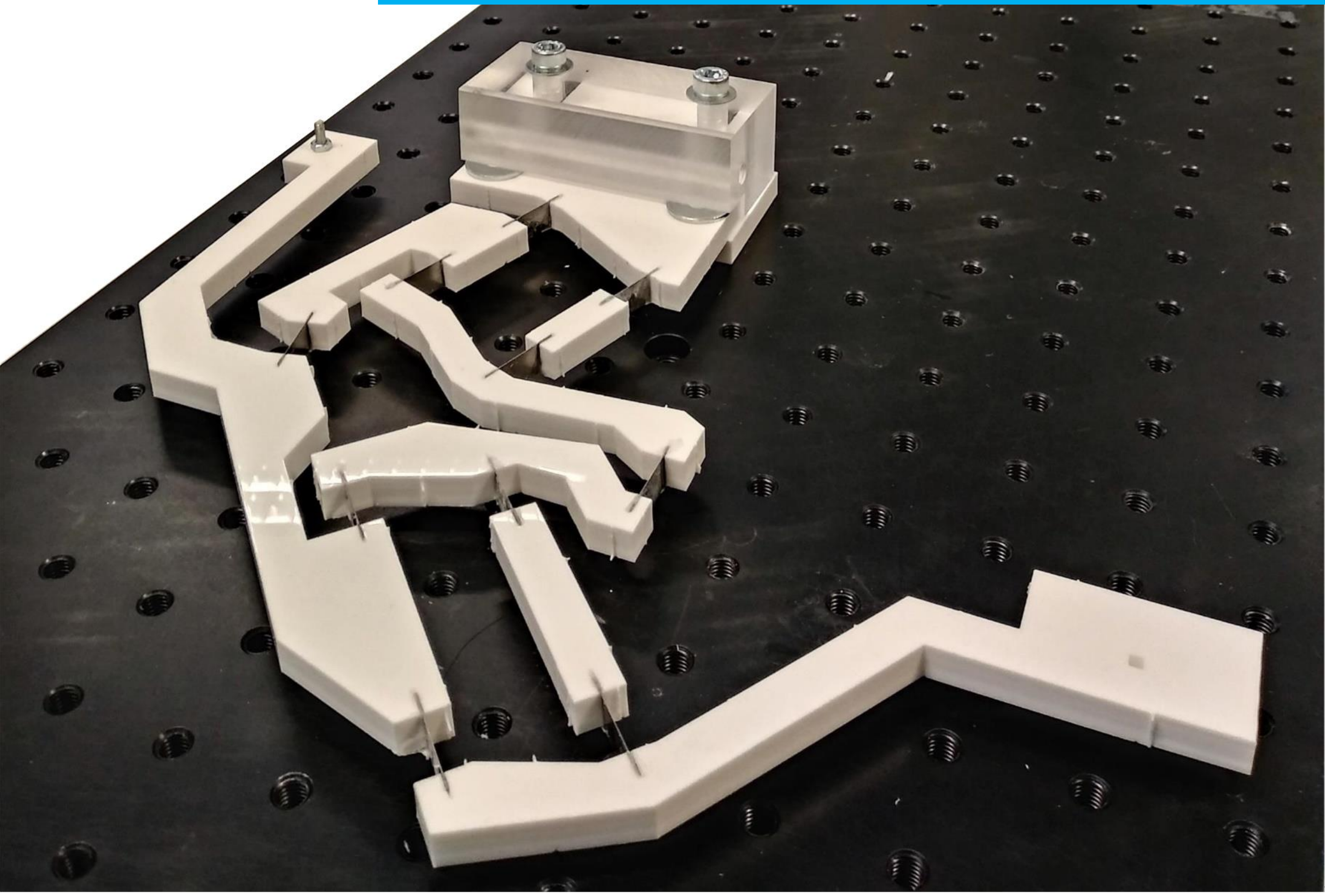
Department of Precision and Microsystems Engineering

Compliant Remote-Center-of-Motion Mechanism Optimized for Energy-Dispersive Spectroscopy

J.L.H.M. Janssen

Report no	:	2018.028
Coach	:	dr. ir. N. Tolou, ir. D. Farhadi Machekposhti
Professor	:	prof. dr. ir. J.L. Herder
Specialisation	:	Mechatronic System Design
Type of report	:	Master Thesis
Date	:	14 September 2018



TU Delft Delft University of Technology

Challenge the future

Compliant Remote-Center-of-Motion Mechanism

Optimized for Energy-Dispersive Spectroscopy

By

J.L.H.M. Janssen

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday September 14, 2018 at 9:00 AM.

Student number:	4051823	
Project duration:	November 13, 2017 – September 14, 2018	
Thesis committee:	prof. dr. ir. J.L. Herder	TU Deflt, Chair
	dr. ir. N. Tolou	TU Delft, Supervisor
	ir. D. Farhadi Machekposhti	TU Delft, Coach
	dr. M.K. Ghatkesar	TU Delft
	ir. D.W. Morsink	DENSsolutions

This thesis is confidential and cannot be made public until December 31, 2019.

Preface

In front of you lies the thesis *Compliant Remote-Center-of-Motion Mechanism Optimized for Energy-Dispersive Spectroscopy*. This thesis concludes my time at the Delft University of Technology and was written in partial fulfillment of the requirements to obtain the degree of Master of Science in the field of Mechanical Engineering. This thesis is the result of a collaboration between the Department of Precision and Microsystems Engineering and DENSsolutions.

DENSsolutions, a small company based in Delft and with great people, gave me the opportunity to embark on an interesting journey by providing me with a great project. This thesis would not be lying in front of you without their help. DENSsolutions offered me a warm welcome into their company and provided me with a good atmosphere and all the tools I needed to finish this thesis.

I especially want to thank Diederik Morsink for guiding me through this process and always having the time to answer my questions. His experience in the transmission electron microscope industry was invaluable for this thesis and his remarks during our weekly meetings always made me think of new ways to solve the problems I encountered. He also made sure that I was involved in the daily goings of DENSsolutions which gave me a nice amount of variety.

Furthermore, I want to thank Davood Farhadi Machekposhti for being a great coach. Our bi-monthly meetings always gave me new insights and his sharp, and always on point criticism lifted me to a higher level.

Finally, I want to thank Nima Tolou for being my supervisor and freeing up time for our meetings. During those meetings he always gave me good advice and guided my project in the right direction.

J.L.H.M. Janssen
Delft, August 2018

Contents

1 Introduction

Not for every phenomenon the observations that are made at the macro scale give enough information to explain its workings; often a view of the micro- or even nanoscale is needed. Take the process of age hardening for example. The strengthening of metal alloys by heating them first and then applying rapid cooling is a technique that has been used for centuries. Nevertheless, the exact workings of this process have remained a mystery until recently, when the interactions at the nanoscale of this process have been identified [1]. To view the growth of individual nanoprecipitates, in-situ transmission electron microscopy (TEM) was used while the sample was heated.

In-situ TEM allows for dynamic imaging of a sample with atomic resolution while applying a stimulus to the sample. At DENSsolutions, in-situ sample holders are developed that are used to insert and position the to be observed sample into the TEM (Fig. 1). The sample is placed on a chip that is a microelectromechanical system (MEMS). This MEMS chip adds in-situ capabilities to the sample holder. The in-situ stimuli include; thermal and electric stimulus, gas/liquid environment simulation and mechanical testing.

Next to identifying different structures at the micro scale it is often convenient to discriminate the elements present in the sample. This is done using energy-dispersive X-ray spectroscopy (EDS) [2]. The DENSsolutions sample holder that can facilitate tilt around two axes of the sample (double-tilt) shows underwhelming performance regarding this analysis technique because the tilting mechanism in the tip of the sample holder blocks the required signals. When only a fraction of the signals emitted from the sample is received by the EDS detectors, the speed and accuracy of EDS analysis is reduced.

This thesis focuses on the optimization of the EDS performance of the DENSsolutions double-tilt sample holder. First the influences on the EDS performance in a TEM are identified. Next, the tilting mechanism in the current sample holder is analyzed and the need for a remote-center-of-motion (RCM) mechanism is identified. To define the whole design space, literature on the classification of RCM mechanisms is reviewed and a knowledge gap is indicated. This led to the development of a novel RCM mechanism design strategy. Using this novel design strategy, a large-scale compliant proof-of-principle mechanism is designed that can be used as the tilting mechanism in a double-tilt sample holder. The negative effects of a complaint mechanism on its accuracy are investigated and the mechanism is optimized to increase its accuracy. Finally, a comparison between results that are obtained via analytical, numerical or experimental methods is given and the accuracy of the mechanism is characterized.

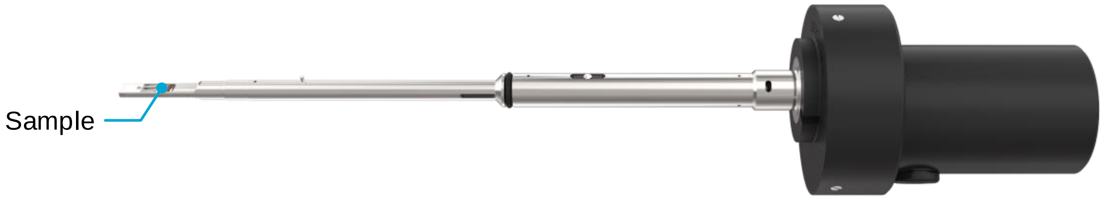


Fig. 1. A DENSsolutions sample holder, which is inserted into the TEM up to the black part, which encapsulates the actuator.

The thesis is structured as follows: Two papers are presented, the first, is a review of the classification and design strategies for RCM mechanisms. The second paper focuses on the design of a RCM mechanism to optimize double-tilt sample holders for EDS. From the results of these two papers conclusions will be drawn and recommendations for further research will be given. An appendix is included that discusses several topics mentioned in the papers in more detail.

2 Paper: Classification of Remote-Center-of-Motion Mechanisms

Classification of Remote-Center-of-Motion Mechanisms

Janeau L.H.M. Janssen, Davood Farhadi Machekposhti, Diederik W. Morsink, and Nima Tolou

***Abstract*—Remote-center-of-motion (RCM) mechanisms are commonly used in applications where limited space is available around the area where motion is required. Applications in which RCM mechanisms have proven to be useful include minimally invasive surgery (MIS), microelectromechanical systems (MEMS) and micro-/nanomanipulators. To aid in the design of RCM mechanisms a novel RCM mechanism classification is introduced that consists of 9 classes: rolling-sliding support, instantaneous center of rotation, circular tracking arcs, parallelograms, synchronous belt transmission, compliant mechanisms, parallel manipulators, spherical linkages and intersecting motion planes. For several classes, RCM mechanism design strategies have been developed which are listed and described. Finally, a structured overview of the different RCM mechanism classes and design strategies is presented to be used as a tool in the design of new RCM mechanisms.**

***Index Terms*—Classification, design strategy, kinematics, remote-center-of-motion.**

I. Introduction

A remote-center-of-motion (RCM) mechanism is a mechanism that is able to implement the spherical rotation of a body around a point, which is distal from any bearing or mechanical supports [1], [2]. The fact that the center of motion is remote from the mechanism, makes RCM mechanisms very useful in applications which have limited space around the area where motion is required.

