2 = 0.08$ m, $L_3 = 0.23$ m, $L_4 = 0.15$ m

.

*2) **Magnitude of misalignment in joint A & B:*** The magnitude of a misalignment is represented by the polar angle; the angle between the positive x_3-axis to the unit position vector, as indicated with θ_a and θ_b in figure 4 and 5. In contrast to the azimuth, the polar angle remains constant throughout a revolution of θ_2. The effect of the magnitude of a misalignment on the load-displacement curve is shown in figure 26. For this plot, the polar angle in joint B remained constant, at 5°. The result shows that a larger polar angle in joint A stretches the load-displacement curve vertically. The location-changes of the crossings between different curves are insignificant compared to the magnitude of polar angle in joint B, shown in figure 27. An increasing polar angle results in a shift to the right of the curve. The shape remains similar; i.e. the number of peaks and valleys remains the same. For this configuration of the four-bar, the influence of the polar angle magnitude in joint B is more pronounced on the left side of the load-displacement curve, whereas the magnitude of the polar angle of joint A has significant influence over the entire curve.

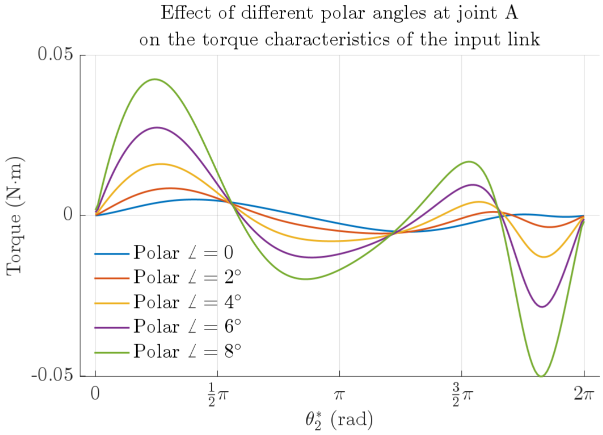


Fig. 26: Polar angle in a in 5 steps: $\theta_a = [0° \;\; 2° \;\; 4° \;\; 6° \;\; 8°]$. $\phi_a = \phi_b = 0$ rad, $\theta_{2,0} = \frac{1}{2}$rad, $L_1 = 0.20$ m, $L_2 = 0.08$ m, $L_3 = 0.23$ m, $L_4 = 0.15$ m. Polar angle of joint B is maintained at 5°.

*3) $\theta_{2,0}$**, in open and cross configuration:*** The initial angle of the input link with respect to the ground link is indicated with $\theta_{2,0}$. In this position, the four-bar is stress free; i.e., the compliant coupler link is in its undeformed state. As shown in figure 2, there is an open and cross configuration of the four-bar. The influence of $\theta_{2,0}$ on the load-displacement curves is shown in figure 28 and 29 for the cross and open configuration respectively. This showcases the significant influence of $\theta_{2,0}$ on the load-displacement characteristics.

*4) **Compliant beam properties:*** Various modifications can be applied to the compliant coupler link to tailor its mechanical properties, thereby influencing the load-displacement response. Adjustments to the cross-sectional geometry alter the second moment of inertia, directly affecting bending stiffness. Additionally, material properties such as Young's modulus, shear modulus, and beam length play a significant role in the overall mechanical behavior. Modifying the cross-sectional shape or employing an anisotropic compliant beam influences the relative contributions of bending and torsion to the load-displacement curve. A cross-section with reduced torsional

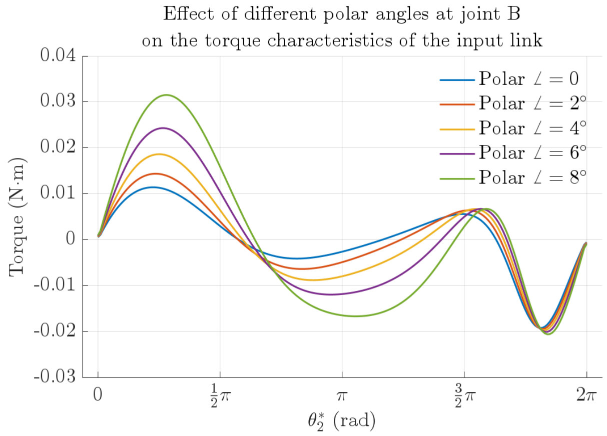


Fig. 27: Polar angle in b in 5 steps θ_b = [0° 2° 5° 8° 10°]. $\phi_a = \phi_b = 0°$, $\theta_{2,0} = \frac{1}{2}$ rad, L_1 = 0.20 m, L_2 = 0.08 m, L_3 = 0.23 m, L_4 = 0.15 m. The polar angle at joint B is maintained at 5°.

.

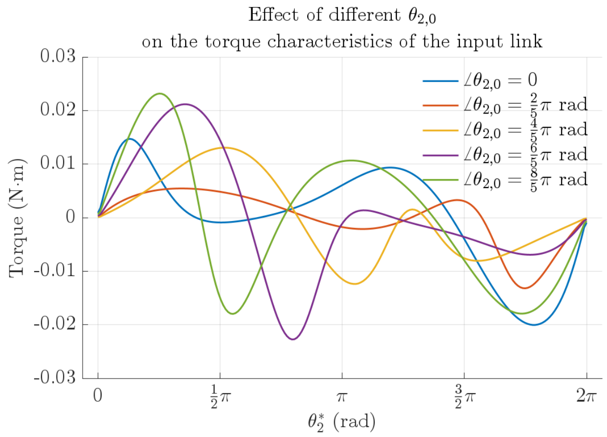


Fig. 28: Varying initial configuration four-bar mechanism in the cross configuration. Polar angles are 5° in joint A & B. Both azimuth angles are maintained at zero radians. L_1 = 0.20 m, L_2 = 0.08 m, L_3 = 0.23 m, L_4 = 0.15 m.

stiffness decreases the torsional load peaks, whereas a stiffer torsional response enhances them. These design modifications enable precise control over the beam's deformation characteristics, optimizing its performance for specific applications. In Figure 30, the beam radius is varied, resulting in vertical stretching of the load-displacement curve. No significant shift in height is observed by peaks corresponding to bending and torsion strain energy, despite the dissimilar scaling of bending and torsion stiffness by an increasing radius as indicated by equation 20.

$$\frac{k_b}{k_t} = \frac{GJ}{EI} = \frac{2G}{E} = 0.75 \tag{20}$$

C. GUI

To discover the wide variety of load-displacement characteristics, a Graphical User Interface (GUI) was programmed.

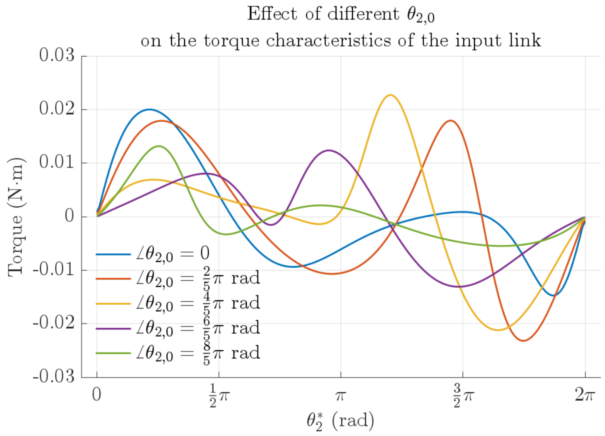


Fig. 29: Varying initial configuration four-bar mechanism in the open configuration. Polar angles are 5° in joint A & B. Both azimuth angles are maintained at zero radians. L_1 = 0.20 m, L_2 = 0.08 m, L_3 = 0.23 m, L_4 = 0.15 m.

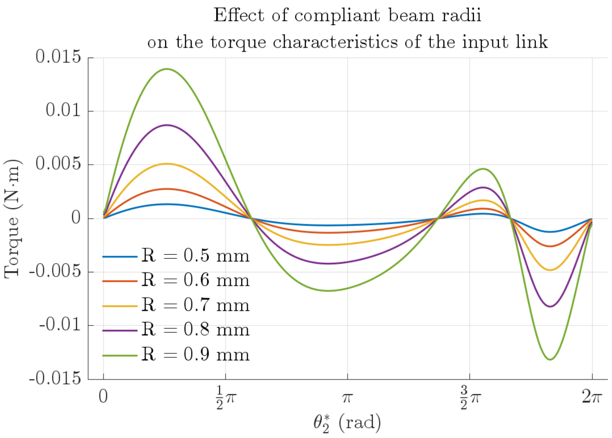
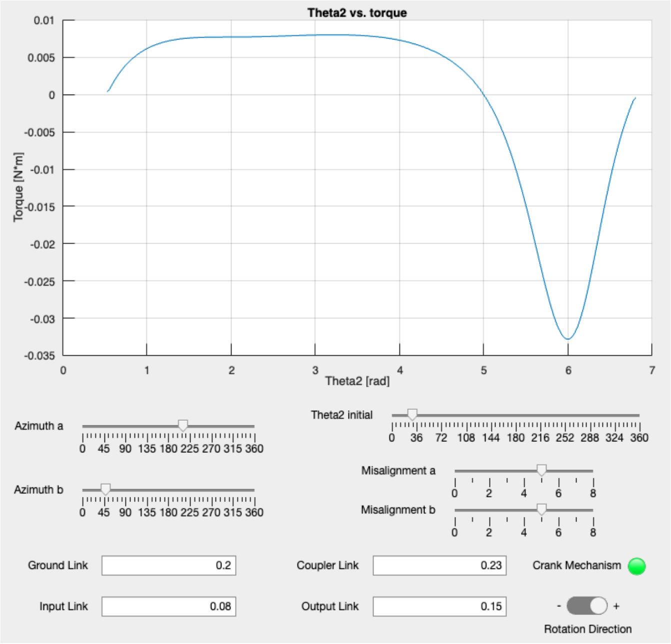


Fig. 30: Compliant beam varying radius: R = [0.5 1.0 1.5] mm. $\phi_a = \phi_b = 0$ rad, $\theta_a = \theta_b = 5°$, $\theta_{2,0} = \frac{1}{2}$ rad, L_1 = 0.20 m, L_2 = 0.08 m, L_3 = 0.23 m, L_4 = 0.15 m

.

This GUI features sliders that allow users to adjust parameters influencing the load-displacement curves of the four-bar mechanism. Figure 31 shows how sliders can be dragged and dropped into the demanded position. Also the beam lengths can be varied as long as the input link is a crank. Otherwise, the LED indicator will turn red. If the input link is a rocker, the input link angle (θ_2) should be bounded to the specific configuration of the four-bar mechanism to ensure proper functionality of the GUI. The GUI provides the option to investigate parameter combinations that would otherwise remain unnoticed. Two unique load-displacement characteristics have been found. A significant region of constant torque and sinusoidal load-displacement curves. These will be elaborated in the next section.



Ground Link 0.2

Input Link 0.08

Coupler Link 0.23

Output Link 0.15

Crank Mechanism ●

- ● +

Fig. 31: Graphical User Interface (GUI) with interchangeable initial values. Azimuth a and azimuth b are the initial values for misalignment starting position for joints a and b respectively. θ_2 is the initial stress-free position of the input link. Misalignment is the polar angle of misalignment. The length of the links can be adjusted. The direction inverts the function in horizontal and vertical direction.

D. *Unique load-displacement characteristics*

The GUI provided the ability to change many parameters simultaneously and see how it influences the load-displacement curves. As a result, two unique load-displacement curves have been found. First, a significant region of constant torque was achieved, for more than one configuration of four-bars. Furthermore, a sinusoidal torque-angle curve is obtained.

1) Constant torque: Constant torque or constant force are unique properties that are often only observed during plastic deformation of materials. In these regions, large displacement can occur at an (almost) constant torque value. It can have many different applications, examples are; power assistance or torque stabilization [23]. A constant-torque configuration, thoroughly analyzed in the previous sections, is illustrated in Figure 13. It shows a region of constant torque for θ_2^* ranging from $\frac{\pi}{4}$ to π. The region of constant torque is a result of the sequence between torsion and bending strain energy, as shown previously in Figure 12.

2) Sine wave torque function: Many industry application involve cyclic loading and unloading of an actuator, i.e. sinusoidal load-displacement characteristics. Numerous sinusoidal-like load-displacement curves have been obtained. In certain cases, segments of the curves exhibit a nearly perfect sinusoidal shape, as shown in Figure 32. Besides, a sinusoidal curve can be observed when the input-to-output link ratio is such that the input link is significantly shorter than the output link, as shown in figure 33.

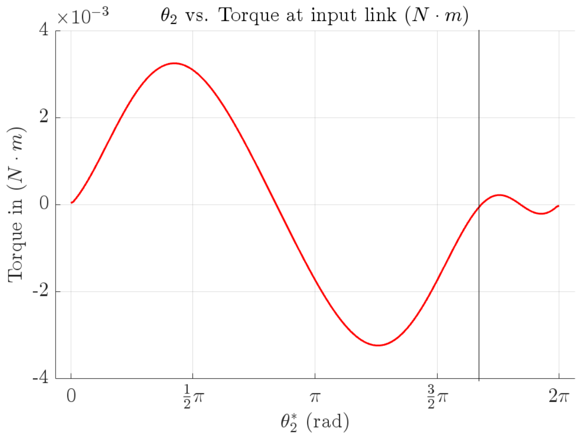


Fig. 32: Partially sinusoidal load-displacement function. $\theta_2^* = \theta_2 - \theta_{2,0}$, where $\theta_{2,0} = \frac{1}{2}$ rad. The azimuth angles are $\phi_{a,0} = 5.8$ rad, $\phi_{b,0} = 3$ rad, the mi salignments are $\theta_a = 0$, $\theta_b = 4°$ and beam lengths are $L_1 = 0.20$ m, $L_2 = 0.08$ m, $L_3 = 0.23$ m and $L_4 = 0.15$ m.

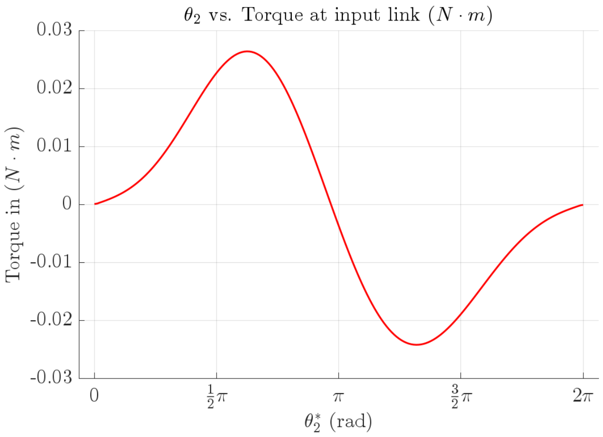


Fig. 33: Sinusoidal load-displacement function as result of short input link. $\theta_2^* = \theta_2 - \theta_{2,0}$, where $\theta_{2,0} = 4.7$ rad. The azimuth angles are $\phi_{a,0} = 4.7$ rad, $\phi_{b,0} = 4.7$ rad, the misalignments are $\theta_a = 5°$, $\theta_b = 5°$ and beam lengths are $L_1 = 0.20$ m, $L_2 = 0.01$ m, $L_3 = 0.23$ m and $L_4 = 0.15$ m.