An example of such an application is the surgical tool used in minimally invasive surgery (MIS). MIS aims to minimize the negative effects of surgery by reducing the size of the incision. This will reduce the wound healing time, the associated pain, and the risk of infection. In Fig. 1 the use of a RCM mechanism in MIS is illustrated; if the RCM of the mechanism lies on the same location as where the surgical gripper enters the abdomen, only a small incision is required while the surgical tool can still be manipulated inside the body of the patient. RCM mechanisms are commonly used in MIS because they offer a higher degree of safety in comparison with robots that do not have a mechanically constrained RCM and thus have a reduced risk of tissue damage [4]. Other fields where RCM mechanisms are used are microelectromechanical systems (MEMS) [5] and micro-/nanomanipulators [6].

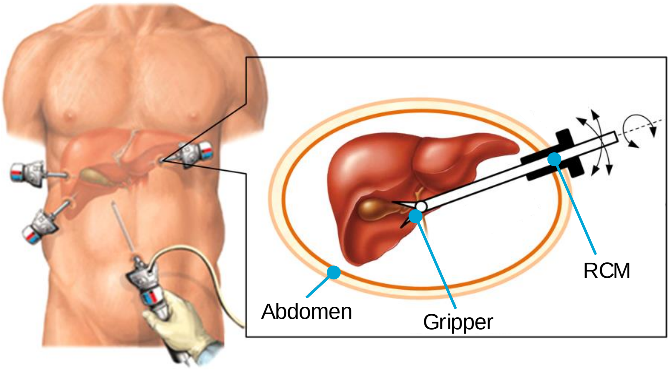


Fig. 1. An illustration of the use of a RCM mechanism in MIS. By having a RCM at the point where the surgical tool (in this case a gripper) enters the abdomen only a very small incision is required while the tool can still be manipulated inside the body. Adapted from [3].

Numerous designs of RCM mechanisms have been introduced which include RCM mechanisms based on circular tracking arcs [5], [7], [8], parallelograms [3], [9]–[13], synchronous belt transmission [14], parallel manipulators [1] and spherical mechanisms [9], [15].

In this paper, a novel classification of RCM mechanisms is introduced. The classification is accompanied by various RCM mechanism design strategies which are described and sorted by their RCM mechanism class. Finally, a structured overview of the different classes and design strategies is presented to be used as a tool in the design of new RCM mechanisms.

II. RCM Mechanism Classification

The most extensive RCM mechanism classification was introduced by Kuo and Dai [4], and the novel RCM mechanism classification will be based upon theirs. Kuo and Dai state that RCM mechanisms can be classified in eight classes: *isocenters*, *circular tracking arcs*, *parallelograms*, *synchronous belt transmission*, *spherical linkages*, *parallel manipulators*, *compliant mechanisms*, and *passive RCM mechanisms*. In contrast to the definition of a RCM mechanism mentioned in the previous chapter, Kuo and Dai define a RCM mechanism as a mechanism that can mechanically decouple rotational and translational motions of tools at a point some distance from the mechanical structure of the mechanism. To adapt Kuo and Dai’s classification to the definition of a RCM mechanism instead of the definition of a chirurgical robot, some classes are removed, and some have to be added. The result is shown in Fig. 2. The classification part of the overview lists the different classes, sorted by their type of motion; planar, spherical or spatial. For every class an example mechanism is shown. The right side of the figure lists the different RCM mechanism design strategies. These strategies are sorted by the RCM mechanism class they belong to and the number of rotational and translation degrees of freedom. The various design strategies will be discussed in the next chapter.

This work was supported by DENSsolutions.

J.L.H.M. Janssen, D. Farhadi Machekposhti and N. Tolou are with the Department of Precision and Microsystems Engineering, Delft University of Technology, Delft 2628 CD, The Netherlands (e-mail: janeaujanssen@gmail.com).

D.W. Morsink is with the Department of Research and Development, DENSsolutions, Delft 2628 ZD, The Netherlands.

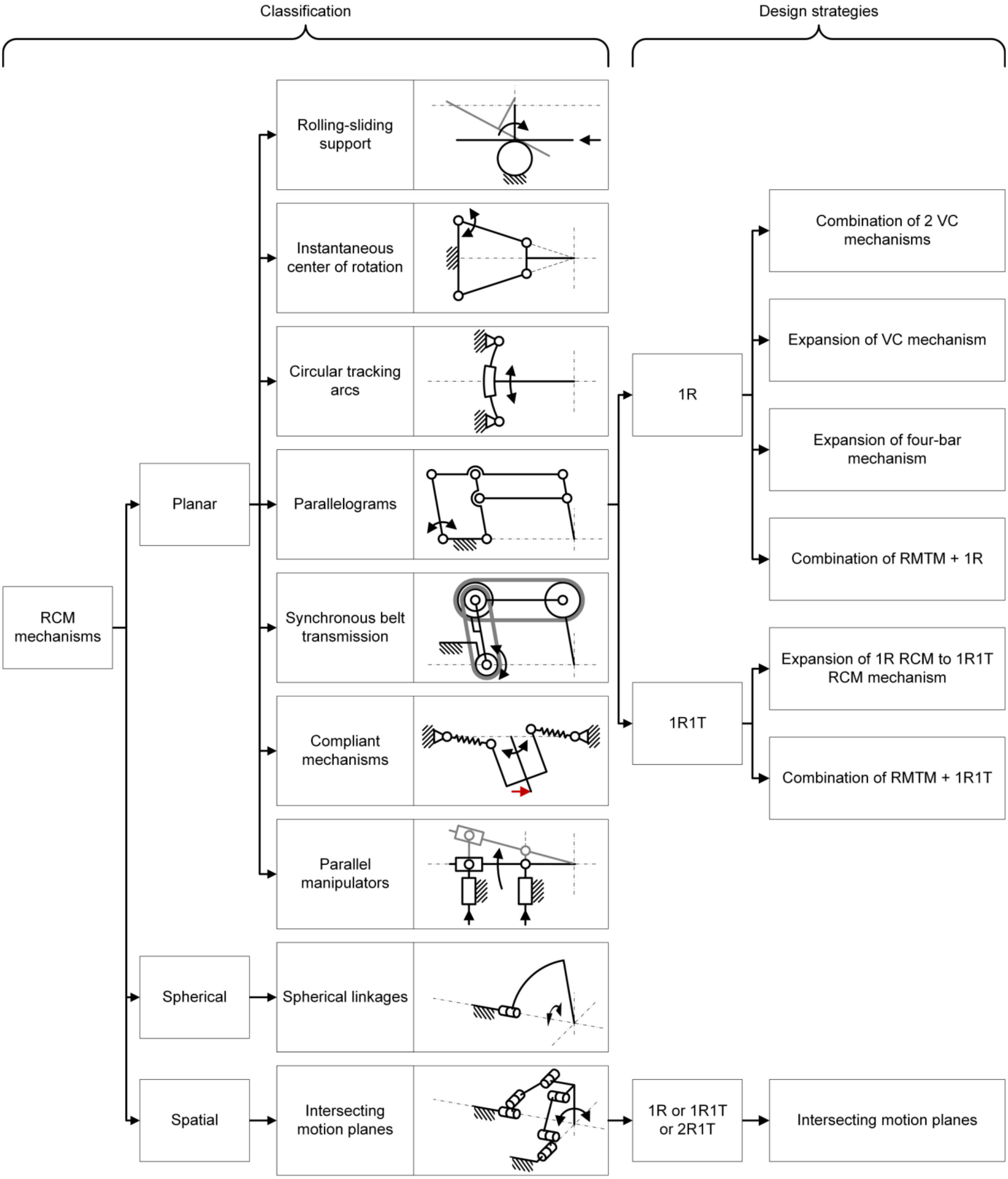

Classification
Design strategies
RCM mechanisms
Planar
Spherical
Spatial
Rolling-sliding support
Instantaneous center of rotation
Circular tracking arcs
Parallelograms
Synchronous belt transmission
Compliant mechanisms
Parallel manipulators
Spherical linkages
Intersecting motion planes
1R
1R1T
1R or 1R1T or 2R1T
Combination of 2 VC mechanisms
Expansion of VC mechanism
Expansion of four-bar mechanism
Combination of RMTM + 1R
Expansion of 1R RCM to 1R1T RCM mechanism
Combination of RMTM + 1R1T
Intersecting motion planes

Fig. 2. An overview of the novel classification and design strategies for RCM mechanisms. The different RCM classes belong to the planar, spherical or spatial mechanisms group. For every class an example mechanism is shown, black arrows indicate movement, red arrows indicate forces. For two classes (the parallelograms and intersecting motion planes classes) design strategies exist which are subdivided by the amount of rotational (R) and translational (T) degrees of freedom.

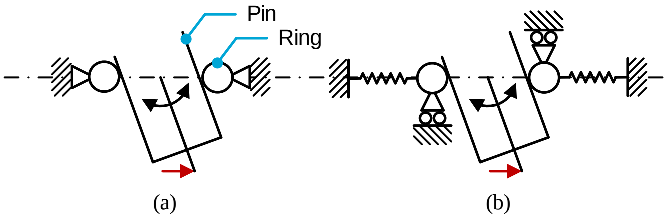


Fig. 3. (a) The planar version of a pin-in-ring joint, which needs play in between the pin and the ring to be able to rotate. (b) The compliant version of a pin-in-ring joint that does not need play to be able to rotate and can be considered a RCM mechanism of the compliant mechanisms class.

The isocenters class [4], defined by a pin-in-ring joint, is replaced by the *rolling-sliding support* class. At its core, a pin-in-ring is identical to two rolling-sliding support mechanisms merged (Fig. 3a). Because the rolling-sliding support class does not have a fixed RCM (it will drift away from the contact point between the beam and the support for higher rotation angles), a pin-in-ring mechanism will need play to be able to rotate. A way to get rid of the play is by using compliant elements as shown in Fig. 3b, but then the mechanism can also be fit in the compliant mechanisms class. So basically, the isocenters class is further simplified into the rolling-sliding support class.

The passive RCM mechanisms class (Fig. 4) is removed from the classification because it is identical to the rolling-sliding support class. The example mechanism of the passive RCM mechanisms class shown in Fig. 4 uses a robotic manipulator arm to rotate the laparoscope, but its RCM is defined by the abdominal wall which acts as a rolling-sliding support for the laparoscope.

Next to the exclusion of these classes two new classes are added; *intersecting motion planes* and *instantaneous center of rotation*. The intersecting motion planes class, as the name already suggests, consists of mechanisms that restrict their motion by coupling two or more motion planes. An example is given in Fig. 2 and a more detailed description is given when the design strategy is discussed in the next chapter.

The instantaneous center of rotation class is based on mechanisms that have a link that rotates around a certain point, the instantaneous center of rotation (ICR), for a certain position of the mechanism. The ICR of the link is only a RCM for small rotations of the mechanism. A parasitic translation of the ICR is induced at higher rotations. Although this is not a perfect RCM mechanism it can be useful for high precision mechanisms that have a low rotation range requirement.

Concluding, the novel classification of RCM mechanisms consists of the following 9 classes: *Rolling-sliding support*, *instantaneous center of rotation*, *circular tracking arcs*, *parallelograms*, *synchronous belt transmission*, *compliant mechanisms*, *parallel manipulators*, *spherical linkages* and *intersecting motion planes*.

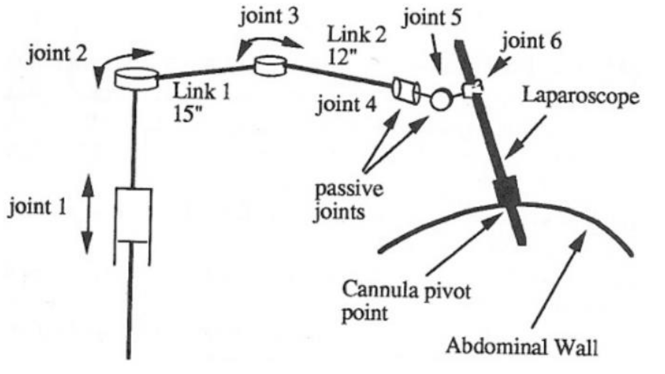


Fig. 4. An example of a mechanism from the passive RCM mechanisms class. The RCM is defined by the support of the abdominal wall. [16]

III. RCM Mechanism Design Strategies

Now the analysis on RCM mechanisms has been completed in the form of a classification it is crucial to review the various design strategies that exist for RCM mechanisms. Six strategies have been proposed (Fig. 2). The core idea of every RCM mechanism is that a body rotates around a certain point, the RCM. In its most elemental form this results in having two points that have circular motion paths which are concentric. Two cases can be distinguished (Fig. 5). Case I is defined by Zong et al. [1]: If any two distinct points in a rigid body can move along concentric circles, and the two points are not collinear with the center, then the rigid body must accomplish a circular motion whose center is coincident with the center of the concentric circles. Next to the non-collinear case I, a body can still rotate around a point when two points on it are collinear with this point. That is, as long as the two circular motion paths of the points have different radii. This defines case II (Fig. 5b).

Zong et al. [1] developed two RCM mechanism design strategies. The first one is based on combining two virtual center (VC) mechanisms (mechanisms that have a point which describes a circular motion) into one mechanism to create a RCM mechanism. Two RCM mechanisms created with this strategy that are based on case I are shown in Fig. 6, two RCM mechanisms that are based on case II are shown in Fig. 7.

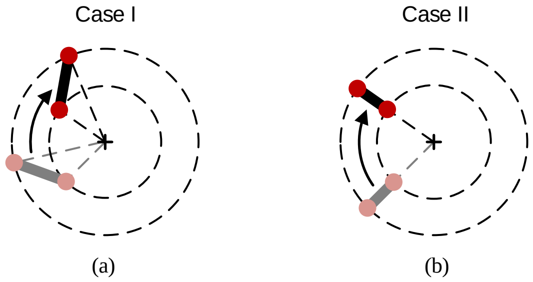


Fig. 5. The two cases in which a body can rotate around a point. (a) Case I, in which two points on a body are not collinear with the center of the concentric circles and where the circles can have arbitrary radii and (b) case II, in which two points on a body are collinear with the center of the concentric circles and the radii of the circles are not equal.

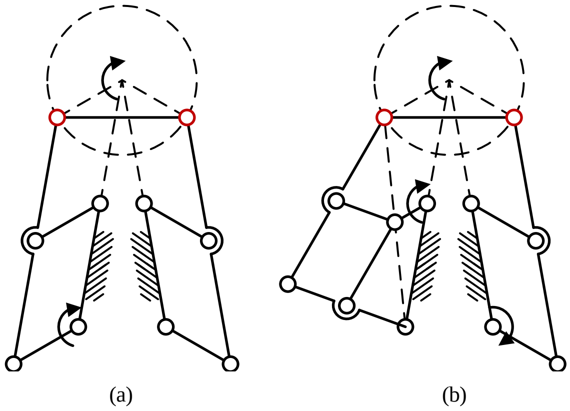


Fig. 6. Two RCM mechanism based on case I, created by using two VC mechanisms; one (a) created with two identical VC mechanisms and another one (b) created with two different VC mechanisms. The points (hinges) that describe circular motion are indicated in red.

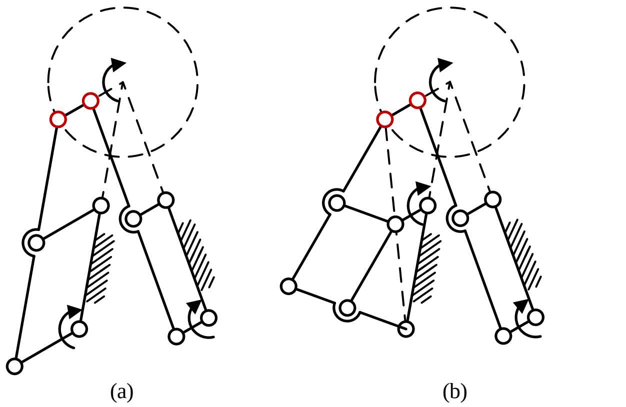


Fig. 7. Two RCM mechanism based on case II, created by using two VC mechanisms; one (a) created with two identical VC mechanisms and another one (b) created with two different VC mechanisms. The points (hinges) that describe circular motion are indicated in red.

The second strategy is based on expanding a VC mechanism to a RCM mechanism (Fig. 8). The basis of the strategy is a VC mechanism with its future RCM at O, which is the center of the circular motion path described by E. An output link is added between E and O. By grounding point O, the amount of degrees of freedom (DOF) will not increase. Next, a link or mechanism is connected between the input link and the output link. If the added link or mechanism does not alter the DOF of the mechanism, it can be converted into a RCM mechanism. Depending on the relation between the input link and output link (parallel, line-symmetry, center-symmetric, constant angle) different mechanisms are used. Finally, the constraint at O is removed and a RCM mechanism is created.

Janssen et al. [17] introduced a RCM mechanism design strategy that uses the very common four-bar mechanism as its starting mechanism (Fig. 9). In four steps, an arbitrary four-bar mechanism can be converted into its RCM equivalent. Step 1: start off with an arbitrary four-bar mechanism. The input link, output link and the RCM location of the final RCM mechanism are link AB, link CD and point D respectively. Step 2: the shape of the four-bar mechanism is scaled into the corner of B. In Fig. 9 the four-bar shape is scaled down which will make sure that the RCM will lie outside of the RCM mechanism. But scaling up is also an option, although this makes sure that the RCM is inside the contours of the RCM mechanism. Step 3: two links are added that create two parallelograms; link EF connects link AD to link A’D’ and is parallel to link AB, link GH connects link CD to link C’D’ and is parallel to link BC. Step 4: finally, the constraint at D is moved to F and the link between F and D is removed. A RCM mechanism is created with input links AB and EF, output link CD and its RCM at D. This design strategy is different from the other ones mentioned here because it gives the option to tune the rotation ratio between the input and output link, while for other design strategies this ratio is always 1. This rotation ratio is dependent on the shape of the initial four-bar shape.

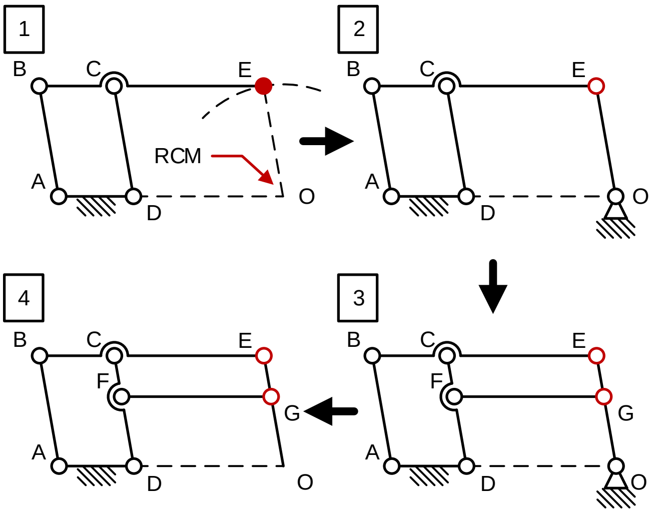


Fig. 8. A RCM mechanism design strategy that is based on the expansion of a VC mechanism to a RCM mechanism. 1. The basis of the design strategy is a VC mechanism with its future RCM defined as the center point (O) of the circle described by E. 2. A link is added between E and O. 3. A link or mechanism is added in between the input link CD and output link EO. The type of mechanism that is added depends on the relation between the input and output link. 4. The constraint at O is removed. The two points as described in Fig. 5 are indicated in red. Adapted from [1].