IV. Discussion

This research contributes to the design and analysis of tailorable, nonlinear compliant mechanisms. Several aspects of the study could be enhanced, beginning with refinements to the physical model. First, the 2 mm diameter compliant coupler link is force-closed to the joints rather than shape-closed, limiting the use of a thicker compliant beam to improved highlighting of load-displacement characteristics from external influences as bearing friction and gravity. Second, an improved reading of the azimuth angles should be implemented, to improve configuration precision. Finally, an experiment with a load applied to the output link to see where the beam would buckle, and where geometric stiffening/softening occurs, is deemed to have significant added value to this research. Besides improvements to the physical model, both computer models could be enhanced too. First would be to include stress simulations of the compliant beam, to showcase at what misalignment plastic deformation would occur. Furthermore, an effort to geometric stiffening/softening as result of the curved compliant beam could be implemented in the Euler-Bernoulli model to include the load-displacement characteristics once the system is subjected to external loading. This would have significant added value, because it would have the same performance of the current model, without the simulation time needed for a finite element analysis (FEA). Finally, to verify the enhanced Euler-Bernoulli model, an FEA that does take into account geometric nonlinearities would have signficant added value. In addition to potential improvements, this study could serve as a foundation for further research. Compliant joints constrain motion in specific directions; therefore, they could also function as misaligned over-constraints. The replacement of traditional ball bearings with compliant joints presents a promising avenue for further exploration. Additionally, optimizing the load-displacement characteristics to meet specific requirements — such as achieving a plateau of constant force — could enhance their applicability. This optimization could lead to advancements in various applications, including suspension systems, mass-balancing mechanisms, and the prevention of singularities through misaligned over-constraints. Next, implicit assumptions — such as having only the coupler link be compliant, the specific joints that are misaligned, an initially straight coupler link, and the use of a circular and constant cross-section — could be relaxed and investigated further. Furthermore, the integration of misaligned over-constraints into other over-constraint mechanisms, such as the Sarrus mechanism, may be of interest. Looking further ahead, the implementation of this technology in unit cells of mechanical metamaterials represents a potential area for future research.

V. Conclusion

Misaligned over-constraints and precision engineering may seem contradictory; however, this study demonstrates the diverse load-displacement characteristics that can be achieved through misaligned over-constraints in four-bar mechanisms. While only a subset of these possibilities has been explored, this paper serves as a foundation for engineers seeking novel load-displacement behaviors in mechanisms. Notably, two distinct characteristics were observed: a region of nearly constant force and a sinusoidal response, both of which have promising applications in constant torque transmission and precision motion control.

The Simscape and Euler-Bernoulli beam models exhibit strong agreement in the unloaded state. However, under external loading, the Simscape model provides a more accurate representation due to fewer simplifying assumptions, albeit at the cost of increased computation time. The complementary use of both models enhances the identification of desired load-displacement characteristics by leveraging their respective strengths. Future improvements could incorporate geometric stiffening and softening effects induced by the curved compliant coupler link to the Euler-Bernoulli model, allowing for a more precise response to external loads. Additionally, the Simscape model could be enhanced by integrating finite element analysis (FEA) that accounts for geometric nonlinearities, enabling the inclusion of buckling behavior and geometric stiffening/softening.

Experimental results were largely within the margin of error, reinforcing the validity of the models and highlighting the impact of real-world behavior on the system.

More broadly, four-bar mechanisms with misaligned over-constraints present a promising approach for passively balancing masses, reducing reliance on high-torque actuators. As a result, typically essential components such as springs and dampers could be minimized or eliminated in specific applications, offering a more efficient and lightweight design alternative.

References

[1] J.-S. Zhao, K. Zhou, and Z.-J. Feng, "A theory of degrees of freedom for mechanisms," *Mechanism and Machine Theory*, vol. 39, no. 6, pp. 621–643, 2004.

[2] G. Gogu, "Chebychev–grübler–kutzbach's criterion for mobility calculation of multi-loop mechanisms revisited via theory of linear transformations," *European Journal of Mechanics-A/Solids*, vol. 24, no. 3, pp. 427–441, 2005.

[3] M. Grübler, *Allgemeine Eigenschaften der zwangläufigen ebenen kinematischen Ketten*. L. Simion, 1884.

[4] W. W. van de Sande, R. G. Aarts, and D. M. Brouwer, "Effects of misalignments on the static and dynamic behaviour of a multiple overconstrained compliant 4-bar mechanism," *Precision Engineering*, vol. 60, pp. 143–151, 2019.

[5] M. Nijenhuis, J. Meijaard, and D. Brouwer, "Misalignments in an overconstrained flexure mechanism: A cross-hinge stiffness investigation," *Precision engineering*, vol. 62, pp. 181–195, 2020.

[6] D. M. Brouwer, S. Boer, J. P. Meijaard, and R. G. Aarts, "Optimization of release locations for small self-stress large stiffness flexure mechanisms," *Mechanism and machine theory*, vol. 64, pp. 230–250, 2013.

[7] W. W. Van De Sande, R. G. Aarts, and D. M. Brouwer, "System behaviour of a multiple overconstrained compliant four-bar mechanism," in *30th ASPE Annual Meeting 2015*. American Society for Precision Engineering, 2015, pp. 95–99.

[8] S. Liu, G. Peng, Z. Li, W. Li, and K. Jin, "Nonlinear stiffness analysis and programming of a composite origami metamaterial with embedded joint-type metastructures," *Composite Structures*, vol. 310, p. 116761, 2023.

[9] X. Jing, "The x-structure/mechanism approach to beneficial nonlinear design in engineering," *Applied Mathematics and Mechanics*, vol. 43, no. 7, pp. 979–1000, 2022.

[10] X. Jing, Y. Chai, X. Chao, and J. Bian, "In-situ adjustable nonlinear passive stiffness using x-shaped mechanisms," *Mechanical Systems and Signal Processing*, vol. 170, p. 108267, 2022.

[11] T. Wongratanaphisan and M. O. Cole, "Analysis of a gravity compensated four-bar linkage mechanism with linear spring suspension," 2008.

[12] G. J. van den Doel, J. L. Herder, and D. Farhadi, "Harnessing elastic energy to overcome singularity issues in four-bar mechanisms with a crank link," *Mechanism and Machine Theory*, vol. 183, p. 105274, 2023.

[13] C. R. Barker, "A complete classification of planar four-bar linkages," *Mechanism and Machine Theory*, vol. 20, no. 6, pp. 535–554, 1985.

[14] R. Hibbeler, *Mechanics of Materials, eBook, SI Edition*. Pearson Education, 2023. [Online]. Available: https://books.google.nl/books?id=7ZrMEAAAQBAJ

[15] F. Freudenstein, *Design of four-link mechanisms*. Columbia University, 1954.

[16] A. Ghosal, “The freudenstein equation: Design of four-link mechanisms,” *Resonance*, vol. 15, pp. 699–710, 2010.

[17] X. Yang, Y. Xiang, Y.-F. Luo, X.-N. Guo, and J. Liu, “Axial compression capacity of steel circular tube with large initial curvature: Column curve and application in structural assessment,” *Journal of Constructional Steel Research*, vol. 177, p. 106481, 2021.

[18] Z. Kala, “Sensitivity assessment of steel members under compression,” *Engineering Structures*, vol. 31, no. 6, pp. 1344–1348, 2009.

[19] C. E. Volle, “Simscape modeling verification in the simulink development environment,” Tech. Rep., 2014.

[20] R. L. Harrison, “Introduction to monte carlo simulation,” in *AIP conference proceedings*, vol. 1204, 2010, p. 17.

[21] R. Hibbeler, *Mechanics of Materials, eBook, SI Edition*. Pearson Education, 2023. [Online]. Available: https://books.google.nl/books?id=7ZrMEAAAQBAJ

[22] N. Sclater and N. P. Chironis, *Mechanisms and mechanical devices sourcebook*. Mcgraw-hill New York, 2001, vol. 3.

[23] E. Saerens, R. Furnémont, J. Legrand, K. Langlois, P. López García, S. Crispel, M. Rossini, T. Verstraten, B. Vanderborght, and D. Lefeber, “Constant torque mechanisms: A survey,” *Applied Mechanics Reviews*, vol. 74, no. 1, p. 010802, 2022.

3 Conclusion

This report presents a comprehensive study on the use of misaligned over-constraints in four-bar mechanisms with a compliant coupler link. The work combined theoretical modeling, numerical simulation and experimental validation. The main body of work is included in the form of a research paper (Chapter 2), which details the concept, methodology and findings.

In a broader context, this project demonstrates the potential of strategical positioning of misaligned over-constraints as a functional design tool for achieving tailorable load–displacement characteristics in a wider range of mechanisms. As a result, it offers an alternative to the use of conventional elements in mechanisms, such as springs, damper and (strong) actuators.

The development of the GUI in MATLAB's app designer added value by enabling rapid exploration of design parameters, which may be useful for future design studies. While this study focuses on a single compliant link with two misaligned over-constraints, future work could extend the approach to other types of mechanisms. Possible directions include the integration of compliant joints, increasing the number of compliant links, incorporating variable or non-circular cross-sections to the compliant link(s), considering geometric nonlinearities of the compliant link, and exploring initially curved compliant links. Additionally, the functionality of the GUI could be expanded, and optimization strategies could be introduced to tailor the mechanism's response to specific load–displacement requirements.

Beyond the nonlinear load-displacement characteristics of four-bar mechanisms, this report highlights the nonlinear progression of the design process, where adjustments, experimentation and reflection played a crucial role. Overall, this project highlights a promising direction for tailorable load-displacement characteristics of mechanisms through intelligent misaligned over-constraint placement. This works invites further exploration into the design space of compliant, misaligned over-constraint mechanisms.

A

Appendix A - Euler-Bernoulli beam theory in MATLAB

```

%%%% --- SOLVE VECTOR LOOP EQUATIONS NUMERICALLY --- %%%%%%

% Parameters
E = 200e9;                      % Young's modulus, Pa (Steel)
d = 2e-3;                       % Diameter of compliant beam in [m]
I = 1/64*pi*d^4;                % Moment of inertia for bending, m^4
J = 1/32*pi*d^4;                % Moment of inertia for torsion, m^4
G = 77e9;                       % Shear modulus
L1 = 0.2;                       % Ground link length
L2 = 0.01;                      % Input link length
L3 = 0.23;                      % Coupler link length
L4 = 0.15;                      % Output link length
theta2_initial = deg2rad(270);  % Initial angle of theta2
misalignment_a = deg2rad(5);    % Polar angle joint A
misalignment_b = deg2rad(5);    % Polar angle joint B
manual_azimuth_a = deg2rad(270);% Adjustable azimuth angle of misalignment A
manual_azimuth_b = deg2rad(270);% Adjustable azimuth angle of misalignment B
steps = 200;                    % Number of steps

% Generate theta2 array
theta2 = linspace(theta2_initial, (theta2_initial + 2*pi), steps);
theta2_star = theta2 -theta2_initial;

% Preallocate arrays for theta3 and theta4
theta3 = zeros(1, steps);
theta4 = zeros(1, steps);

% Define the vector loop equation
loop_equation = @(x, theta2_val) [
    L2*cos(theta2_val) + L3*cos(x(1)) - L4*cos(x(2)) - L1;
    L2*sin(theta2_val) + L3*sin(x(1)) - L4*sin(x(2));
];

x3 = L2*cos(theta2_initial); % x-coordinate of coupler
y3 = L2*sin(theta2_initial); % y-coordinate of coupler
theta3_guess = deg2rad(30);  % Approximate theta3, modify for open/cross config.
theta4_guess = deg2rad(89);  % Approximate theta4, Modify for open/cross config.
x0 = [theta3_guess, theta4_guess]; % Initial guess for fsolve

options = optimoptions('fsolve','Display','iter');

% Solve for each theta2 value
for i = 1:steps
    theta2_val = theta2(i);
    solution = fsolve(@(x) loop_equation(x, theta2_val), x0, options);
    theta3(i) = solution(1);
    theta4(i) = solution(2);
    x0 = solution; % Use the previous solution as the next initial guess
end

% Get the initial angles for theta3 and theta4
theta3_initial = theta3(1);
theta4_initial = theta4(1);


%%%%%%% ---- CREATE ARRAYS OF AZIMUTH ANGLES NEEDED FOR BENDING AND TORSION FUNCTIONS ---- %%%%%%%
% Mechanism the joints connected to compliant link are 'a' and 'b'.

% In joint a: Angle between L2 and L3 is calculated at each position of
% theta2 ranging from [theta2_initial, theta2_initial + 2*pi]. This angle
% will be the azimuth angle of the misalignment in 'a' corresponding to theta2
% with the offset that ensures zero initial position (perpendicular to complaint beam).
azimuth_a_offset = (pi - theta2_initial + theta3_initial);
azimuth_a = -(pi - theta2 + theta3) + azimuth_a_offset + manual_azimuth_a;

% In joint b: Angle between L3 and L4 is calculated at each position of
% theta2 ranging from [theta2_initial, theta2_initial + 2*pi]. This angle
% will be the azimuth angle of the misalignment in 'b' corresponding to theta2.
azimuth_b_offset = (pi - theta4_initial + theta3_initial);
azimuth_b = -(pi - theta4 + theta3) + azimuth_b_offset + manual_azimuth_b;


%%%%%% ---- BENDING IN JOINT A AND B ---- %%%%%%%%
% Bending angle of a and b stored in an array. Calculated as the projection
% of the spherical coordinate system on the YZ plane.
% Bending angle in 'a'. Should start at zero due to offset.
bending_a_offset = atan2((sin(misalignment_a) * sin(manual_azimuth_a)), cos(misalignment_a));
bending_a = -atan2((sin(misalignment_a) * sin(azimuth_a)), (cos(misalignment_a))) + bending_a_offset;

% Bending angle in b as opposite minus sign due to chosen orientation of
% spherical coordinate system.
bending_b_offset = atan2((sin(misalignment_b) * sin(manual_azimuth_b)), cos(misalignment_b));
bending_b = -atan2((sin(misalignment_b) * sin(azimuth_b)), (cos(misalignment_b))) + bending_b_offset;

%From bending angles in a and b, the moments can be calculated and
%subsequently the bending energy
bending_energy = zeros(1, steps);
Ma_values = zeros(1, steps);
Mb_values = zeros(1, steps);

for i = 1: steps
    %Calculate Ma and Mb at each step
    Ma = -2*E*I*(2*bending_a(i) + bending_b(i))/L3;
    Mb = 2*E*I*(bending_a(i) + 2*bending_b(i))/L3;

    Ma_values(i) = Ma;
    Mb_values(i) = Mb;

    %The moment function:
    M = @(x) ((Mb - Ma) / L3) * x + Ma;

    %Integrate the squared moment function from 0 to L3
    bending_energy(i) = (1 / (2 * E * I)) * integral(@(x) M(x).^2, 0, L3);
end

%%% PLOT MA AND MB
hfig = figure;
plot(theta2_star, Ma_values, 'color',[0 0.5 0.5], 'DisplayName', 'Bending␣moment␣in␣joint␣A', 'LineWidth', 1.5);
hold on;
plot(theta2_star, Mb_values, 'color',[0.85 0.33 0.1],'DisplayName', 'Bending␣moment␣in␣joint␣B', 'LineWidth', 1.5);