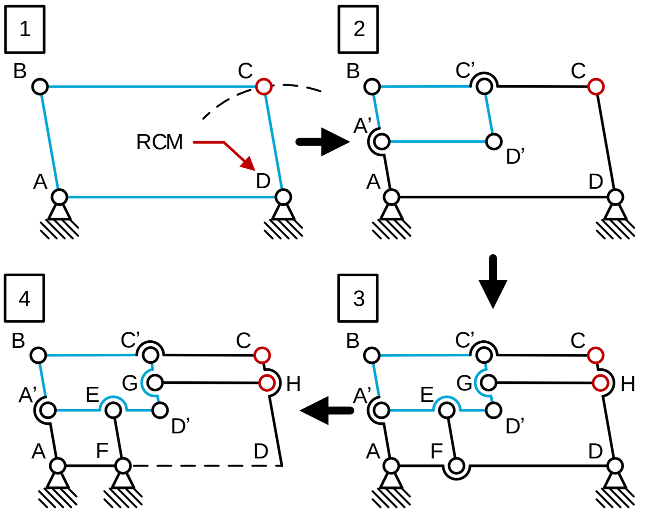


Fig. 9. The expansion of a four-bar mechanism design strategy. 1. The basis of the design strategy is an arbitrary four bar mechanism, with its future RCM defined as point D. 2. The shape of the four-bar mechanism is scaled down into the corner of B. 3. Two links are added; link EF which is parallel to link AB that connects the scaled four bar mechanism with the original one. And link GH which is parallel to link BC that also connects the scaled four bar mechanism with the original one. 4. Finally the constraint at D moved to F and link FD is removed. The two points as described in Fig. 5 are indicated in red.

Long et al. [13] developed a design strategy that makes use of a mechanism that mimics the 3 DOF motion of one of its links with another link, called a rigid motion tracking mechanism (RMTM) (Fig. 10). When the input link of a RMTM is combined with a 1R1T mechanism, the output link of the RMTM will show the same 1R1T behavior but without the physical revolute joint. By combining a 1R mechanism instead of a 1R1T mechanism with the RMTM a 1R RCM mechanism can be created.

Another design strategy that can be used to create 1R1T

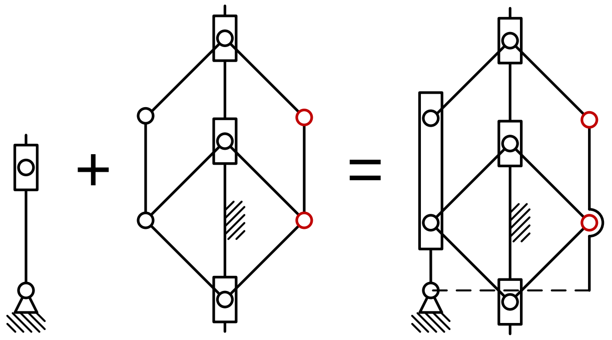


Fig. 10. An illustration of the strategy of combining a 1R1T mechanism and a RMTM to create a 1R1T RCM mechanism. By using a 1R mechanism and a RMTM mechanism a 1R RCM mechanism can be created. Adapted from [13].

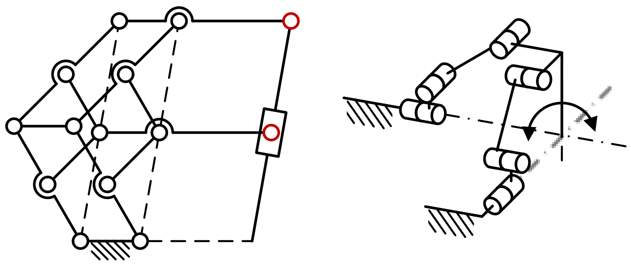


Fig. 11. (a) A 1R RCM mechanism (dashed lines) that is expanded to a 1R1T RCM mechanism. (b) A 1R RCM mechanism based on intersecting motion planes. The output link has a parasitic translation through its RCM when it is rotated.

RCM mechanisms is developed by Li et al. [3], their strategy uses a 1R RCM mechanism as the starting point and expand it to a 1R1T mechanism (Fig. 11a). The core of this mechanism is identical to the RCM mechanism shown in Fig. 8, but by adding a mechanism that gives the possibility to vary link lengths the translation functionality is added.

All the previous strategies are used to create RCM mechanisms that fit in the parallelograms class. One other strategy [18] is developed which can be used to create RCM mechanisms of the intersecting motion planes class (Fig. 11b). The strategy is based on combining two planar mechanisms that have more than 1 DOF in such a way that their planes are not parallel. This reduces the movement space of the mechanism from two motion planes to the intersection of these two planes; a line. If both motion planes are given an axis around which they can rotate, and these two axes intersect somewhere on the intersection line of the two planes, this intersection point becomes a RCM. Depending on the DOF of the planar mechanisms that are used either a 1R, 1R1T or a 2R1T RCM mechanism can be created.

IV. Conclusion

Existing RCM mechanisms have been reviewed and a novel classification of RCM mechanisms has been introduced. Furthermore, the existing RCM mechanism design strategies have been listed, described and sorted by the RCM mechanism class they belong to. Finally, a schematic overview of the RCM mechanism classification and different design strategies is created.

V. References

[1] G. Zong, X. Pei, J. Yu, and S. Bi, “Classification and type synthesis of 1-DOF remote center of motion mechanisms,” *Mech. Mach. Theory*, vol. 43, pp. 1585–1595, 2008.

[2] J. F. Jensen, “Remote center positioner with channel shaped linkage element,” US 5800423, 1998.

[3] J. Li, G. Zhang, Y. Xing, H. Liu, and S. Wang, “A Class of 2-Degree-of-Freedom Planar Remote Center-of-Motion Mechanisms Based on Virtual Parallelograms,” *J. Mech. Robot.*, vol. 6, no. 3, p. 031014, Jun. 2014.

[4] C.-H. Kuo and J. S. Dai, “Robotics for Minimally Invasive Surgery: A Historical Review from the Perspective of Kinematics,” in *International symposium on history of machines and mechanisms*, 2009, pp. 337–354.

[5] C. P. Lusk and L. L. Howell, “Spherical Bistable Micromechanism,” *J. Mech. Des.*, vol. 130, no. 4, p. 045001, Apr. 2008.

[6] J. Qu, W. Chen, J. Zhang, and W. Chen, “A piezo-driven 2-DOF compliant micropositioning stage with remote center of motion,” *Sensors Actuators A Phys.*, vol. 239, pp. 114–126, Mar. 2016.

[7] J. O. Jacobsen, B. G. Winder, L. L. Howell, and S. P. Magleby, “Lamina Emergent Mechanisms and Their Basic Elements,” *J. Mech. Robot.*, vol. 2, no. 1, p. 011003, Feb. 2010.

[8] H. M. Yip, P. Li, D. Navarro-Alarcon, Z. Wang, and Y. Liu, “A new circular-guided remote center of motion mechanism for assistive surgical robots,” in *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, 2014, pp. 217–222.

[9] R. Baumann, W. Maeder, D. Glauser, and R. Clavel, “The PantoScope: a spherical remote-center-of-motion parallel manipulator for force reflection,” in *Proceedings of International Conference on Robotics and Automation*, 1997, vol. 1, pp. 718–723.

[10] M. Fontana, A. Dettori, F. Salsedo, and M. Bergamasco, “Mechanical design of a novel Hand Exoskeleton for accurate force displaying,” in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 1704–1709.

[11] R. Beira, L. Santos-Carreras, G. Rognini, H. Bleuler, and R. Clavel, “Dionis: A Novel Remote-Center-of-Motion Parallel Manipulator for Minimally Invasive Surgery,” *Appl. Bionics Biomech.*, vol. 8, no. 2, pp. 191–208, 2011.

[12] C. Freschi, V. Ferrari, F. Melfi, M. Ferrari, F. Mosca, and A. Cuschieri, “Technical review of the da Vinci surgical telemanipulator,” *Int. J. Med. Robot. Comput. Assist. Surg.*, vol. 9, no. 4, pp. 396–406, Dec. 2013.

[13] H. Long, Y. Yang, X. Jingjing, and S. Peng, “Type Synthesis of 1R1T Remote Center of Motion Mechanisms Based on Pantograph Mechanisms,” *J. Mech. Des.*, vol. 138, no. 1, p. 014501, Nov. 2015.

[14] D. Stoianovici, L. L. Whitcomb, D. Mazilu, Russel H. Taylor, and L. R. Kavoussi, “Remote center of motion robotic system and method,” US 7021173 B2, 04-Apr-2006.

[15] C.-B. Chng, B. Duan, and C.-K. Chui, “Modeling and simulation of a Remote Center of Motion mechanism,” in *2016 IEEE Region 10 Conference (TENCON)*, 2016, pp. 1755–1758.

[16] J. M. Sackier and Y. Wang, “Robotically assisted laparoscopic surgery,” *Surg. Endosc.*, vol. 8, no. 1, pp. 63–66, Jan. 1994.

[17] J. L. H. M. Janssen, D. Farhadi Machekposhti, D. Morsink, and N. Tolou, “Compliant Remote-Center-of-Motion Mechanism Optimized for Energy Dispersive Spectroscopy,” 2018.

[18] J. Li, G. Zhang, A. Müller, and S. Wang, “A Family of Remote Center of Motion Mechanisms Based on Intersecting Motion Planes,” *J. Mech. Des.*, vol. 135, no. 9, p. 091009, Jul. 2013.

3 Paper: Compliant Remote-Center-of-Motion Mechanism Optimized for Energy Dispersive Spectroscopy

Compliant Remote-Center-of-Motion Mechanism Optimized for Energy Dispersive Spectroscopy

Janeau L.H.M. Janssen, Davood Farhadi Machekposhti, Diederik W. Morsink, and Nima Tolou

***Abstract*—In situ sample holders with double-tilting capabilities are used to insert and position samples inside a transmission electron microscope (TEM) for dynamic imaging. However, the performance of these sample holders, regarding energy-dispersive X-ray spectroscopy (EDS) analysis, is not on par with their single-tilt counterparts. By analyzing the EDS influences and the tilting mechanism of double-tilt sample holders, the need for a remote-center-of-motion (RCM) mechanism as a tilting mechanism is identified. Because existing RCM mechanism design strategies limit the design flexibility, a novel design strategy is developed. The novel strategy gives more flexibility in terms of the use of space, design for stiffness and it gives variable input/output link rotation ratio functionality. A compliant proof-of-principle mechanism, which is designed using the novel RCM mechanism design strategy, is manufactured to characterize the accuracy, force/displacement behavior and the input/output link rotation ratio. Analytical, numerical and experimental results are compared, and it can be concluded that the compliant RCM mechanism has potential to be used in double-tilt sample holders.**

***Index Terms*—Compliant mechanisms, design strategy, energy-dispersive X-ray spectroscopy, remote-center-of-motion, sample holder.**

I. Introduction

Microscopy has given us a better understanding of the world around us. The electron microscope specifically, has been invaluable to the fields of medicine, biology, chemistry and physics for the last century. To keep on pushing the boundaries of research using electron microscopes, new techniques are being developed. One of the more recent advances is in-situ transmission electron microscopy (TEM) [1], [2]. In situ TEM allows for dynamic imaging of a sample with atomic resolution by applying a stimulus to the sample. The effect of the stimulus can be observed in real time, resulting in a better understanding of the fundamental behavior of materials at the nanoscale. The stimulus effects are enabled by microelectromechanical systems (MEMS), and consist of, but are not limited to, thermal and electric stimulus, gas/liquid environment simulation and mechanical testing. Because of the limited space inside a transmission electron microscope, the small footprint of MEMS means that they are ideal for in situ TEM [3].

To insert and position the sample and MEMS chip in the TEM, a sample holder is used (Fig. 1) [4, Ch. 8]. Numerous different designs have been developed [3], [5]–[11] that can facilitate static or dynamic imaging and can have single- or double-tilt motion, depending on the type of experiment and analysis technique. The type of sample holder that is treated in this paper is the dynamic, double-tilt variant and is developed by DENSsolutions (Fig. 1).

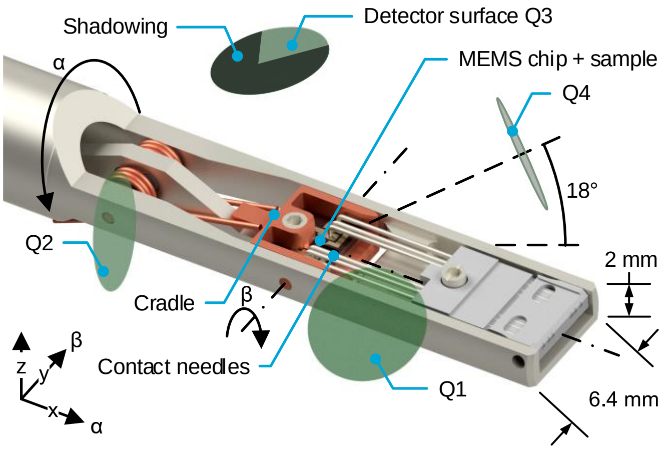


Fig. 1. The DENSsolutions double-tilt sample holder. The MEMS chip with the sample on top is placed inside the cradle. Contacts on the MEMS chip are connected to the control hardware via the contact needles. The four circular surfaces (Q1-Q4) are the EDS detector windows of a FEI Super X detector. The sample holder facilitates tilt of the MEMS chip and sample around the x-axis (α-tilt) and around the y-axis (β-tilt). Due to the low take-off angle of the detectors (~18°), the cradle and contact needles will block the EDS signal coming from the sample, keeping it from reaching the EDS detectors. This results in EDS shadowing on the detector surfaces (only shown for Q3 in this figure).

The sample preparation technique that is commonly used with this sample holder, is focused ion beam (FIB) milling [12] because this technique has a high β-tilt requirement (±10°). The tilting capabilities of the sample holder are used to align the crystal structure of the sample with the incident electron beam, this is done to optimize the quality of the sample image.

Next to the primary use of a TEM to create an image of the sample, additional analytical techniques can be used. A technique that is commonly used to identify the material composition of a sample is energy-dispersive X-ray spectroscopy (EDS). Single-tilt sample holders can get good results using this technique, but the performance of double-tilt sample holders using in-situ TEM is not on par. The problem of the double-tilt sample holder is that a percentage of the EDS signal coming from the sample is blocked from reaching the EDS detector surfaces. The two main obstructions are the cradle and the contact needles (Fig. 1). When only a fraction of the EDS signal is received by the EDS detectors, the speed and/or accuracy of the EDS analysis will be reduced. To solve this problem, ways to optimize the EDS performance of the double-tilt sample holder are investigated, a novel RCM mechanism design strategy is introduced, and a compliant proof-of-principle mechanism is tested and characterized.

This work was supported by DENSsolutions.

J.L.H.M. Janssen, D. Farhadi Machekposhti and N. Tolou are with the Department of Precision and Microsystems Engineering, Delft University of Technology, Delft 2628 CD, The Netherlands (e-mail: janeaujanssen@gmail.com).

D.W. Morsink is with the Department of Research and Development, DENSsolutions, Delft 2628 ZD, The Netherlands.

II. EDS

All the imaging modes of an electron microscope use signals emitted from the sample as a result of the interaction between the incident electron beam and the atoms in the sample. The signal that is used to identify the material composition of the sample using EDS, consists of the characteristic X-rays. The energy level of these X-rays is unique to the atom, so by measuring the energy of the emitted X-ray, the element it came from can be identified. The energy level measurement is done by an EDS detector; for every characteristic X-ray that enters the detector through its window, the energy is measured. The different energy levels of the characteristic X-rays is then counted and a spectrum is obtained that gives information about the elements in the sample. [4], [13]

The rule of thumb for EDS analysis is that the higher the number of counts, the more accurate the spectrum will be, which will result in a higher accuracy of identifying the right elements. The number of counts, driven by different influences, is not the only factor that drives the accuracy of the EDS analysis, therefore a detailed overview is created (Fig. 2). [4], [14]–[16]

The six main influence groups are:

- Detector type/window (yellow)
- Detector errors (purple)
- X-ray absorption (green)
- Desired collection angle (black)
- Undesired collection angle (red)
- Background noise (blue)

In this case, all influences up to the point that the signal exits the detector and enters the electronics and data processing part are taken into consideration. For a more detailed description of the EDS influences see appendix A.

Not all the influences on the EDS signal are relevant for the design of the sample holder. Therefore, the detailed EDS influences overview can be filtered to only include the influences through which the sample holder design can limit the amount and quality of the EDS signal. These influences are indicated by the solid bubbles in Fig. 2. Only *sample holder geometry*, *energy scatter* and *sample drift* are affected by the sample holder design and *sample holder angle* is affected by how the sample holder is used, so these are the "crucial" influences that should be taken into consideration in a new sample holder design. From those four influences, the most important one is the sample holder geometry. Because EDS detectors generally have low take-off angles (~18°), the cradle of the β-tilting mechanism and the MEMS chip contact needles in the tip of the sample holder will block the EDS signal from reaching the EDS detector (Fig. 1, Fig. 3). This "shadowing" effect will reduce the detector illumination to less than 30 percent, dramatically reducing the effectiveness of EDS analysis. For lower α-tilt angles, even worse illumination percentages are observed, dropping to 13 percent illumination at 0° of α-tilt.

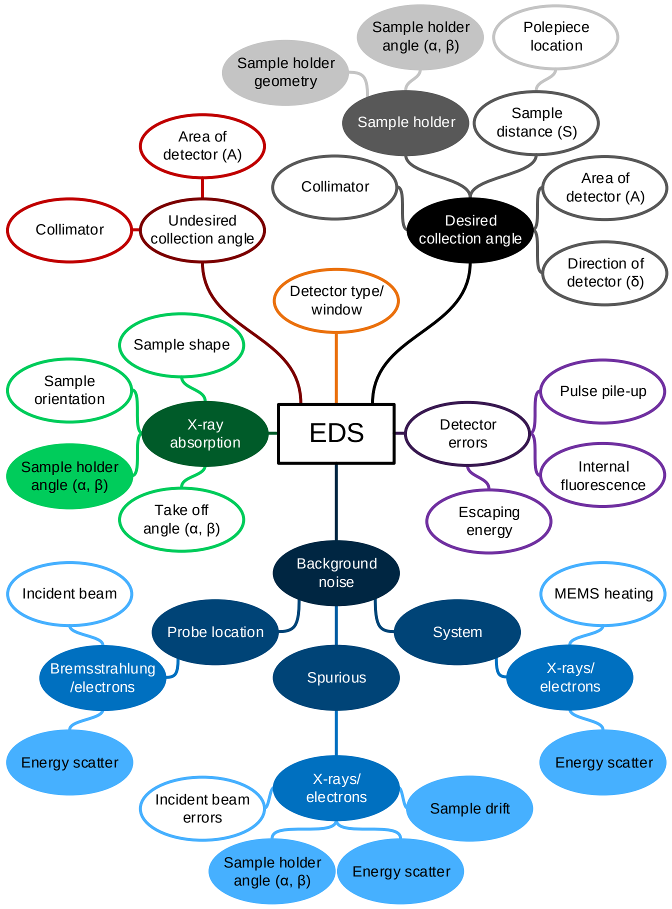


Fig. 2. A detailed overview of the EDS influences up to the point that the signal exits the detector and enters the electronics and data processing part. The overview shows through which paths the influences in the outermost bubbles have an effect on the performance of the EDS analysis. The influences and their paths that are relevant for the sample holder design are indicated by the solid bubbles.

III. Tilting Mechanism

As discussed in the previous chapter, the β-tilting mechanism and the contact needles are the main causes for the subpar EDS performance of the DENSsolutions sample holder. The EDS shadowing because of the contact needles can be reduced by moving the contact points on the MEMS chip further away from the sample location. The EDS shadowing due to the tilting mechanism is more difficult to reduce. First, the current tilting mechanism (Fig. 4) that is used in the DENSsolutions sample holder is investigated.

The core requirement of the current tilting mechanism is that

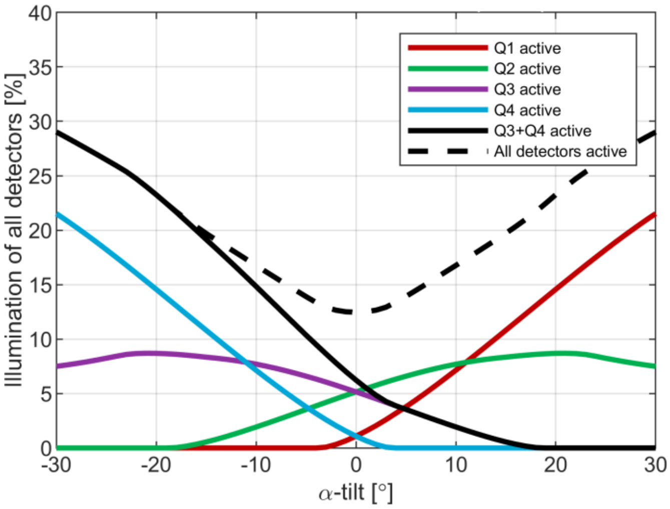


Fig. 3. The illumination percentage of the EDS detector, which shows the percentage of the total surface area of the detector that is used to detect characteristic X-rays, at different α-tilt angles, and for different configurations of active detectors. Valid for the DENSsolutions double-tilt sample holder. Detectors Q2 and Q3 are shadowed by the cradle, detectors Q1 and Q4 are shadowed by the contact needles.