```

```
xticks(0:pi/2:2*pi); % Set tick marks from 0 to 2*pi in steps of pi/2
xticklabels({'$0$', '$\frac{1}{2}\pi$', '$\pi$', '$\frac{3}{2}\pi$', '$2\pi$'}); % Label the tick marks

xlabel('$\theta_2^*$␣(rad)');
ylabel('Bending␣moment␣$M_a$␣and␣$M_b$␣($N␣\cdot␣m$)');
% legend('Ma', 'Mb');
title('$\theta_2$␣vs.␣Bending␣moment␣induced␣by␣joint␣A␣and␣B');
grid on;
legend('show', 'Location', 'northwest', 'Interpreter', 'latex');

xlim([-0.2 2*pi+0.2]); % Adds a small buffer before the first tick at 0
ax = gca;

% xlabel('Angle $\theta_2$ (rad)');
% ylabel('Torque (N$\cdot$m)');
% title('Torque vs. $\theta_2$');
fname = 'Bending␣moments␣in␣a␣and␣b,␣theta2_star';
grid on;

picturewidth = 20; % set this parameter and keep it forever
hw_ratio = 0.65; % feel free to play with this ratio
set(findall(hfig,'-property','FontSize'),'FontSize',17) % adjust fontsize to your document
set(findall(hfig,'-property','Box'),'Box','off') % optional
set(findall(hfig,'-property','Interpreter'),'Interpreter','latex')
set(findall(hfig,'-property','TickLabelInterpreter'),'TickLabelInterpreter','latex')
set(hfig,'Units','centimeters','Position',[3 3 picturewidth hw_ratio*picturewidth])
pos = get(hfig,'Position');
set(hfig,'PaperPositionMode','Auto','PaperUnits','centimeters','PaperSize',[pos(3), pos(4)])
% print(hfig,fname,'-dpdf','-painters','-fillpage')
print(hfig,fname,'-dpng','-painters')


%%%%%% ---- TORSION IN COMPLIANT BEAM ---- %%%%%%%%
torsion_a_offset = atan2((sin(misalignment_a) * cos(manual_azimuth_a)), cos(misalignment_a));
torsion_a = atan2((sin(misalignment_a) * cos(azimuth_a)), (cos(misalignment_a))) - torsion_a_offset;

torsion_b_offset = atan2((sin(misalignment_b) * cos(manual_azimuth_b)), cos(misalignment_b));
torsion_b = atan2((sin(misalignment_b) * cos(azimuth_b)), (cos(misalignment_b))) - torsion_b_offset;

% Combined torsion angle for Torsion formula
relative_torsion = (torsion_b - torsion_a);


% Compute torsion as a vector
torsion = relative_torsion .* J .* G ./ L3;

% Compute torsion energy as a vector
torsion_energy = (torsion.^2 .* L3) ./ (2 .* G .* J);


%% PLOTS OF TORSION
hfig = figure;
plot(theta2_star, torsion_a, 'color',[0.3 0.6 0], 'DisplayName', '$\phi_{T,a}$', 'LineWidth', 1.5);
hold on;
plot(theta2_star, torsion_b, 'color', [0.7 0.1 0.4],'DisplayName', '$\phi_{T,b}$', 'LineWidth', 1.5);
hold on;
plot(theta2_star, relative_torsion, 'color', [0.3 0.3 0.7], 'DisplayName', '$\phi_T$', 'LineWidth', 2);
xlabel('$\theta_2^*$␣(rad)');
ylabel('Torsion␣angle␣(rad)');
% legend('torsion angle a', 'torsion angle b', 'relative torsion angle');
title(['Torsion␣analysis:␣$\theta_2$␣vs.␣Angles␣of␣position␣vectors' newline...
    'projected␣on␣the␣$x_3$-$x_1$-plane']);
grid on;

xticks(0:pi/2:2*pi); % Set tick marks from 0 to 2*pi in steps of pi/2
```

```
xticklabels({'$0$', '$\frac{1}{2}\pi$', '$\pi$', '$\frac{3}{2}\pi$', '$2\pi$'}); % Label the tick marks

xlim([-0.2 2*pi+0.2]); % Adds a small buffer before the first tick at 0
ax = gca;

legend('show', 'Location', 'northwest', 'Interpreter', 'latex');
h = legend; % Store legend handle
% set(h, 'Position', [0.43, 0.4, 0.2, 0.1]); % Manually adjust position
% xlabel('Angle $\theta_2$ (rad)');
% ylabel('Torque (N$\cdot$m)');
% title('Torque vs. $\theta_2$');
fname = 'Torsion␣angles';
grid on;

picturewidth = 20; % set this parameter and keep it forever
hw_ratio = 0.65; % feel free to play with this ratio
set(findall(hfig,'-property','FontSize'),'FontSize',17) % adjust fontsize to your document
set(findall(hfig,'-property','Box'),'Box','off') % optional
set(findall(hfig,'-property','Interpreter'),'Interpreter','latex')
set(findall(hfig,'-property','TickLabelInterpreter'),'TickLabelInterpreter','latex')
set(hfig,'Units','centimeters','Position',[3 3 picturewidth hw_ratio*picturewidth])
pos = get(hfig,'Position');
set(hfig,'PaperPositionMode','Auto','PaperUnits','centimeters','PaperSize',[pos(3), pos(4)])
% print(hfig,fname,'-dpdf','-painters','-fillpage')
print(hfig,fname,'-dpng','-painters')




%%

%%%%%%%%%% ----- COMBINED BENDING ENERGY AND TORSION ENERGY ----- %%%%%%%

total_energy = bending_energy + torsion_energy;

% Calcute the derivative of total energy wrt theta2
d_total_energy_d_theta2 = gradient(total_energy, theta2);

% Plot total energy in compliant beam
hfig = figure;
plot(theta2_star, total_energy, 'color',[0 0.7 0.7], 'DisplayName','Total␣strain␣energy' ,'LineWidth', 2);
hold on;
plot(theta2_star, bending_energy, 'color',[0.6 0.6 0], 'DisplayName', 'Bending␣strain␣energy', 'LineWidth', 1.5);
hold on;
plot(theta2_star, torsion_energy, 'color',[0.7 0 0.7], 'DisplayName', 'Torsion␣strain␣energy', 'LineWidth', 1.5);
xlabel('$\theta_2^*$␣(rad)');
ylabel('Total␣strain␣energy␣$U_{\mathrm{tot}}(\theta_2)$␣in␣(J)', 'Interpreter', 'latex');
% legend('Total strain energy', 'Bending strain energy', 'Torsion strain energy');
% title('Total strain energy (Bending + Torsion) Vs. Input Angle \theta_2');
grid on


xticks(0:pi/2:2*pi); % Set tick marks from 0 to 2*pi in steps of pi/2
xticklabels({'$0$', '$\frac{1}{2}\pi$', '$\pi$', '$\frac{3}{2}\pi$', '$2\pi$'}); % Label the tick marks

xlim([-0.2 2*pi+0.2]); % Adds a small buffer before the first tick at 0
ax = gca;


legend('show', 'Location', 'northwest', 'Interpreter', 'latex');
% xlabel('Angle $\theta_2$ (rad)');
% ylabel('Torque (N$\cdot$m)');
title('$\theta_2$␣vs.␣Total␣strain␣energy␣(Bending␣+␣Torsion)');
fname = 'energy␣comparison';
grid on;

```

```matlab
picturewidth = 20; % set this parameter and keep it forever
hw_ratio = 0.65; % feel free to play with this ratio
set(findall(hfig,'-property','FontSize'),'FontSize',17) % adjust fontsize to your document
set(findall(hfig,'-property','Box'),'Box','off') % optional
set(findall(hfig,'-property','Interpreter'),'Interpreter','latex')
set(findall(hfig,'-property','TickLabelInterpreter'),'TickLabelInterpreter','latex')
set(hfig,'Units','centimeters','Position',[3 3 picturewidth hw_ratio*picturewidth])
pos = get(hfig,'Position');
set(hfig,'PaperPositionMode','Auto','PaperUnits','centimeters','PaperSize',[pos(3), pos(4)])
% print(hfig,fname,'-dpdf','-painters','-fillpage')
print(hfig,fname,'-dpng','-painters')



%% Constant torque profile d_total_energy_d_theta2


% Plot the derivative of Energy function to obtain torque
hfig = figure;
plot(theta2_star, d_total_energy_d_theta2, 'Color', 'r', 'LineWidth', 1.5);

xlabel('$\theta_2^*$␣(rad)');
ylabel('Torque␣in␣$(N␣\cdot␣m)$');
grid on

xticks(0:pi/2:2*pi); % Set tick marks from 0 to 2*pi in steps of pi/2
xticklabels({'$0$', '$\frac{1}{2}\pi$', '$\pi$', '$\frac{3}{2}\pi$', '$2\pi$'}); % Label the tick marks

xlim([-0.2 2*pi+0.2]); % Adds a small buffer before the first tick at 0
ax = gca;


% legend('show', 'Location', 'best', 'Interpreter', 'latex');
% xlabel('Angle $\theta_2$ (rad)');
% ylabel('Torque (N$\cdot$m)');
title('$\theta_2$␣vs.␣Torque␣at␣input␣link␣$(N␣\cdot␣m)$');
fname = 'Sinus␣V3';
grid on;

picturewidth = 20; % set this parameter and keep it forever
hw_ratio = 0.65; % feel free to play with this ratio
set(findall(hfig,'-property','FontSize'),'FontSize',17) % adjust fontsize to your document
set(findall(hfig,'-property','Box'),'Box','off') % optional
set(findall(hfig,'-property','Interpreter'),'Interpreter','latex')
set(findall(hfig,'-property','TickLabelInterpreter'),'TickLabelInterpreter','latex')
set(hfig,'Units','centimeters','Position',[3 3 picturewidth hw_ratio*picturewidth])
pos = get(hfig,'Position');
set(hfig,'PaperPositionMode','Auto','PaperUnits','centimeters','PaperSize',[pos(3), pos(4)])
% print(hfig,fname,'-dpdf','-painters','-fillpage')
print(hfig,fname,'-dpng','-painters')
% xlabel('\theta_2 (rad)');
% ylabel('Torque in N*m');






%% Plot FEM and Euler-Bernouilli beam theory results


% Plot the derivative of Energy function to obtain torque
hfig = figure;
plot(theta2_star, d_total_energy_d_theta2, 'Color', 'r', 'LineWidth', 1.5, 'DisplayName','Euler-Bernouilli');
hold on
plot(x_trimmed-deg2rad(30), y_trimmed, 'b', 'LineWidth', 1.5, 'DisplayName','FEM␣in␣SimScape');

xlabel('$\theta_2^*$␣(rad)');
```

```matlab
ylabel('Torque␣in␣$(N␣\cdot␣m)$');
grid on

xticks(0:pi/2:2*pi); % Set tick marks from 0 to 2*pi in steps of pi/2
xticklabels({'$0$', '$\frac{1}{2}\pi$', '$\pi$', '$\frac{3}{2}\pi$', '$2\pi$'}); % Label the tick marks

xlim([-0.2 2*pi+0.2]); % Adds a small buffer before the first tick at 0
ax = gca;

legend('show', 'Location', 'southwest', 'Interpreter', 'latex');
title({'Model␣comparison␣Euler-Bernouilli␣and␣Simscape:'},{'$\theta_2$␣vs.␣torque'});
fname = 'Model-Comparison';
grid on;

picturewidth = 20; % set this parameter and keep it forever
hw_ratio = 0.65; % feel free to play with this ratio
set(findall(hfig,'-property','FontSize'),'FontSize',17) % adjust fontsize to your document
set(findall(hfig,'-property','Box'),'Box','off') % optional
set(findall(hfig,'-property','Interpreter'),'Interpreter','latex')
set(findall(hfig,'-property','TickLabelInterpreter'),'TickLabelInterpreter','latex')
set(hfig,'Units','centimeters','Position',[3 3 picturewidth hw_ratio*picturewidth])
pos = get(hfig,'Position');
set(hfig,'PaperPositionMode','Auto','PaperUnits','centimeters','PaperSize',[pos(3), pos(4)])
% print(hfig,fname,'-dpdf','-painters','-fillpage')
print(hfig,fname,'-dpng','-painters')
% xlabel('\theta_2 (rad)');
% ylabel('Torque in N*m');

%%
ts_interp = interp1(ts_angle, theta2_star, 'linear');

%x2_scale = linspace(1.51844, 7.80162, )
p = 4; % Percentage of data to hide (e.g., 10% hidden)
startIndex = round(p/100 * length(ts_interp));

hfig = figure;  % save the figure handle in a variable
%t = 0:0.02:10; x = t.*sin(2*pi*t)+ 2*rand(1,length(t)); % data
plot(theta2_star, d_total_energy_d_theta2,'Color',[1 0.38 0.53],'LineWidth',1.5,'DisplayName','Euler-Bernouilli');
hold on
plot(theta2_star, ts_interp, 'Color', [0, 0.7 0.7], 'LineWidth', 1.5, 'DisplayName', 'FEM␣in␣SimScape');
hold on
% plot(fliplr(x2_subset), y2_subset, 'Color',[1, 0.5, 0], 'LineWidth',1.5, 'DisplayName','Physical Experiment')
% plot(ts_angle, ts_torque, 'Color', [0, 0.7 0.7], 'LineWidth', 1.5, 'DisplayName', 'FEM in SimScape');
legend('show', 'Location', 'southwest', 'Interpreter', 'latex');
xlabel('Angle␣$\theta_2$␣(rad)');
ylabel('Torque␣(N$\cdot$m)');
title('Torque␣vs.␣$\theta_2$');
fname = 'comparitive_methods';
grid on;

picturewidth = 20; % set this parameter and keep it forever
hw_ratio = 0.65; % feel free to play with this ratio
set(findall(hfig,'-property','FontSize'),'FontSize',17) % adjust fontsize to your document
set(findall(hfig,'-property','Box'),'Box','off') % optional
set(findall(hfig,'-property','Interpreter'),'Interpreter','latex')
set(findall(hfig,'-property','TickLabelInterpreter'),'TickLabelInterpreter','latex')
set(hfig,'Units','centimeters','Position',[3 3 picturewidth hw_ratio*picturewidth])
pos = get(hfig,'Position');
set(hfig,'PaperPositionMode','Auto','PaperUnits','centimeters','PaperSize',[pos(3), pos(4)])
% print(hfig,fname,'-dpdf','-painters','-fillpage')
print(hfig,fname,'-dpng','-painters')


%% MONTE CARLO SAMPLING

num_samples = 50; % Number of Monte Carlo samples
```

```

% Preallocate storage for results
d_total_energy_d_theta2_samples = zeros(num_samples, steps);

% Define parameter variation ranges
theta2_initial_range = deg2rad([30-5, 30+5]);
misalignment_a_range = deg2rad([5-3, 5+3]);
misalignment_b_range = deg2rad([5-3, 5+3]);
E_range = [0.95*E, 1.05*E];
G_range = [0.95*G, 1.05*G];
manual_azimuth_a_range = deg2rad([210-10, 210+10]);
manual_azimuth_b_range = deg2rad([48-10, 48+10]);

% Monte Carlo Sampling
for k = 1:num_samples
    % Randomly sample parameters
    theta2_initial_sample = rand() * diff(theta2_initial_range) + theta2_initial_range(1);
    misalignment_a_sample = rand() * diff(misalignment_a_range) + misalignment_a_range(1);
    misalignment_b_sample = rand() * diff(misalignment_b_range) + misalignment_b_range(1);
    E_sample = rand() * diff(E_range) + E_range(1);
    G_sample = rand() * diff(G_range) + G_range(1);
    manual_azimuth_a_sample = rand() * diff(manual_azimuth_a_range) + manual_azimuth_a_range
        (1);
    manual_azimuth_b_sample = rand() * diff(manual_azimuth_b_range) + manual_azimuth_b_range
        (1);

    % Compute theta2 range for this sample
    theta2 = linspace(theta2_initial_sample, (theta2_initial_sample + 2*pi), steps);
    theta2_star = theta2 - theta2_initial_sample;

    % Solve for theta3 and theta4 (same fsolve approach as original code)
    x0 = [theta3_guess, theta4_guess];
    for i = 1:steps
        theta2_val = theta2(i);
        solution = fsolve(@(x) loop_equation(x, theta2_val), x0, options);
        theta3(i) = solution(1);
        theta4(i) = solution(2);
        x0 = solution;
    end

    % Compute dependent variables
    azimuth_a = -(pi - theta2 + theta3) + (pi - theta2_initial_sample + theta3(1)) +
        manual_azimuth_a_sample;
    azimuth_b = -(pi - theta4 + theta3) + (pi - theta4(1) + theta3(1)) +
        manual_azimuth_b_sample;

    bending_a = -atan2(sin(misalignment_a_sample) .* sin(azimuth_a), cos(
        misalignment_a_sample)) + atan2(sin(misalignment_a_sample) * sin(
        manual_azimuth_a_sample), cos(misalignment_a_sample));
    bending_b = -atan2(sin(misalignment_b_sample) .* sin(azimuth_b), cos(
        misalignment_b_sample)) + atan2(sin(misalignment_b_sample) * sin(
        manual_azimuth_b_sample), cos(misalignment_b_sample));

    for i = 1:steps
        Ma = -2*E_sample*I*(2*bending_a(i) + bending_b(i))/L3;
        Mb = 2*E_sample*I*(bending_a(i) + 2*bending_b(i))/L3;
        M = @(x) ((Mb - Ma) / L3) * x + Ma;
        bending_energy(i) = (1 / (2 * E_sample * I)) * integral(@(x) M(x).^2, 0, L3);
    end

    torsion_a = atan2(sin(misalignment_a_sample) * cos(azimuth_a), cos(misalignment_a_sample)
        ) - atan2(sin(misalignment_a_sample) * cos(manual_azimuth_a_sample), cos(
        misalignment_a_sample));
    torsion_b = atan2(sin(misalignment_b_sample) * cos(azimuth_b), cos(misalignment_b_sample)
        ) - atan2(sin(misalignment_b_sample) * cos(manual_azimuth_b_sample), cos(
        misalignment_b_sample));

    torsion = (torsion_b - torsion_a) .* J .* G_sample ./ L3;
    torsion_energy = (torsion.^2 .* L3) ./ (2 .* G_sample .* J);
    total_energy = bending_energy + torsion_energy;