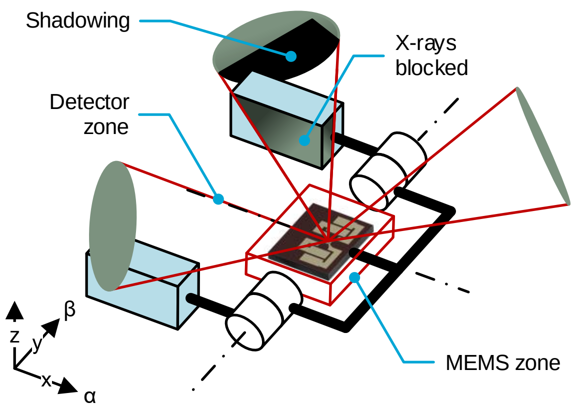


Fig. 4. An illustration of the RCM mechanism that facilitates the β-tilt in the DENSsolutions double-tilt sample holder. The mechanism does not interfere with the MEMS zone, but it does interfere with the detector zones, blocking X-rays from reaching the EDS detector. For clarity only three out of four EDS detectors are illustrated.

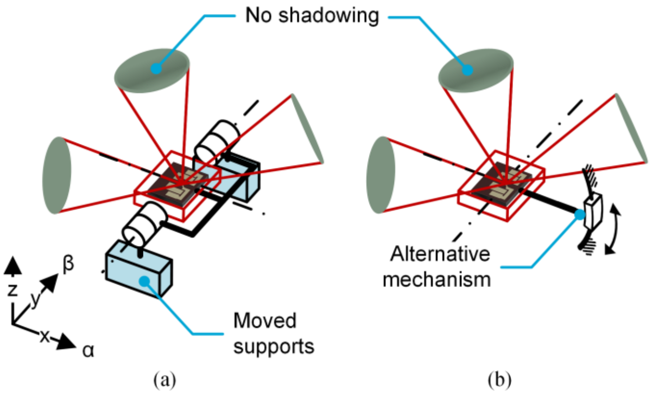


Fig. 5. (a) Optimization path I; reducing or changing the footprint of the current mechanism to make sure the tilting mechanism does not interfere with both the MEMS zone and the detector zones. (b) Optimization path II; using another type of RCM mechanism that does not interfere with the MEMS zone and the detector zones. A goniometer mechanism is shown here. For clarity only three out of four EDS detectors are illustrated.

it facilitates β-tilt of the sample without interfering with the MEMS chip. This means that the mechanism should be situated outside a certain volume; a mechanism free zone. Such a mechanism is called a remote-center-of-motion (RCM) mechanism and its definition is; a mechanism that is able to implement the spherical rotation of a body around a point, which is distal from any bearing or mechanical supports [17], [18]. In the DENSsolutions sample holder design, the mechanism free zone encapsulates the MEMS chip, so it fulfills the above-mentioned core requirement. But for an EDS optimized sample holder there is the additional requirement of no EDS detector shadowing. This requirement adds 4 additional mechanism free zones (detector zones) to the initial zone defined by the MEMS chip (MEMS zone) (Fig. 4). This means the current tilting mechanism will interfere with the detector zones, blocking the X-rays coming from the sample.

To optimize the tilting mechanism for EDS, it should not interfere with any of the five mechanism free zones. There are two optimization paths that can fulfill this requirement. Optimization path I is changing the current tilting mechanism or reducing its footprint to make sure it does not interfere with the detector zones anymore (Fig. 5a). Optimization path II is using another type of RCM mechanism that is easier to position outside of the mechanism free zones. An example of another RCM mechanism that can be used, in this case a goniometer mechanism, is shown in Fig. 5b. In recent years optimization path I has been followed, with minor improvements in EDS performance as a result, therefore, in the rest of this paper, the focus will be on optimization path II and the development of an alternative RCM mechanism for the double-tilt sample holder.

Next to the EDS optimization, the tilting mechanism also has requirements for standard use. The critical requirements are listed in Table 1. Next to the fact that the tilting mechanism should be able to operate in a very high vacuum, the small workspace, low sample drift and very high stability requirements are challenging to meet.

TABLE 1
CRITICAL REQUIREMENTS

Requirement	Value
Workspace	Ø 6.5 mm x 17 mm
β rotation range	> ±10°
Sample drift	< ±50 µm
MEMS chip stability	< 1 nm/min
Vacuum compatibility	< 10^{-6} Pa

IV. Novel RCM Mechanism Design Strategy

Using the RCM mechanism classification [19] and various RCM mechanism design strategies [18], [20]–[23], concepts for the sample holder tiling mechanism have been developed. Exact details and evaluation of these concepts can be found in

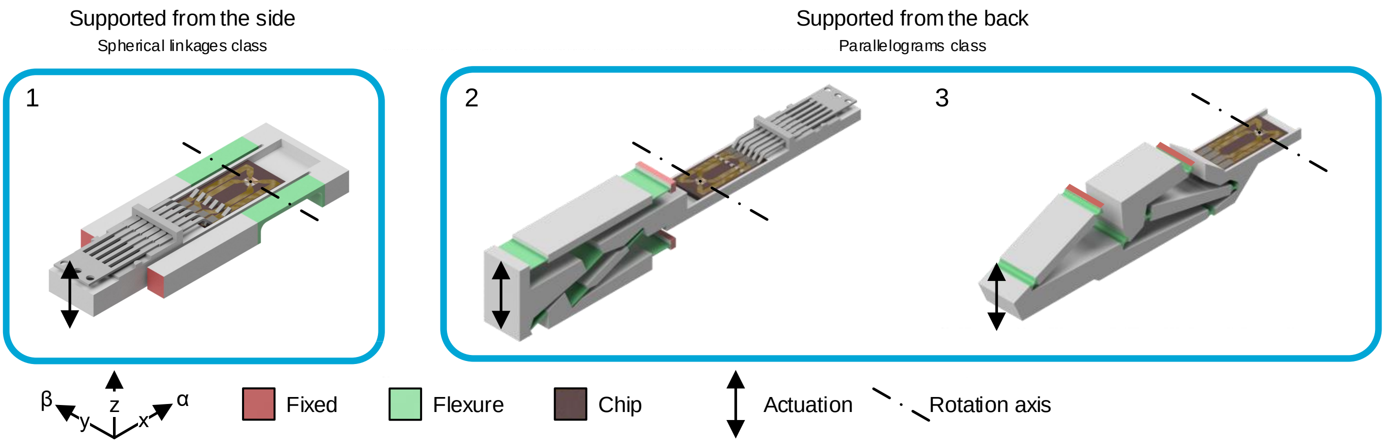


Fig. 6. Tilting mechanism concepts. The concepts can be divided into two groups; (left) concepts that replace the conventional revolute joints in the DENSsolutions sample holder with flexures so that the MEMS chip is supported from the side, or (right) concepts that support the MEMS chip from the back, creating more space around the MEMS chip. The first group consists of RCM mechanisms from the spherical linkages class and the second group consists of RCM mechanisms from the parallelograms class.

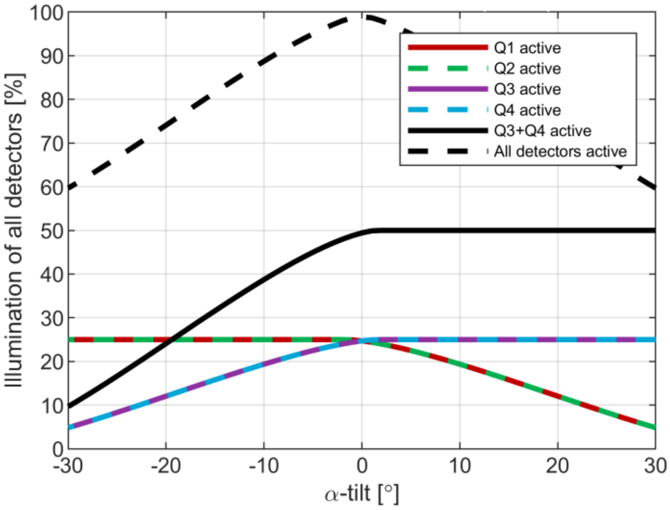


Fig. 7. The detector illumination as a function of α-tilt for concept 1, 2 and 3. At 0° of α-tilt the illumination goes up to 100%, which is a sevenfold increase in comparison with the current sample holder design. For larger α-tilt angles this number drops down to a twofold increase in EDS illumination.

appendix B. The concepts that have the highest potential are shown in Fig. 6 and can be split into two groups; concepts that support the MEMS chip from the side and concepts that support the MEMS chip from the back. The concepts in the second group are preferred because a back support leaves more space around the MEMS chip and thus more flexibility for its design. More space around the MEMS chip also means that higher α-tilt angles can be reached. The benefit of concept 1 is its small footprint and simplicity but its design is suboptimal.

What all concepts have in common is that they have identical EDS performance, and this performance is a big improvement in comparison with the current sample holder design (Fig. 7). At 0° α-tilt the detector illumination is 100%, which is a more than sevenfold increase in comparison with the current sample holder design. For larger α-tilt angles the illumination drops down to 60%, but this is still a twofold increase in EDS illumination.

The drawback of concepts like concept 2 and 3 are that they are difficult to fit into the limited space available. They consist of multiple links and hinges, and those links have constraints on them that limit the design flexibility. If a look is taken at the double parallelogram RCM mechanism in Fig. 8a, which is also used for the concepts in group 2, it can be seen that its input links (in blue) and output link (in red) are parallel and of equal length. These two constraints need to stay intact to have a mechanism with the RCM property. If the output link is not parallel or of equal length to the input links (Fig. 8b), the parallelogram shape of CEGF is lost and thus also the RCM property of the mechanism. Because of these two constraints the flexibility to design the RCM mechanism for a certain shape or stiffness direction is lowered. All design strategies for parallelogram-based RCM mechanisms suffer from this same drawback, therefore a novel RCM mechanism design strategy is developed.

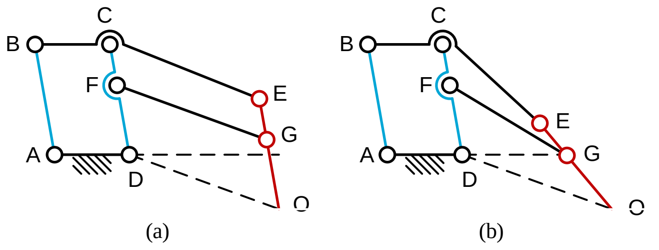


Fig. 8. Two variations on the standard double-parallelogram RCM mechanism with its input links in blue and its output link in red. (a) The standard mechanism which has the RCM property and (b) a mechanism with a rotated output link with respect to its input links which has lost its RCM property.