```

```matlab
    % Compute derivative
    d_total_energy_d_theta2_samples(k, :) = gradient(total_energy, theta2);
end

% Compute statistics (mean, min, max)
d_total_energy_d_theta2_mean = mean(d_total_energy_d_theta2_samples, 1);
d_total_energy_d_theta2_min = min(d_total_energy_d_theta2_samples, [], 1);
d_total_energy_d_theta2_max = max(d_total_energy_d_theta2_samples, [], 1);

% Plot results
hfig = figure;
hold on;
fill([theta2_star, fliplr(theta2_star)], [d_total_energy_d_theta2_max, fliplr(d_total_energy_d_theta2_min)], ...
    [0.8, 0.8, 0.8], 'FaceAlpha', 0.3, 'EdgeColor', 'none', 'DisplayName','Bandwidth'); % Shaded bandwidth
% plot(theta2_star, d_total_energy_d_theta2_mean, 'b', 'LineWidth', 2); % Mean curve
hold on
plot(x1_range_scaled, -y1_range, 'b', 'LineWidth', 1.5, 'DisplayName','Experimental␣data');
hold on
plot(x_trimmed-deg2rad(30), y_trimmed+0.0002, 'r', 'LineWidth', 1.5, 'DisplayName','FEM␣in␣Simscape');
hold on
plot(theta2_star, d_total_energy_d_theta2, 'g', 'LineWidth',1.5, 'DisplayName','Euler-Bernouilli');

legend('show', 'Location', 'southwest', 'Interpreter', 'latex');
xlabel('Angle␣$\theta_2^*$␣(rad)');
ylabel('Torque␣$(N␣\cdot␣m)$');

xticks(0:pi/2:2*pi); % Set tick marks from 0 to 2*pi in steps of pi/2
xticklabels({'$0$', '$\frac{1}{2}\pi$', '$\pi$', '$\frac{3}{2}\pi$', '$2\pi$'}); % Label the tick marks

xlim([-0.2 2*pi+0.2]); % Adds a small buffer before the first tick at 0
ax = gca;

title({'Torque␣bandwidth␣under␣parameter␣relaxation:'},{'Validation'});
fname = 'TRILOGYPT';
grid on;

picturewidth = 20; % set this parameter and keep it forever
hw_ratio = 0.65; % feel free to play with this ratio
set(findall(hfig,'-property','FontSize'),'FontSize',17) % adjust fontsize to your document
set(findall(hfig,'-property','Box'),'Box','off') % optional
set(findall(hfig,'-property','Interpreter'),'Interpreter','latex')
set(findall(hfig,'-property','TickLabelInterpreter'),'TickLabelInterpreter','latex')
set(hfig,'Units','centimeters','Position',[3 3 picturewidth hw_ratio*picturewidth])
pos = get(hfig,'Position');
set(hfig,'PaperPositionMode','Auto','PaperUnits','centimeters','PaperSize',[pos(3), pos(4)])
% print(hfig,fname,'-dpdf','-painters','-fillpage')
print(hfig,fname,'-dpng','-painters')

%% Difference of Simscape and Euler/Bernoulli
x_trimmed_interp = interp1((x_trimmed-deg2rad(30)), y_trimmed, theta2_star, 'linear');  % You can also use 'spline' or 'pchip'

% Subtract the datasets
y_different = d_total_energy_d_theta2-x_trimmed_interp;

hfig = figure;
hold on;
plot(theta2_star, y_different, 'm', 'LineWidth',1.5);

% legend('show', 'Location', 'southwest', 'Interpreter', 'latex');
xlabel('Angle␣$\theta_2^*$␣(rad)');
ylabel('Torque␣$(N␣\cdot␣m)$');

xticks(0:pi/2:2*pi); % Set tick marks from 0 to 2*pi in steps of pi/2
xticklabels({'$0$', '$\frac{1}{2}\pi$', '$\pi$', '$\frac{3}{2}\pi$', '$2\pi$'}); % Label the tick marks
```

```

xlim([-0.2 2*pi+0.2]); % Adds a small buffer before the first tick at 0
ax = gca;

title({'Residual␣plot:'},{'Simscape␣minus␣Euler-Bernouilli'});
fname = 'Residual␣plot';
grid on;

picturewidth = 20; % set this parameter and keep it forever
hw_ratio = 0.65; % feel free to play with this ratio
set(findall(hfig,'-property','FontSize'),'FontSize',17) % adjust fontsize to your document
set(findall(hfig,'-property','Box'),'Box','off') % optional
set(findall(hfig,'-property','Interpreter'),'Interpreter','latex')
set(findall(hfig,'-property','TickLabelInterpreter'),'TickLabelInterpreter','latex')
set(hfig,'Units','centimeters','Position',[3 3 picturewidth hw_ratio*picturewidth])
pos = get(hfig,'Position');
set(hfig,'PaperPositionMode','Auto','PaperUnits','centimeters','PaperSize',[pos(3), pos(4)])
% print(hfig,fname,'-dpdf','-painters','-fillpage')
print(hfig,fname,'-dpng','-painters')
```

B

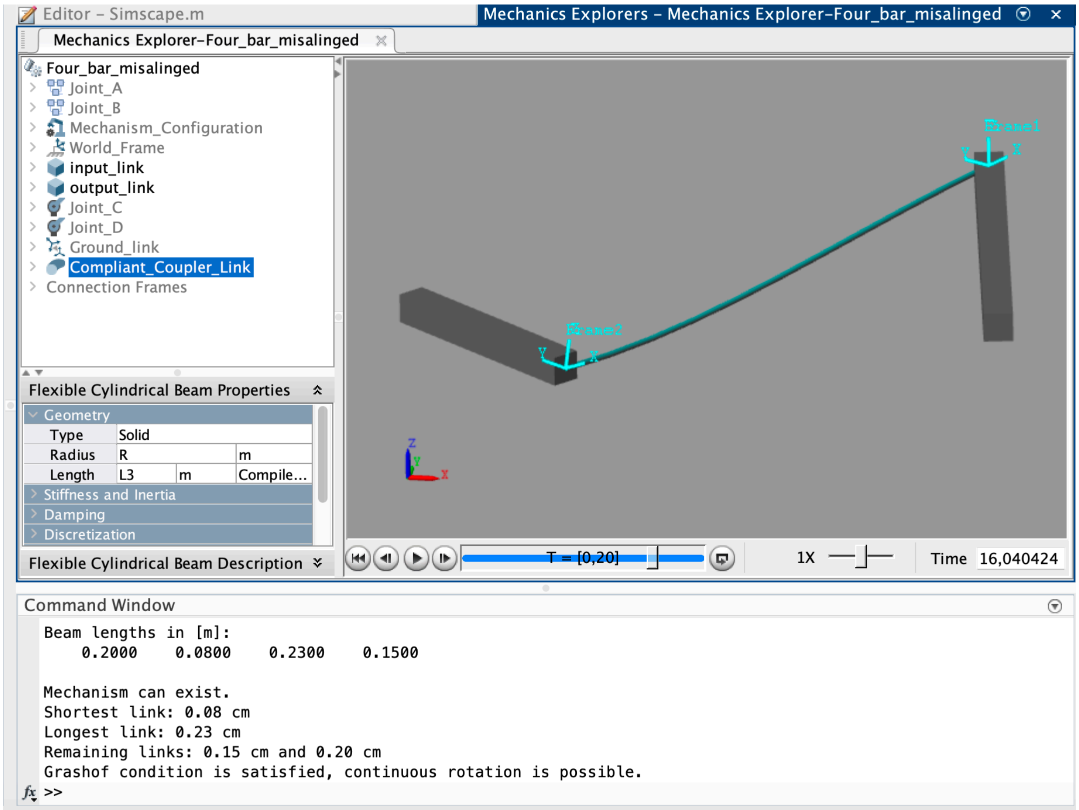
Appendix B - Simscape Multibody

```
% System parameters of compliant beam
E_mod = 200*10^9; % E-modulus in [Pa]
rho = 900; % Density in [kg/m^3]
R = 1e-3; % Cross-sectional radius in [m]
J = 1/32*pi*(2*R)^4; % Moment of inertia in torsion
I = 1/64*pi*(2*R)^4; % Moment of inertia in bending
G_mod = 77*10^9; % Shear modulus of hardened steel
manual_azimuth_a = deg2rad(210); %deg2rad(75); % Azimuth angle of misalignment in a
manual_azimuth_b = deg2rad(48); %deg2rad(35); % Azimuth angle of misalignment in b
misalignment_a = deg2rad(5); % Polar angle of misalignment in a
misalignment_b = deg2rad(5); % Polar angle of misalignment in b
theta2 = deg2rad(30);
direction = 1; % Direction of input link
runtime = 20;

% Number of bars
num_bars = 4;

% Manual beam lengths in [m]
% Assign the lengths directly instead of normalized values
% Ensure the lengths are within the desired range
beam_lengths = [0.2, 0.08, 0.23, 0.15]; % Replace these with your desired lengths

% Check whether the number of lengths matches the number of bars
if length(beam_lengths) ~= num_bars
    error('The␣number␣of␣beam␣lengths␣must␣match␣the␣number␣of␣bars.');
end

% Validate lengths against minimum and maximum constraints
l_min = 0.00001; % Minimum length in [m]
l_max = 0.30;   % Maximum length in [m]

for i = 1:num_bars
    if beam_lengths(i) < l_min || beam_lengths(i) > l_max
        error('Beam␣length␣%d␣is␣out␣of␣the␣allowed␣range␣[%.5f,␣%.2f]␣m.', i, l_min, l_max);
    end
end

% Assign lengths to specific bars
ground = beam_lengths(1);
input = beam_lengths(2);
coupler = beam_lengths(3);
output = beam_lengths(4);

% Display beam lengths
disp('Beam␣lengths␣in␣[m]:');
disp(beam_lengths);

%Check inequality condition to ensure mechanism can exist.
```

```
function checkTriangleInequality(beam_lengths)

    ground = beam_lengths(1);
    input = beam_lengths(2);
    coupler = beam_lengths(3);
    output = beam_lengths(4);

    condition1 = (ground + input + coupler > output);
    condition2 = (ground + input + output > coupler);
    condition3 = (ground + coupler + output > input);
    condition4 = (input + coupler + output > ground);

%Display results
    if condition1 && condition2 && condition3 && condition4
        disp('Mechanism␣can␣exist.');
    else
        disp('Mechanism␣can␣NOT␣exist.');
        error('Invalid␣linkage␣configuration␣due␣to␣triangle␣inequality␣violation.');
    end
end

checkTriangleInequality(beam_lengths);


% Grashof theorem states relative motion between two beams is possible when
% longest + shortest < sum of remaining two

function checkGrashof(beam_lengths)
    % Sort the link lengths to easily identify the shortest and longest links
    sorted_lengths = sort(beam_lengths);

    S = sorted_lengths(1); % Shortest link
    L = sorted_lengths(4); % Longest link
    P = sorted_lengths(2); % Second shortest link
    Q = sorted_lengths(3); % Third shortest link

    % Display the identified links
    fprintf('Shortest␣link:␣%.2f␣cm\n', S);
    fprintf('Longest␣link:␣%.2f␣cm\n', L);
    fprintf('Remaining␣links:␣%.2f␣cm␣and␣%.2f␣cm\n', P, Q);

    % Check Grashof's condition
    if S + L <= P + Q
        disp('Grashof␣condition␣is␣satisfied,␣continuous␣rotation␣is␣possible.');
    else
        disp('Grashof␣condition␣is␣NOT␣satisfied,␣continuous␣rotation␣is␣NOT␣possible.');
    end


end

% Check Grashof condition
checkGrashof(beam_lengths);




%Remaining parameters
k = 0; %spring stiffness in (N*m/deg)
c = 0; %Damping coefficient (N*m/(deg/s))




L1 = ground;
L2 = input;
L3 = coupler;
L4 = output;

%Angles of links
theta2_max = 360;
```

```
theta1 = deg2rad(0);


%Angular velocities and accelerations
w2 = 1; %rad/s
alpha2 = 0;

%End positions of coupler link end points
Bx = [];
By = [];

%Solve the unknows values
K1 = L1/L2;
K2 = L1/L4;
K3 = (L2^2 - L3^2 + L4^2 + L1^2)/(2* L2* L4);
K4 = (L1/L3);
K5 = (L4^2 - L1^2 - L2^2 - L3^2)/(2* L2* L3);

A = cos(theta2) - K1 - K2.*cos(theta2) + K3;
B = -2.*sin(theta2);
C = K1 - (K2 + 1).*cos(theta2) + K3;
D = cos(theta2) - K1 + K4.*cos(theta2) + K5;
E = -2.*sin(theta2);
F = K1 + (K4-1).*cos(theta2) + K5;

%Open configuration
theta3 = 2.*atan((-E - sqrt(E.^2 - 4.*D.*F))./(2.*D));
theta4 = 2.*atan((-B - sqrt(B.^2 - 4.*A.*C))./(2.*A));
theta5 = acos((L1 - L2*cos(theta2)-L3*cos(theta3))/L4);


w3 = (L2*w2*sin(theta4 - theta2)) ./ (L3*sin(theta3 - theta4));
w4 = (L2*w2*sin(theta2 - theta3)) ./ (L4*sin(theta4 - theta3));

A = L4*sin(theta4);
B = L3*sin(theta3);
C = L2*alpha2*sin(theta2) + L2*w2^2*cos(theta2) + ...
    L3*w3^2*cos(theta3) - L4*w4^2*cos(theta4);
D = L4*cos(theta4);
E = L3*cos(theta3);
F = L2*alpha2*cos(theta2) - L2*w2^2*sin(theta2) - ...
    L3*w3^2*sin(theta3) + L4*w4^2*sin(theta4);

alpha3 = (C*D - A*F)/(A*E - B*D);
alpha4 = (C*E - B*F)/(A*E - B*D);

%In polar complex form
R1 = L1*exp(theta1*1j);
R2 = L2*exp(theta2*1j);
R3 = L3*exp(theta3*1j);
R4 = L4*exp(theta4*1j);

% %Misalignment of joint 2
% gamma = deg2rad(5); %rad
% epsilon = deg2rad(0); %rad
%
%
% %Misalignment of joint 3
% zeta = deg2rad(5); %rad
% psi = deg2rad(0); %rad


%% Simout 1: No misalignment with load

simOut1 = sim('Four_bar_misalinged.slx');
%% Simout 1 code

ts_angle1 = simOut1.angle.Data;
ts_torque1 = simOut1.torque.Data;

% Define the index to start from (4% into the dataset)
```

```
start1_idx = round(0.02 * length(ts_torque1)) + 1;

% Extract the remaining data
x1_trimmed = ts_angle1(start1_idx:end);
y1_trimmed = ts_torque1(start1_idx:end);
```