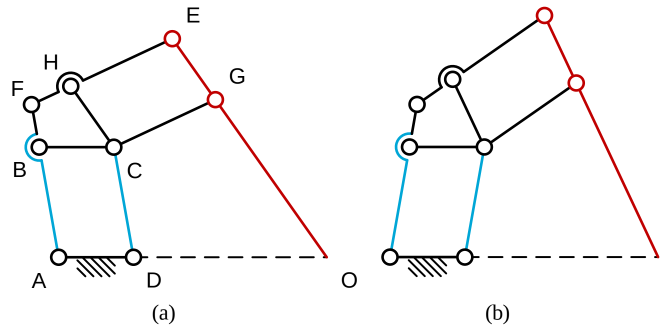


Fig. 9. A RCM mechanism that has input (in blue) and output links (in red) whose lengths are non-equal, and which are non-parallel. Two configurations are shown; (a) the initial and (b) the rotated configuration. Adapted from [24].

A RCM mechanism that does not have the parallel and equal length constraint on its input and output links is developed by Liu et al. [24] (Fig. 9). Based on this mechanism a novel RCM mechanism design strategy is developed (Fig. 10). In four steps, an arbitrary four-bar mechanism can be converted into its RCM equivalent. Step 1: start off with an arbitrary four-bar mechanism. The input link, output link and the RCM location of the final RCM mechanism are link AB, link CD and point D respectively. Step 2: the shape of the four-bar mechanism is scaled into the corner of B. In Fig. 10 the four-bar shape is scaled down which will make sure that the RCM will lie outside of the RCM mechanism. But scaling up is also an option, although this makes sure that the RCM is inside the contours of the mechanism. Step 3: two links are added that create two parallelograms; link EF connects link AD to link A'D' and is parallel to link AB, link GH connects link CD to link C'D' and is parallel to link BC. Step 4: finally, the constraint at D is moved to F and the link between F and D is removed. A RCM mechanism is created with input links AB and EF, output link CD and its RCM at D.

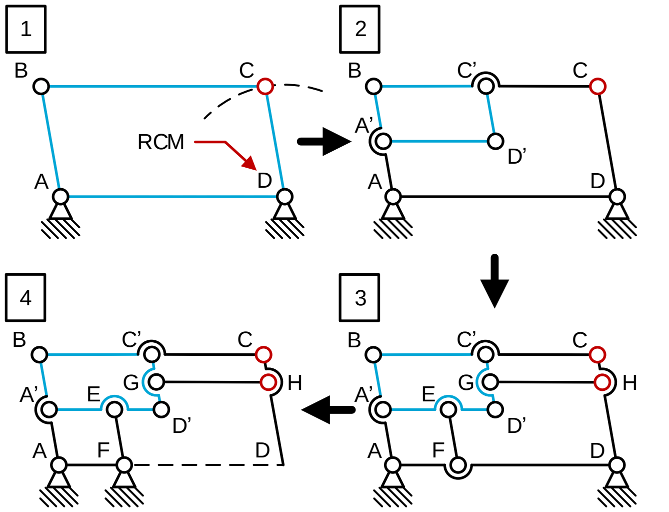


Fig. 10. The novel RCM mechanism design strategy. 1. The basis of the design strategy is an arbitrary four-bar mechanism, with its future RCM defined as point D. 2. The shape of the four-bar mechanism is scaled down into the corner of B. 3. Two links are added; link EF which is parallel to link AB that connects link AD with link A'D'. And link GH which is parallel to link BC that connects link CD with link C'D'. 4. Finally the constraint at D is moved to F and link FD is removed.

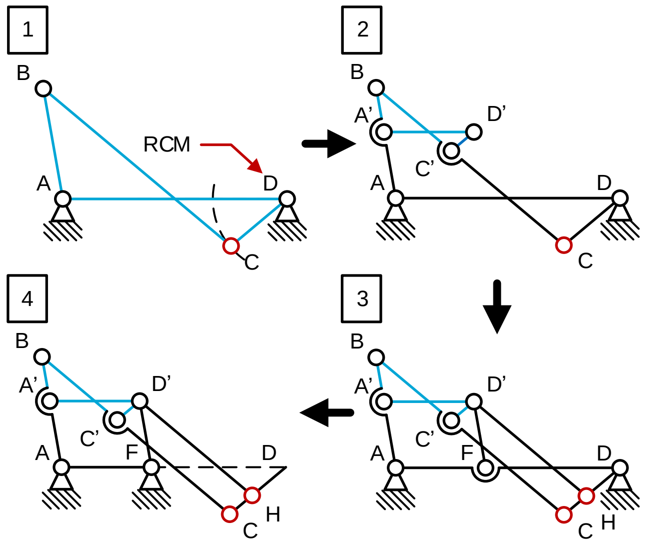


Fig. 11. The novel RCM mechanism design strategy with a more complex four-bar mechanism as its starting point. In this case the hinge points E and G of Fig. 10 are merged with D'.

Of course, this mechanism still has input and output links that are parallel and of equal length, but this is no longer a constraint. If a four-bar mechanism with a different shape is converted into its RCM mechanism equivalent (Fig. 11) it can be seen that the parallel and equality constraints are no longer required for the RCM property of the mechanism.

While the rotation angle of the input and output links are equal for RCM mechanisms created by existing RCM mechanism design strategies, the ratio between the output and input rotation angle (Φ_{out}/Φ_{in}) for the novel design strategy depends on the initial four-bar shape. In the case of the RCM mechanism in Fig. 10 this ratio is 1, while the RCM mechanism in Fig. 11 has a reduction of the rotation angle from input to output link. The fact that the novel RCM mechanism design strategy has control over the Φ_{out}/Φ_{in} ratio gives an additional degree of flexibility during mechanism design.

A last use for the novel RCM mechanism design strategy is that every newly created RCM mechanism can be converted back into a virtual center (VC) mechanism. For the mechanism shown in Fig. 10 this can be done by removing link CD and link GH, for the mechanism of Fig. 11, link CD and link D'H have to be removed. These VC mechanisms can then be used to design alternative RCM mechanisms using existing design strategies that are based on VC mechanisms [23].

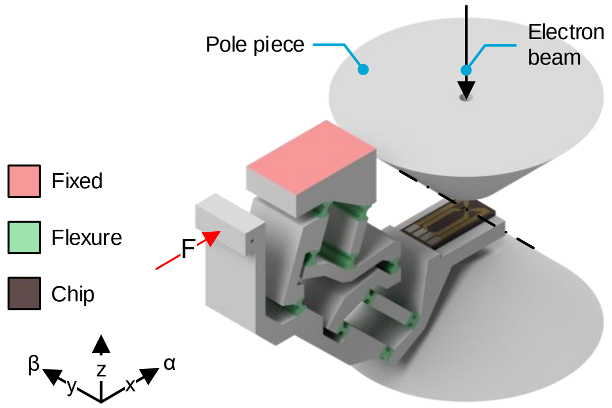


Fig. 12. An illustration of how a compliant RCM mechanism created with the novel RCM mechanism design strategy can be used in the sample holder. The mechanism is fixed at the top and actuated from the left.

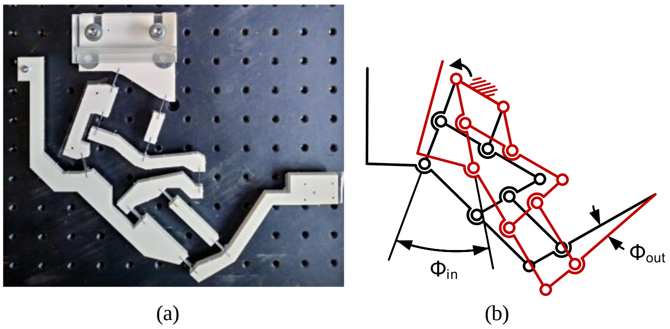


Fig. 13. (a) The large-scale proof-of-principle mechanism used to characterize the novel RCM mechanism design strategy and to validate the COMSOL results. Dimensions (w x h); 245 mm x 197 mm. (b) The rigid-body equivalent of the proof-of-principle mechanism in two different configurations. The black initial configuration is equal to the undeformed proof-of-principle mechanism as shown in (a). The red configuration is the rotated configuration when the input link is rotated counter-clockwise with input link rotation Φ_{in} and output link rotation Φ_{out}.

V. Compliant Proof-of-Principle Mechanism

If a RCM mechanism created with the novel RCM mechanism design strategy should be used in the sample holder (Fig. 12), it has to be compliant because of the manufacturing and assembly problems that currently exist with the very small revolute joints in the DENSsolutions double-tilt sample holder. A compliant RCM mechanism will bring benefits like ease of manufacturing, zero play and no need for lubrication but it will also introduce negative effects that influence the accuracy of the mechanism. A lower accuracy will result in higher sample drift of the sample holder, this can cause two problems for the user of the microscope; the image will defocus, or the area of interest might move outside of the viewing area of the microscope. To investigate these effects and to characterize and validate a RCM mechanism created with the novel RCM mechanism design strategy, a large-scale proof-of-principle mechanism is manufactured from acrylic links and spring steel flexures (Fig. 13).

A. Influences on Accuracy

Three effects that are inherent to compliant mechanism influence the accuracy of the RCM of the mechanism. First of all, a compliant hinge does not behave as a perfect hinge. The instantaneous center of rotation (ICR) of a flexure is not stationary; it will shift when the flexure is deflected [25]. This influences the drift of the RCM but not the offset because for

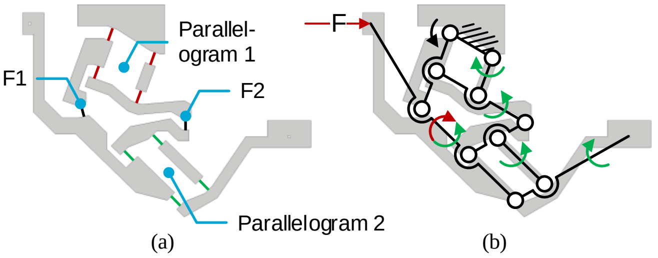


Fig. 14. (a) The orientations of the flexure hinges in the proof-of-principle mechanism. The four red flexures of parallelogram 1 and four green flexures of parallelogram 2 are oriented in the same direction to keep the parallelogram shape of both parallelograms intact when the ICR of the flexure hinges will shift. The two black flexures F1 and F2 are oriented in such a way that the scaled down four-bar shape has minimal change during actuation. (b) The parasitic rotations (green arrows) induced in the links due to the stiffness in the hinges when the mechanism is actuated in the direction of the actuation force. Red arrows show the actuation force and its resulting moment to compensate for these parasitic moments/rotations.