Type	Solid		
Radius	R		m
Length	L3	m	Compile...

Time 16,040424

```
Beam lengths in [m]:
    0.2000    0.0800    0.2300    0.1500

Mechanism can exist.
Shortest link: 0.08 cm
Longest link: 0.23 cm
Remaining links: 0.15 cm and 0.20 cm
Grashof condition is satisfied, continuous rotation is possible.
>>
```

Figure B.1: Mechanics Explorer of Simscape model. Compliant coupler link is indicated in blue. The Mechanics Explorer lets users visualize and explore multi-body models. This tool allows for validating the mechanism's configuration and deformation modes of the compliant coupler link.

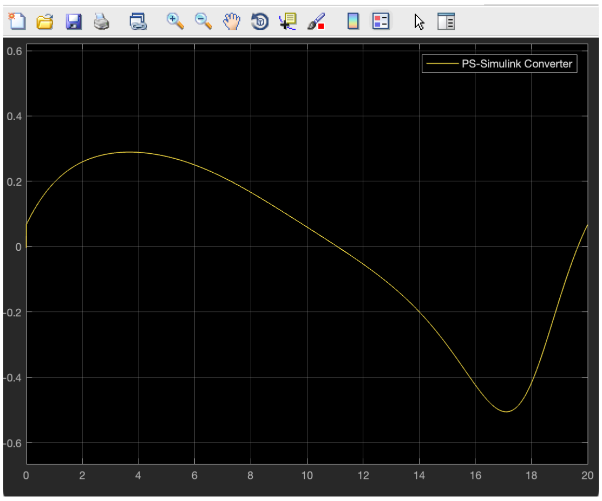


Figure B.2: Figure of the scope in Simscape. The scope displays output signals with respect to simulation time. ODE23t (mod. stiff/ Trapezoidal solver) is used as solver.

C

Appendix C - Graphical User Interface (GUI) in MATLAB App Designer

```
classdef Misalignment < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                      matlab.ui.Figure
        RotationDirectionSwitch       matlab.ui.control.Switch
        RotationDirectionSwitchLabel  matlab.ui.control.Label
        CrankMechanismLamp            matlab.ui.control.Lamp
        CrankMechanismLampLabel       matlab.ui.control.Label
        OutputLinkEditField           matlab.ui.control.NumericEditField
        OutputLinkEditFieldLabel      matlab.ui.control.Label
        CouplerLinkEditField          matlab.ui.control.NumericEditField
        CouplerLinkEditFieldLabel     matlab.ui.control.Label
        InputLinkEditField            matlab.ui.control.NumericEditField
        InputLinkEditFieldLabel       matlab.ui.control.Label
        GroundLinkEditField           matlab.ui.control.NumericEditField
        GroundLinkEditFieldLabel      matlab.ui.control.Label
        Theta2initialSlider           matlab.ui.control.Slider
        Theta2initialSliderLabel      matlab.ui.control.Label
        AzimuthbSlider                matlab.ui.control.Slider
        AzimuthbSliderLabel           matlab.ui.control.Label
        AzimuthaSlider                matlab.ui.control.Slider
        MisalignmentbSlider           matlab.ui.control.Slider
        MisalignmentbSliderLabel      matlab.ui.control.Label
        MisalignmentaSlider           matlab.ui.control.Slider
        MisalignmentaSliderLabel      matlab.ui.control.Label
        AzimuthaSliderLabel           matlab.ui.control.Label
        UIAxes                        matlab.ui.control.UIAxes
    end


    properties (Access = private)
        E = 200e9;
        d = 2e-3;
        % L1;
        % L2;
        % L3;
        % L4;
        G = 77e9;
        steps = 200;
        I;
        J;

    end



    % Callbacks that handle component events
    methods (Access = private)

        % Code that executes after component creation
        function startupFcn(app)
            app.I = 1/64 * pi * app.d^4;
            app.J = 1/32 * pi * app.d^4;


        end

        % Button down function: UIAxes
        function UIAxesButtonDown(app, event)
            L1 = app.GroundLinkEditField.Value;
            L2 = app.InputLinkEditField.Value;
            L3 = app.CouplerLinkEditField.Value;
            L4 = app.OutputLinkEditField.Value;

            T1 = L1 + L3 - L2 - L4;
            T2 = L4 + L1 - L2 - L3;
            T3 = L4 + L3 - L2 - L1;

            if T1 < 0 && T2 < 0 && T3 > 0
                app.CrankMechanismLamp.Color = 'g';
            elseif T1 > 0 && T2 > 0 && T3 > 0
                app.CrankMechanismLamp.Color = 'g';
            else
                app.CrankMechanismLamp.Color = 'r';
            end


            if app.RotationDirectionSwitch.Value == "+"
                direction = 1;
            else
                direction = -1;
            end



            %Generation of array of theta2
            theta2 = linspace(deg2rad(app.Theta2initialSlider.Value), deg2rad(app.Theta2initialSlider.Value) + direction * 2*pi, app.steps);
            theta3 = zeros(1, app.steps);
            theta4 = zeros(1, app.steps);


            loop_equation = @(x, theta2_val) [
                L2*cos(theta2_val) + L3*cos(x(1)) - L4*cos(x(2)) - L1;
                L2*sin(theta2_val) + L3*sin(x(1)) - L4*sin(x(2));
                ];

            % Initial geometric approximation
            x3 = L2*cos(deg2rad(app.Theta2initialSlider.Value)) + L1; % x-coordinate of coupler
            y3 = L2*sin(deg2rad(app.Theta2initialSlider.Value));     % y-coordinate of coupler
            theta3_guess = deg2rad(30); % Approximate theta3
            theta4_guess = deg2rad(89);      % Approximate theta4
            x0 = [theta3_guess, theta4_guess]; % Initial guess for fsolve


           options = optimoptions('fsolve','Display','none');

           for i = 1:app.steps
               theta2_val = theta2(i);
               solution = fsolve(@(x) loop_equation(x, theta2_val), x0, options);
               theta3(i) = solution(1);
               theta4(i) = solution(2);
               x0 = solution; % Use the previous solution as the next initial guess
           end
           theta3_initial = theta3(1);
           theta4_initial = theta4(1);

           azimuth_a_offset = (pi - deg2rad(app.Theta2initialSlider.Value) + theta3_initial);
           azimuth_a = -(pi - theta2 + theta3) + azimuth_a_offset + deg2rad(app.
               AzimuthaSlider.Value);

           azimuth_b_offset = (pi - theta4_initial + theta3_initial);
           azimuth_b = -(pi - theta4 + theta3) + azimuth_b_offset + deg2rad(app.
               AzimuthbSlider.Value);

           bending_a_offset = atan2((sin(deg2rad(app.MisalignmentaSlider.Value)) * sin(
               deg2rad(app.AzimuthaSlider.Value))), cos(deg2rad(app.MisalignmentaSlider.Value
               )));
           bending_a = -atan2((sin(deg2rad(app.MisalignmentaSlider.Value)) * sin(azimuth_a)),
                (cos(deg2rad(app.MisalignmentaSlider.Value)))) + bending_a_offset;

           bending_b_offset = atan2((sin(deg2rad(app.MisalignmentbSlider.Value)) * sin(
               deg2rad(app.AzimuthbSlider.Value))), cos(deg2rad(app.MisalignmentbSlider.Value
               )));
           bending_b = -atan2((sin(deg2rad(app.MisalignmentbSlider.Value)) * sin(azimuth_b)),
                (cos(deg2rad(app.MisalignmentbSlider.Value)))) + bending_b_offset;

           bending_energy = zeros(1, app.steps);
           Ma_values = zeros(1, app.steps);
           Mb_values = zeros(1, app.steps);

           for i = 1: app.steps
           %Calculate Ma and Mb at each step
               Ma = -(2*app.E*app.I*(2*bending_a(i) + bending_b(i)))/(L3);
               Mb =  (2*app.E*app.I*(bending_a(i) + 2*bending_b(i)))/(L3);
               % Ma = -2*app.E*app.I*(4*bending_a(i) - bending_b(i)) / (5*L3);
               % Mb = -2*app.E*app.I*(7*bending_a(i) + 2*bending_b(i)) / (5*L3);

               Ma_values(i) = Ma;
               Mb_values(i) = Mb;

           %The moment function:
               M = @(x) ((Mb - Ma) / L3) * x + Ma;

               %Integrate the squared moment function from 0 to L3
               bending_energy(i) = (1 / (2 * app.E * app.I)) * integral(@(x) M(x).^2, 0, L3);

           end

           torsion_a_offset = atan2((sin(deg2rad(app.MisalignmentaSlider.Value)) * cos(
               deg2rad(app.AzimuthaSlider.Value))), cos((deg2rad(app.MisalignmentaSlider.
               Value))));
           torsion_a = atan2((sin(deg2rad(app.MisalignmentaSlider.Value)) * cos(azimuth_a)),
               (cos(deg2rad(app.MisalignmentaSlider.Value)))) - torsion_a_offset;

           torsion_b_offset = atan2((sin(deg2rad(app.MisalignmentbSlider.Value)) * cos(
               deg2rad(app.AzimuthbSlider.Value))), cos((deg2rad(app.MisalignmentbSlider.
               Value))));
           torsion_b = atan2((sin(deg2rad(app.MisalignmentbSlider.Value)) * cos(azimuth_b)),
               (cos(deg2rad(app.MisalignmentbSlider.Value)))) - torsion_b_offset;

           torsion_energy = zeros(1, app.steps);
           torsion = zeros(1, app.steps);

           for i = 1: app.steps
               absolute_torsion = abs(torsion_b(i) - torsion_a(i));
               torsion(i) = absolute_torsion * app.J * app.G / L3;
               torsion_energy(i) = torsion(i)^2 * L3 / (2 * app.G * app.J);
           end

           total_energy = bending_energy + torsion_energy;
           d_total_energy_d_theta2 = gradient(total_energy, theta2);

           if app.RotationDirectionSwitch.Value == "-"
               app.UIAxes.XDir = 'reverse';
```

```
           else
               app.UIAxes.XDir = 'normal';
           end


           plot(app.UIAxes, theta2, d_total_energy_d_theta2);
        end
    end

    % Component initialization
    methods (Access = private)

        % Create UIFigure and components
        function createComponents(app)

            % Create UIFigure and hide until all components are created
            app.UIFigure = uifigure('Visible', 'off');
            app.UIFigure.Position = [100 100 654 606];
            app.UIFigure.Name = 'MATLAB␣App';

            % Create UIAxes
            app.UIAxes = uiaxes(app.UIFigure);
            title(app.UIAxes, 'Theta2␣vs.␣torque')
            xlabel(app.UIAxes, 'Theta2␣[rad]')
            ylabel(app.UIAxes, 'Torque␣[N*m]')
            zlabel(app.UIAxes, 'Z')
            app.UIAxes.XGrid = 'on';
            app.UIAxes.YGrid = 'on';
            app.UIAxes.ButtonDownFcn = createCallbackFcn(app, @UIAxesButtonDown, true);
            app.UIAxes.Position = [14 305 615 290];

            % Create AzimuthaSliderLabel
            app.AzimuthaSliderLabel = uilabel(app.UIFigure);
            app.AzimuthaSliderLabel.HorizontalAlignment = 'right';
            app.AzimuthaSliderLabel.Position = [14 251 58 22];
            app.AzimuthaSliderLabel.Text = 'Azimuth␣a';

            % Create MisalignmentaSliderLabel
            app.MisalignmentaSliderLabel = uilabel(app.UIFigure);
            app.MisalignmentaSliderLabel.HorizontalAlignment = 'right';
            app.MisalignmentaSliderLabel.Position = [356 202 86 22];
            app.MisalignmentaSliderLabel.Text = 'Misalignment␣a';

            % Create MisalignmentaSlider
            app.MisalignmentaSlider = uislider(app.UIFigure);
            app.MisalignmentaSlider.Limits = [0 8];
            app.MisalignmentaSlider.MajorTicks = [0 2 4 6 8];
            app.MisalignmentaSlider.MinorTicks = [1 2 3 4 5 6 7 8];
            app.MisalignmentaSlider.Position = [463 211 96 3];
            app.MisalignmentaSlider.Value = 4;

            % Create MisalignmentbSliderLabel
            app.MisalignmentbSliderLabel = uilabel(app.UIFigure);
            app.MisalignmentbSliderLabel.HorizontalAlignment = 'right';
            app.MisalignmentbSliderLabel.Position = [355 159 87 22];
            app.MisalignmentbSliderLabel.Text = 'Misalignment␣b';

            % Create MisalignmentbSlider
            app.MisalignmentbSlider = uislider(app.UIFigure);
            app.MisalignmentbSlider.Limits = [0 8];
            app.MisalignmentbSlider.MajorTicks = [0 2 4 6 8];
            app.MisalignmentbSlider.MinorTicks = [1 2 3 4 5 6 7 8];
            app.MisalignmentbSlider.Position = [463 168 96 3];
            app.MisalignmentbSlider.Value = 4;

            % Create AzimuthaSlider
            app.AzimuthaSlider = uislider(app.UIFigure);
            app.AzimuthaSlider.Limits = [0 360];
            app.AzimuthaSlider.Position = [93 260 150 3];

            % Create AzimuthbSliderLabel
            app.AzimuthbSliderLabel = uilabel(app.UIFigure);
            app.AzimuthbSliderLabel.HorizontalAlignment = 'right';
            app.AzimuthbSliderLabel.Position = [13 181 59 22];
            app.AzimuthbSliderLabel.Text = 'Azimuth␣b';

            % Create AzimuthbSlider
            app.AzimuthbSlider = uislider(app.UIFigure);
            app.AzimuthbSlider.Limits = [0 360];
            app.AzimuthbSlider.Position = [93 190 150 3];

            % Create Theta2initialSliderLabel
            app.Theta2initialSliderLabel = uilabel(app.UIFigure);
            app.Theta2initialSliderLabel.HorizontalAlignment = 'right';
            app.Theta2initialSliderLabel.Position = [300 263 73 22];
            app.Theta2initialSliderLabel.Text = 'Theta2␣initial';

            % Create Theta2initialSlider
            app.Theta2initialSlider = uislider(app.UIFigure);
            app.Theta2initialSlider.Limits = [0 360];
            app.Theta2initialSlider.Position = [394 272 216 3];

            % Create GroundLinkEditFieldLabel
            app.GroundLinkEditFieldLabel = uilabel(app.UIFigure);
            app.GroundLinkEditFieldLabel.HorizontalAlignment = 'right';
            app.GroundLinkEditFieldLabel.Position = [28 98 71 22];
            app.GroundLinkEditFieldLabel.Text = 'Ground␣Link';

            % Create GroundLinkEditField
            app.GroundLinkEditField = uieditfield(app.UIFigure, 'numeric');
            app.GroundLinkEditField.Position = [114 98 100 22];

            % Create InputLinkEditFieldLabel
            app.InputLinkEditFieldLabel = uilabel(app.UIFigure);
            app.InputLinkEditFieldLabel.HorizontalAlignment = 'right';
            app.InputLinkEditFieldLabel.Position = [41 53 58 22];
            app.InputLinkEditFieldLabel.Text = 'Input␣Link';

            % Create InputLinkEditField
            app.InputLinkEditField = uieditfield(app.UIFigure, 'numeric');
            app.InputLinkEditField.Position = [114 53 100 22];

            % Create CouplerLinkEditFieldLabel
            app.CouplerLinkEditFieldLabel = uilabel(app.UIFigure);
            app.CouplerLinkEditFieldLabel.HorizontalAlignment = 'right';
            app.CouplerLinkEditFieldLabel.Position = [276 98 73 22];
            app.CouplerLinkEditFieldLabel.Text = 'Coupler␣Link';

            % Create CouplerLinkEditField
            app.CouplerLinkEditField = uieditfield(app.UIFigure, 'numeric');
            app.CouplerLinkEditField.Position = [364 98 100 22];

            % Create OutputLinkEditFieldLabel
            app.OutputLinkEditFieldLabel = uilabel(app.UIFigure);
            app.OutputLinkEditFieldLabel.HorizontalAlignment = 'right';
            app.OutputLinkEditFieldLabel.Position = [281 53 68 22];
            app.OutputLinkEditFieldLabel.Text = 'Output␣Link';

            % Create OutputLinkEditField
            app.OutputLinkEditField = uieditfield(app.UIFigure, 'numeric');
            app.OutputLinkEditField.Position = [364 53 100 22];

            % Create CrankMechanismLampLabel
            app.CrankMechanismLampLabel = uilabel(app.UIFigure);
            app.CrankMechanismLampLabel.HorizontalAlignment = 'right';
            app.CrankMechanismLampLabel.Position = [488 98 102 22];
            app.CrankMechanismLampLabel.Text = 'Crank␣Mechanism';

            % Create CrankMechanismLamp
            app.CrankMechanismLamp = uilamp(app.UIFigure);
            app.CrankMechanismLamp.Position = [597 98 20 20];

            % Create RotationDirectionSwitchLabel
            app.RotationDirectionSwitchLabel = uilabel(app.UIFigure);
            app.RotationDirectionSwitchLabel.HorizontalAlignment = 'center';
            app.RotationDirectionSwitchLabel.Position = [503 28 101 21];
            app.RotationDirectionSwitchLabel.Text = 'Rotation␣Direction';

            % Create RotationDirectionSwitch
            app.RotationDirectionSwitch = uiswitch(app.UIFigure, 'slider');
            app.RotationDirectionSwitch.Items = {'-', '+'};
            app.RotationDirectionSwitch.Position = [530 55 45 20];
            app.RotationDirectionSwitch.Value = '-';

            % Show the figure after all components are created
            app.UIFigure.Visible = 'on';
        end
    end

    % App creation and deletion
    methods (Access = public)

        % Construct app
        function app = Misalignment

            % Create UIFigure and components
            createComponents(app)

            % Register the app with App Designer
            registerApp(app, app.UIFigure)

            % Execute the startup function
            runStartupFcn(app, @startupFcn)

            if nargout == 0
                clear app
            end
        end

        % Code that executes before app deletion
        function delete(app)

            % Delete UIFigure when app is deleted
            delete(app.UIFigure)
        end
    end
end
```

D

Appendix D - Physical model

The physical model was machined out of 6000-series aluminium. 9x4x3 mm bearings were used in the joints. All bearings were glued into reamed H7 holes. The compliant beam was 200 mm in length with a 2 mm diameter. The cylinders that hold the misaligned revolute joints (figure D.4), were hold into place with M4 set-screws. A Thorlabs structure was attached to the ground link (figure D.3). This structure provided unrestricted motion of the four bar mechanism while maintaining the connection between the frame and the torque sensor. Two coupler links were used to prevent jamming as result of a possible misalignment between axis of driver and axis of input link, as shown in figure D.1.

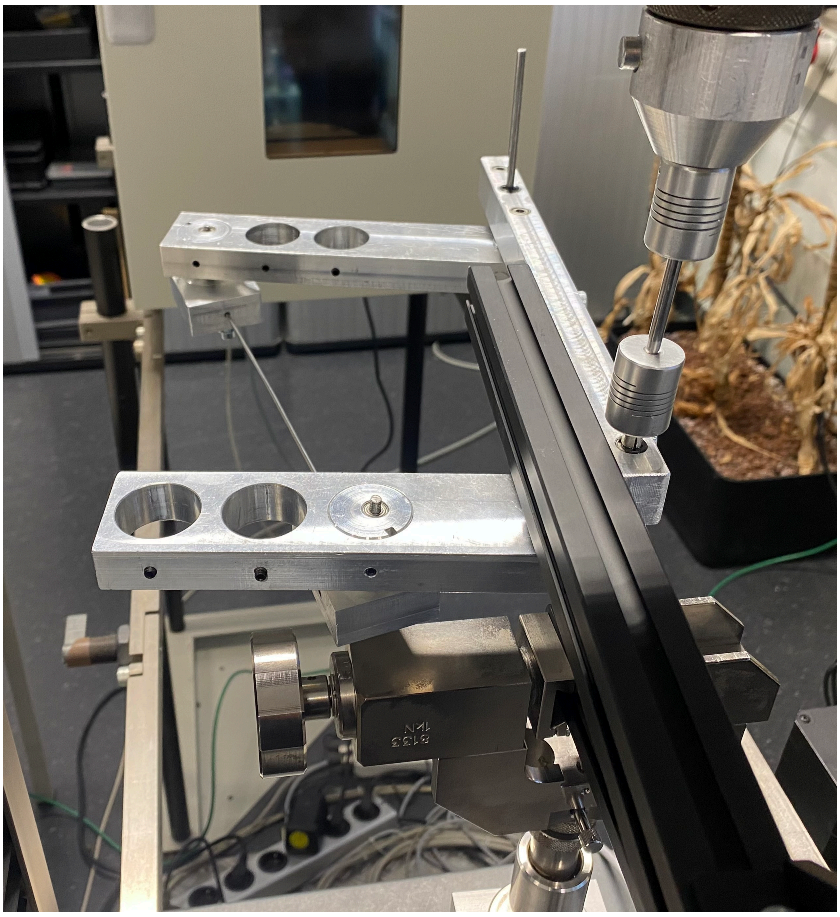


Figure D.1: Physical model. Input link is connected to two coupler links to prevent misalignment between constant velocity driver and axis of input link. Parts are hold into place using set screws, resulting in a modular design allowing for adjusting lengths, misalignment magnitudes and initial configuration.

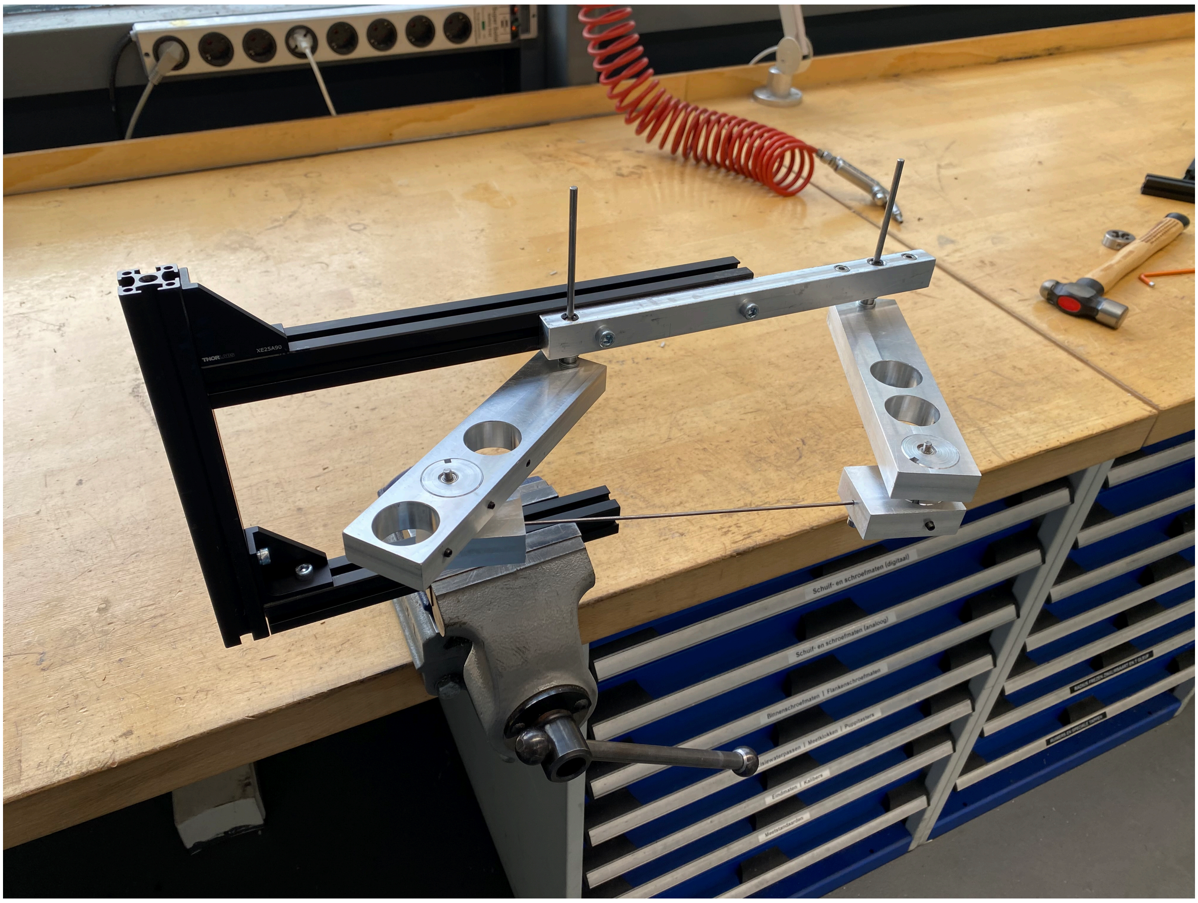


Figure D.2: Physical model with Thorlabs structure to make connection between driver and torque sensor without limiting motion of four bar mechanism.

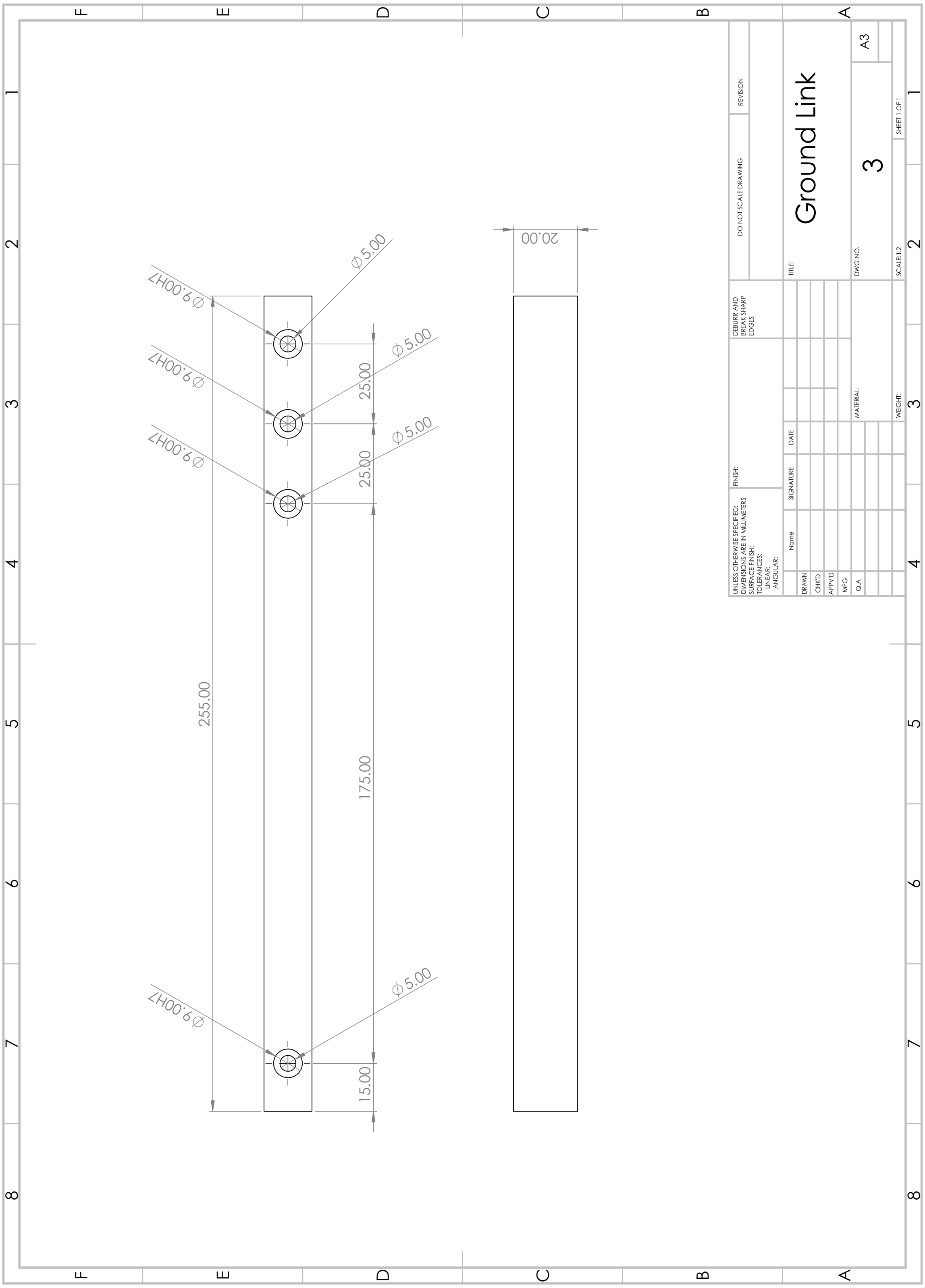


Figure D.3: Technical drawing of ground link. Three different length options: 175, 200 or 225 mm. Bearings were glued into place using Locktite.

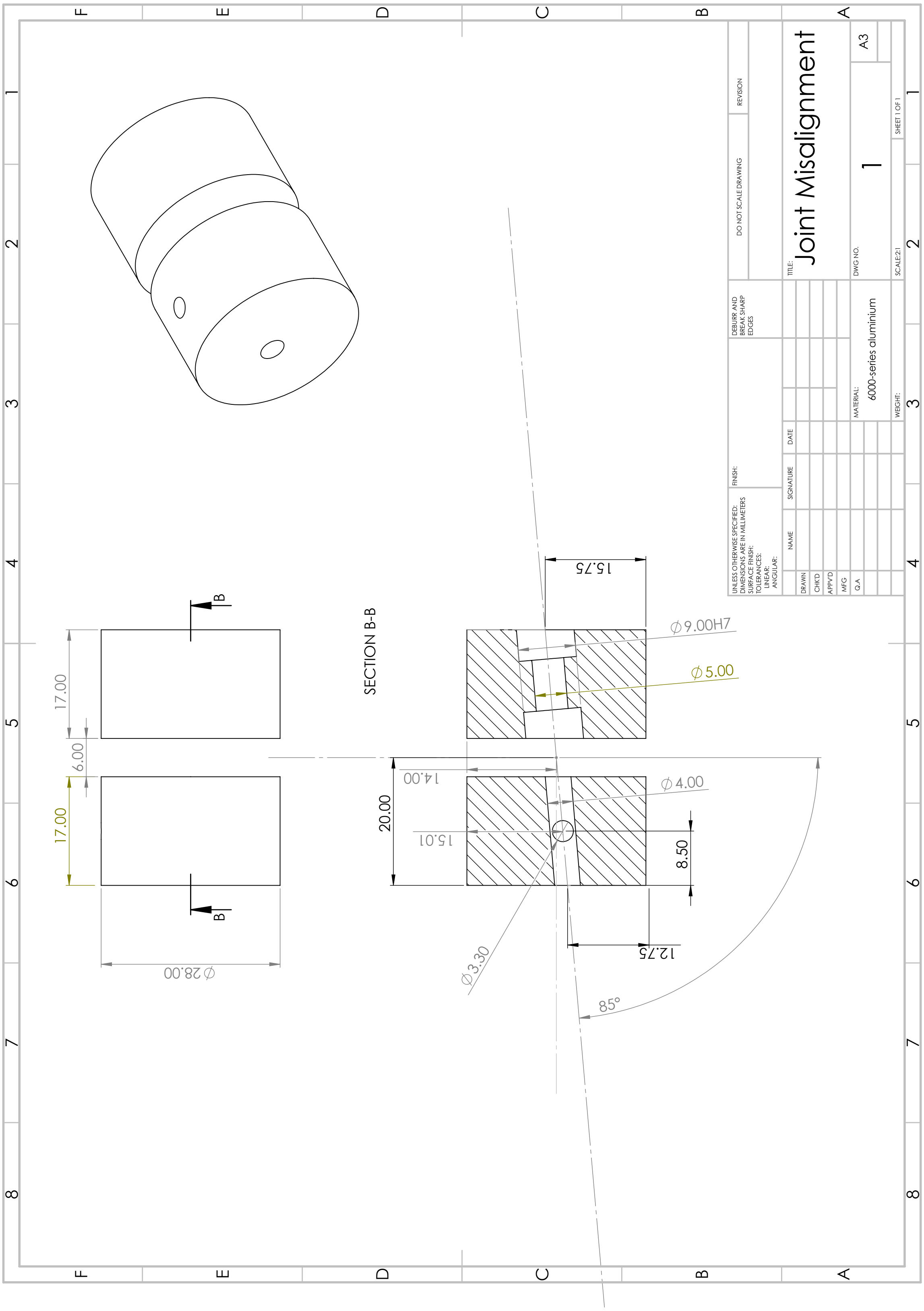


Figure D.4: Technical drawing of joint misalignment cylinders. Ball bearings were glued into place using locktite. Machined out of 6000-series aluminium. Ball bearings were glued into place using locktite.

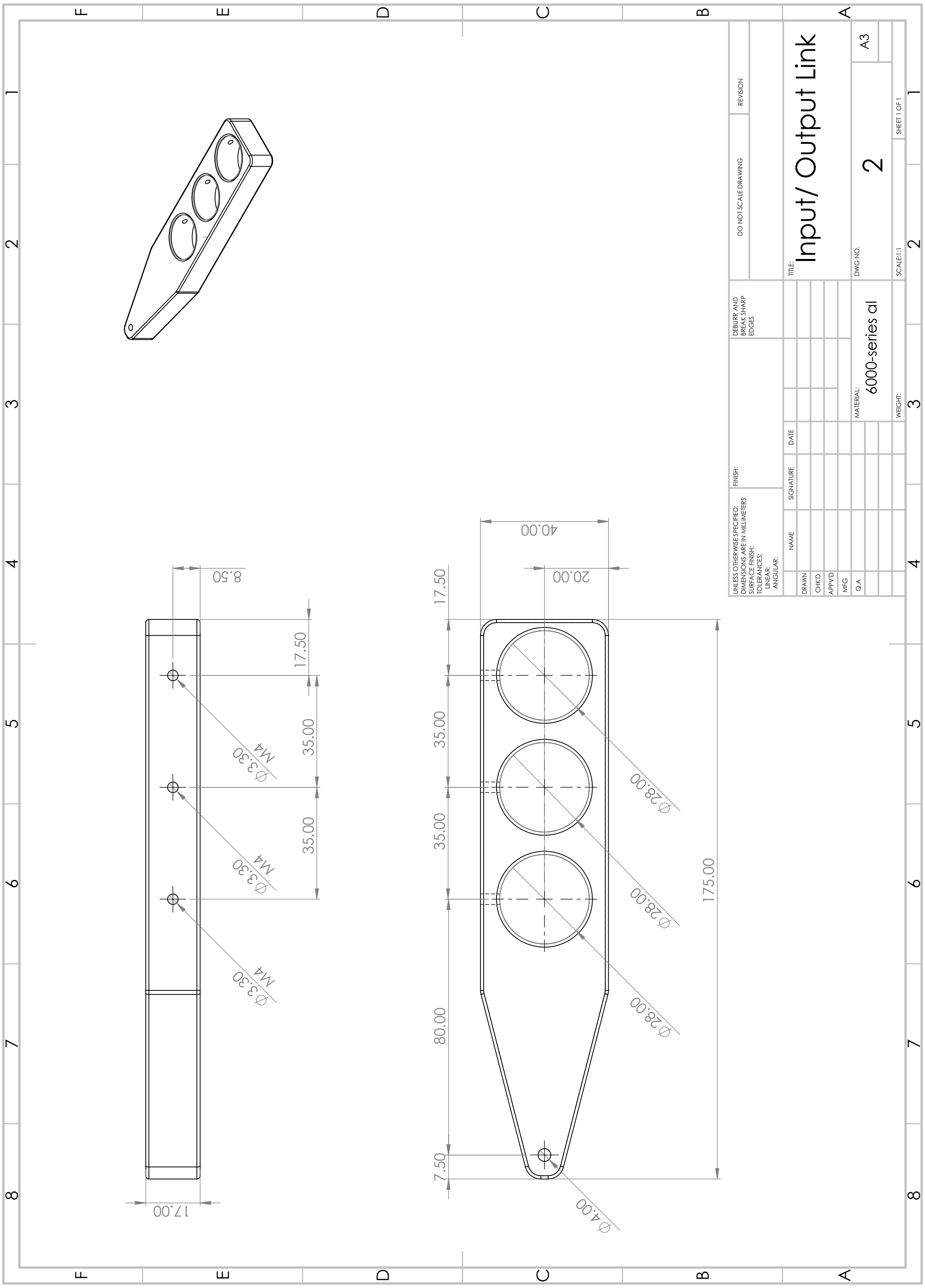


Figure D.5: Input and output link, both with three different options for misalignment cylinders. Cylinders are hold into place using set-screws.

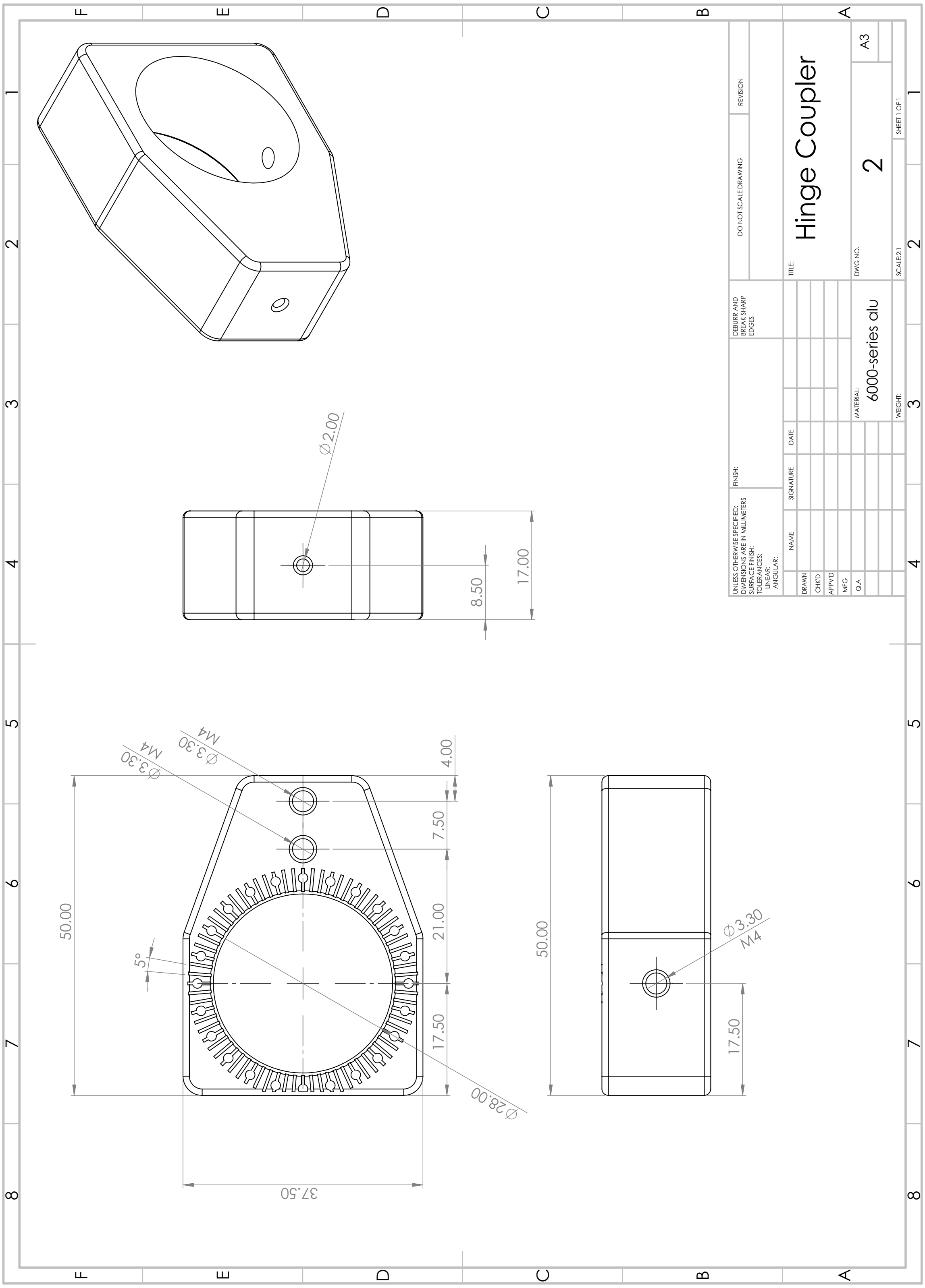


Figure D.6: Coupler part that connects the misaligned joints with the compliant coupler link. Laser engravings of 5° increments ensured positioning accuracy of azimuth angles.

E

Appendix E - Concepts

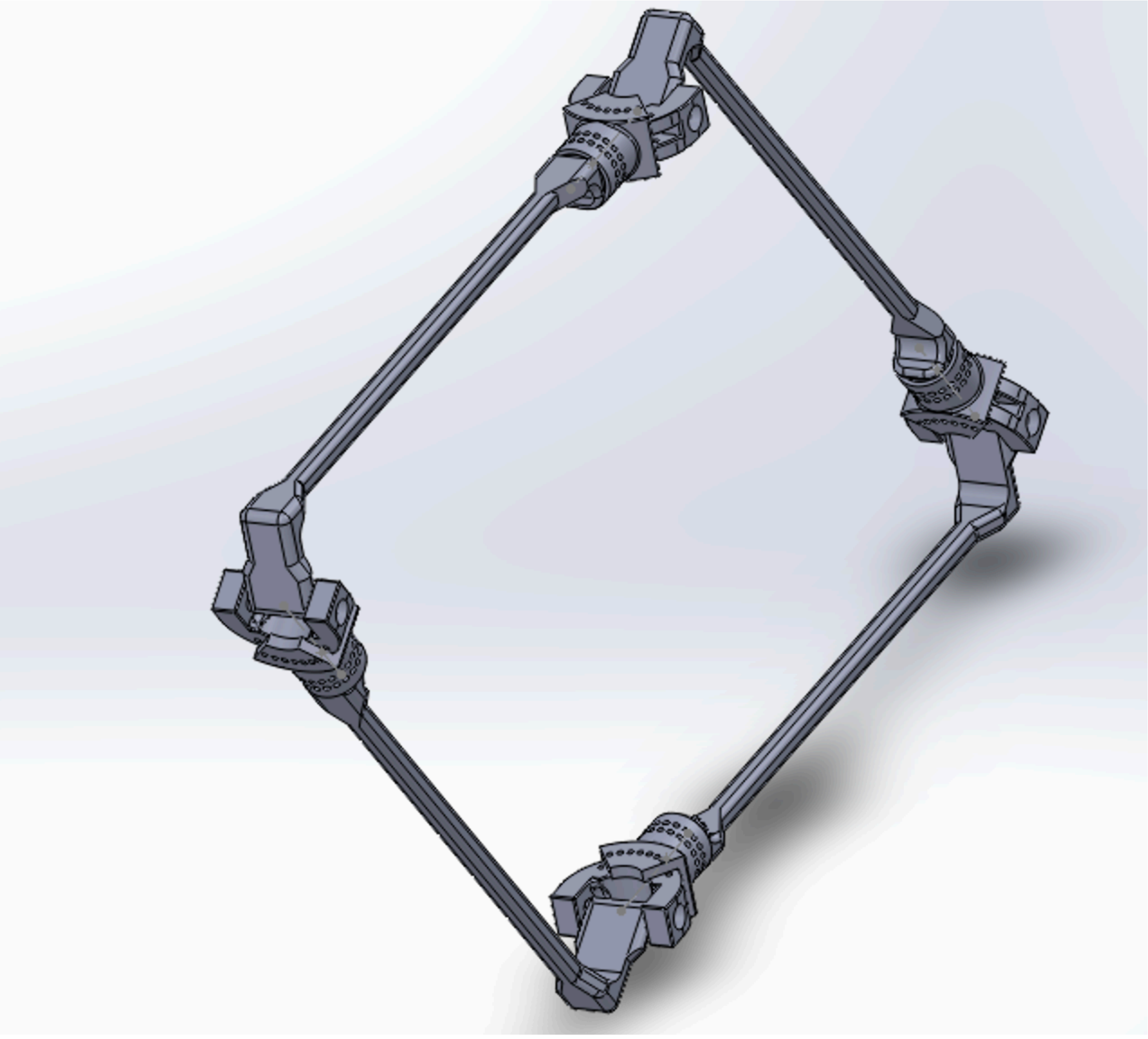


Figure E.1: Concept of four-bar with misaligned over-constraints. Possibility of misaligning all over-constraints. Concept has four compliant links which length is not adjustable once printed.

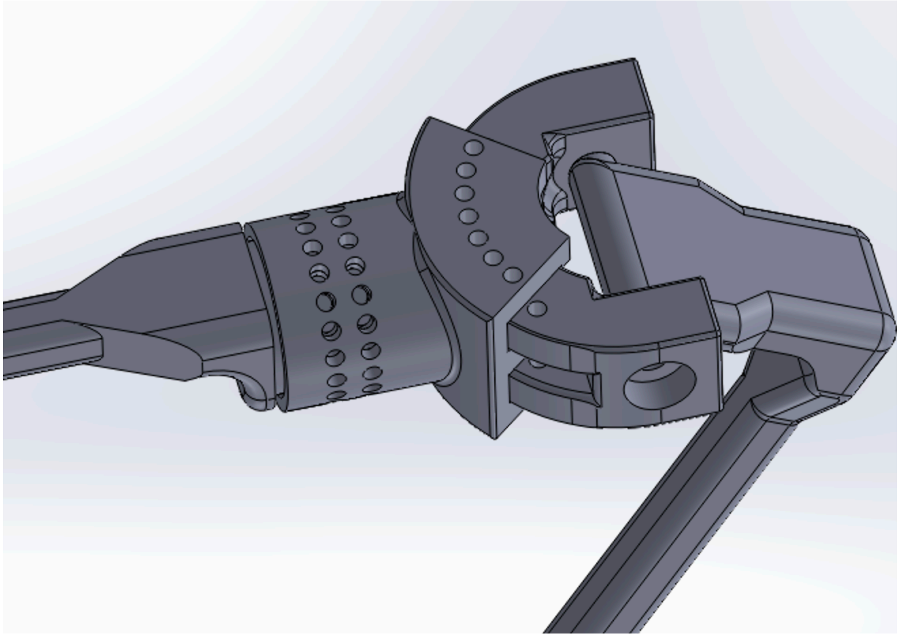


Figure E.2: Adjustable misalignment of a revolute joint. Two ball bearings have to be inserted on each side if the C-shaped part. Small pins can be inserted in the holes to make misalignment fixed. The four-bar has the ability to have a initial stress free configuration with a modular design.

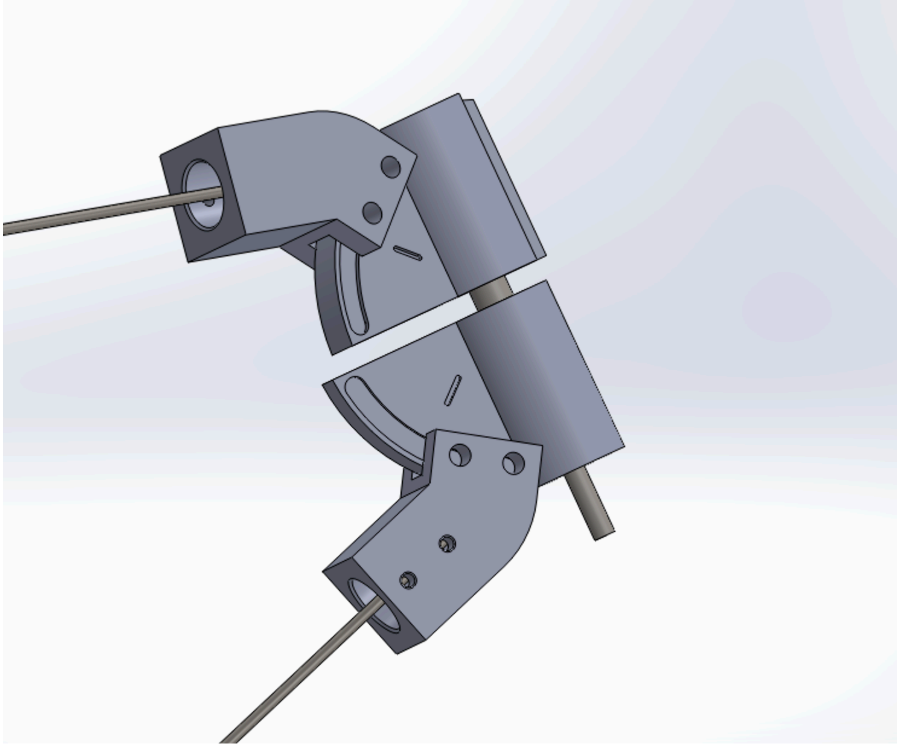


Figure E.3: Iteration of adjustable misaligned joint. The idea is that the end parts of the two compliant links can slide in the slots with their pins (not shown). Once the demanded misalignment is set, the screws can be tightened to fixate the misalignment.

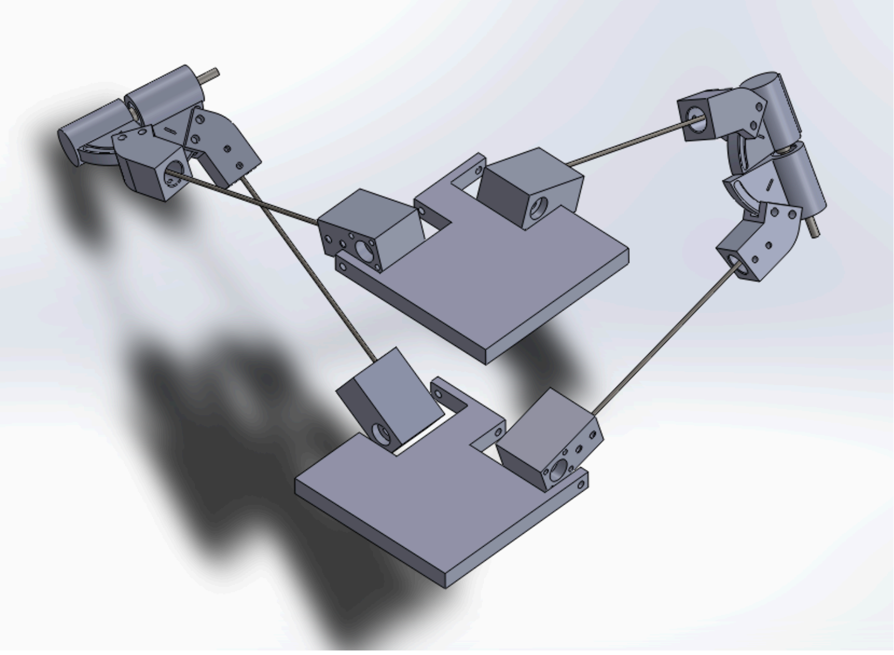


Figure E.4: Concept of Sarrus mechanism with similar misaligned revolute type as shown in Figure E.3.

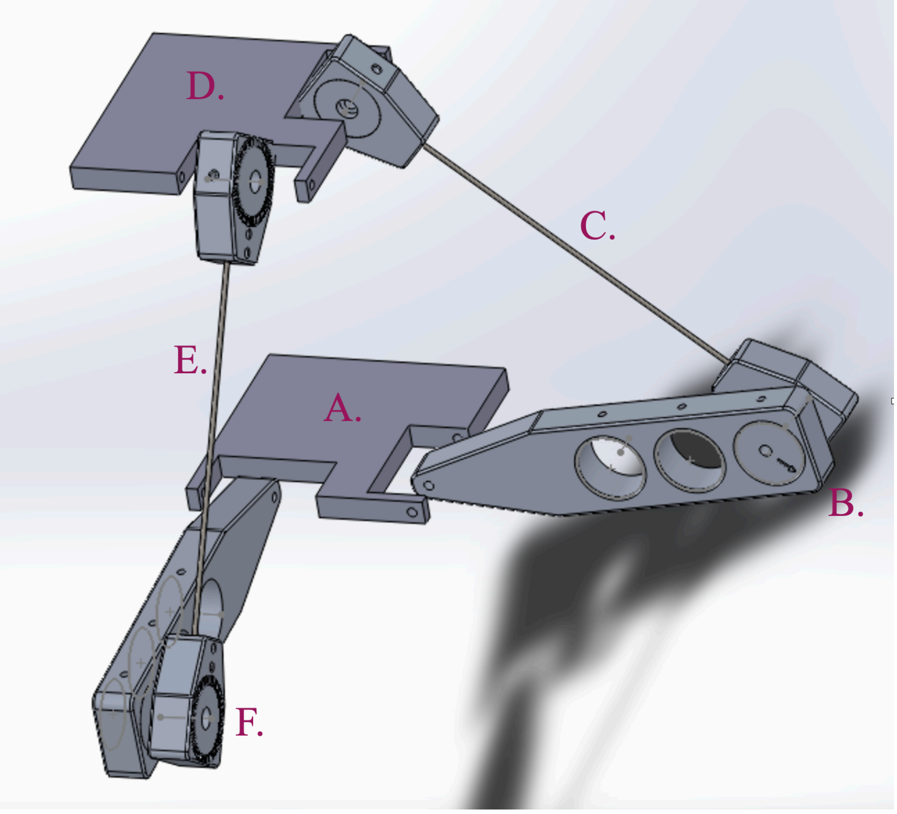


Figure E.5: Sarrus mechanism concept based on misaligned joint type used from physical prototype of research paper. Bottom and top plate are additional parts. Furthermore, all parts of the misaligned over-constraint four-bar mechanism from the research paper can be re-used. A. Bottom plate; B. Misaligned revolute joint; C. Compliant link; D. Top plate; E. Compliant link; F. Misaligned revolute joint.

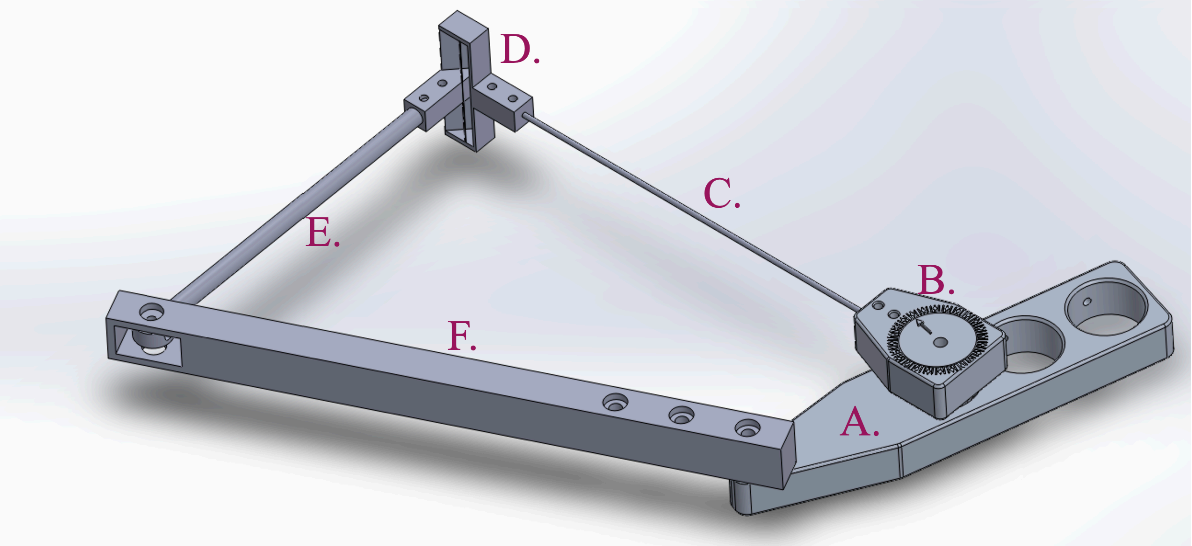


Figure E.6: Connection between output link and compliant coupler link could be a compliant joint when it replaces the joint that makes a reciprocating motion. In addition, this joint could be positioned with a misalignment too. Note that this compliant joint type does introduce joint stiffness to the system. This can be modeled in Simscape by assigning a value to the joint stiffness. A. Input link; B. Misaligned revolute joint; C. Compliant coupler link; D. Compliant joint; E. Output link; F. Ground link.

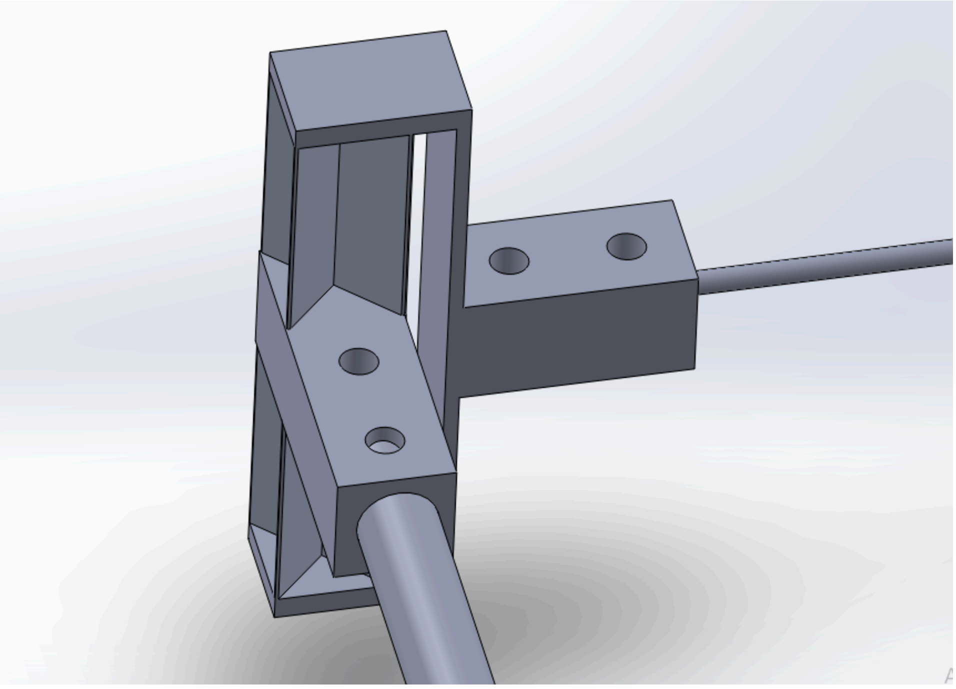


Figure E.7: Close-up of compliant joint of figure E.6, that allows for one DOF over a bounded angle. It forms the connection between the compliant coupler link and the output link.