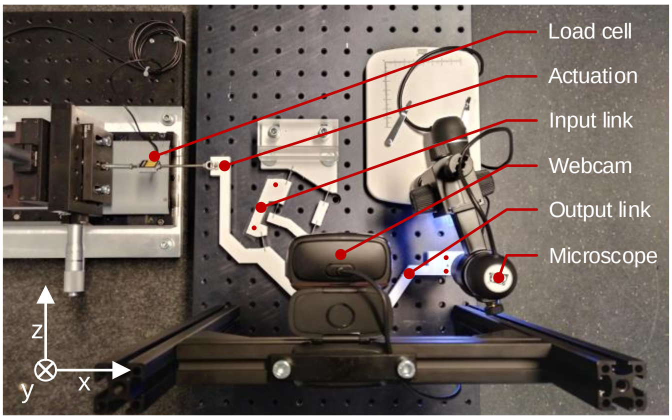


Fig. 15. An overview of the measurement setup with its different components. The proof-of-principle mechanism is actuated on the left side using a micro positioning stage. A webcam and a microscope are used to track dots on the mechanism and obtain the required data.

very small rotation angles a flexure can be considered as a perfect hinge without ICR shift. The orientations of the flexure hinges in the proof-of-principle mechanism (Fig. 14a) are defined in such a way that the amount of RCM drift because of hinge shift is minimized. For parallelogram 1 and 2, the orientations of the four flexure hinges are equal to keep the parallelogram shape intact even when hinge shift is introduced. The orientations of the two remaining hinges F1 and F2 of the scaled down four-bar mechanism shape are chosen in such a way that the shape of the four-bar mechanism undergoes minimal change during mechanism actuation. The proof-of-principle mechanism with the hinges in this configuration is analyzed using its kinematic- and hinge shift model and the optimal hinge shift orientations were verified. The rigid-body kinematics and the hinge shift model are discussed in detail in appendix C.

The second effect that influences the accuracy of the mechanism is the non-zero stiffness in the flexure hinges. When a hinge is deflected it will induce moments on the neighboring links because it has a certain bending stiffness. These links will then undergo parasitic rotation because the stiffness of the hinges in transverse direction is not high (Fig. 14b). Parasitic rotation of the links influences the offset and drift of the RCM.

Next to the stiffness in the hinges, the actuation force and location will also induce moments and parasitic rotations in the mechanism. But because the location of the actuation can be controlled the induced moments can be put to good use. If the mechanism is actuated at a location such that the resulting moment counteracts the moments induced by the stiffness in the hinges, the amount of parasitic link rotation can be reduced. This also explains why the mechanism is actuated on the extended bottom link.

B. *Testing Methodology*

The proof-of-principle mechanism will be tested on four aspects:

- Force/displacement
- Rotation ratio Φ_{out}/Φ_{in}
- RCM offset and drift
- Sample drift

The experimental results will be compared to results obtained via analytical and numerical methods. An overview of the measurement setup is shown in Fig. 15, it consists of the proof-of-principle mechanism that is actuated on the left side using a micro positioning stage connected via a load cell, there is no load on the output link. A webcam tracks two dots on the input link and a microscope tracks 4 dots on the output link. The actuation force will be measured by the load cell and the displacement is obtained through the position of the stage. For the rotation ratio, two tracked dots on the input link and two tracked dots on the output link are used to calculate the input and output link rotation and their ratio. For the RCM offset and drift and the sample drift four dots are tracked on the output link, using the coordinates of the dots the RCM and sample location can be calculated.

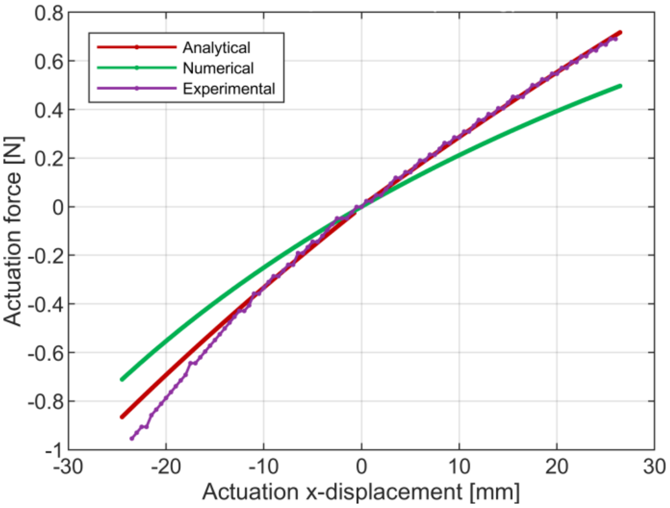


Fig. 16. Three force/displacement curves obtained by analytical, numerical or experimental analysis. The displacement range corresponds to a rotation of the input link of ±13°.

C. *Results*

In Fig. 16 three force/displacement curves are shown (for ±13° of input link rotation) which are obtained via analytical, numerical or experimental methods. The analytical curve is created using the pseudo-rigid-body model of the compliant proof-of-principle mechanism (appendix C). For the numerical results COMSOL Multiphysics software is used.

What first can be noticed is that the force/displacement curves are non-linear while the flexures in the proof-of-principle mechanism behave linear. However, the individual hinge rotations are not equal and are non-linear with respect to

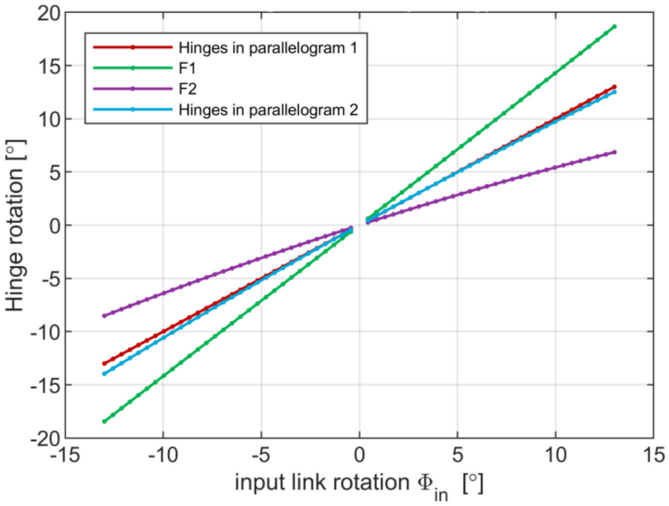


Fig. 17. The individual hinge rotations when the input link of the proof-of-principle mechanism is rotated over a range of ±13°. For identification of the hinges see Fig. 14a.

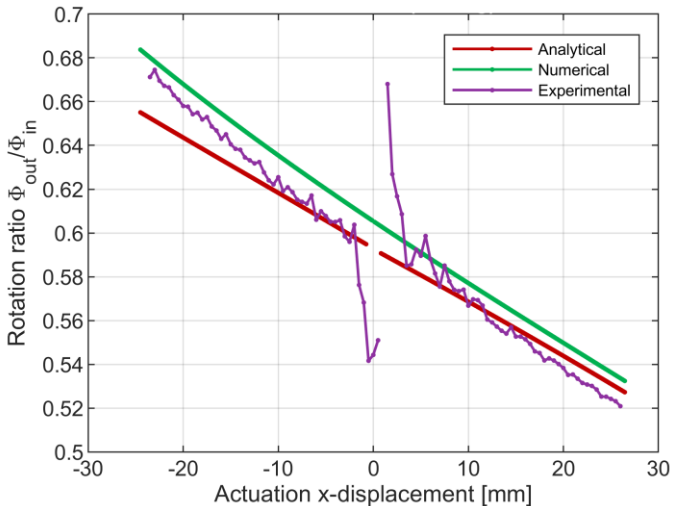


Fig. 18. The rotation ratio Φ_{out}/Φ_{in} as a function of the actuation x displacement. Obtained by analytical, numerical or experimental analysis. The displacement range corresponds to a rotation of the input link of ±13°.

the input link rotation (Fig. 17), therefore, the sum of all flexures results in non-linear force/displacement behavior.

The experimental results in Fig. 16 are similar to the analytical results, but they both show higher actuation forces than the numerical results. The higher actuation forces of the analytical model in comparison with the numerical model can be explained by the infinite transverse stiffness of the hinges in the analytical model. In reality, the transverse stiffness in the hinges is relatively low, so when moments are induced on the links, the mechanism can move to a lower energy state by moving the ends of the links in the transverse directions of the neighboring hinges. This means less energy is stored in the mechanism and thus lower actuation forces are required.

The difference in actuation force between the numerical and experimental results can come from a discrepancy between the Young’s modulus used in the numerical model and the actual Young’s modulus of the hinge material. Another cause can be glue residue on the hinges. The flexure hinges are glued into the acrylic links and glue residue at this connection can give higher stiffness to the flexures or reduce the length of the working part of the flexure, which increases the hinge stiffness and results in higher actuation forces.

Fig. 18 shows the rotation ratio Φ_{out}/Φ_{in} as a function of the actuation x-displacement (for ±13° of input link rotation). The difference between the analytical and numerical results is less than 5%, while the experimental results match even better with the analytical and numerical results except for the large errors around zero actuation x-displacement. Those errors can be explained by the bigger influence of the measurement error when rotation angles are small. The proof-of-principle mechanism was designed to have a rotation ratio of about 0.6, just like a standard four bar mechanism this ratio is dependent on its current state, but in this case the sensitivity of the rotation ratio to the actuation is relatively low.

The results of the last two measurements, the locations of the RCM and the sample, are shown in Fig. 19. The negative effects of a compliant mechanism are shown clearly in the RCM point measurements. In the ideal case of a rigid body mechanism that has no play in its hinges, the RCM is stationary and has no offset from the theoretically expected RCM location. For the compliant version we see that at zero degrees of rotation the RCM lies at an offset of the theoretically expected RCM location. Next to this, when the mechanism is actuated, the RCM will drift because of induced moments on the links and hinge shift.

Looking at Fig. 19 it can be seen that the numerical and experimental sample drift show similar behavior but different amounts of drift for positive and negative rotation of the mechanism. The sample drift is dependent on the location of the RCM and thus the variation in sample drift can be explained by the difference in shift between the numerical and experimental RCM curve. The experimental RCM curve is shifted to the left with respect to the numerical RCM curve and is thus more centered around the sample location. This will make the amount of drift of the sample during positive and negative rotation of the mechanism more equal, resulting in a reduction of the drift during positive rotation and an increase in drift during negative

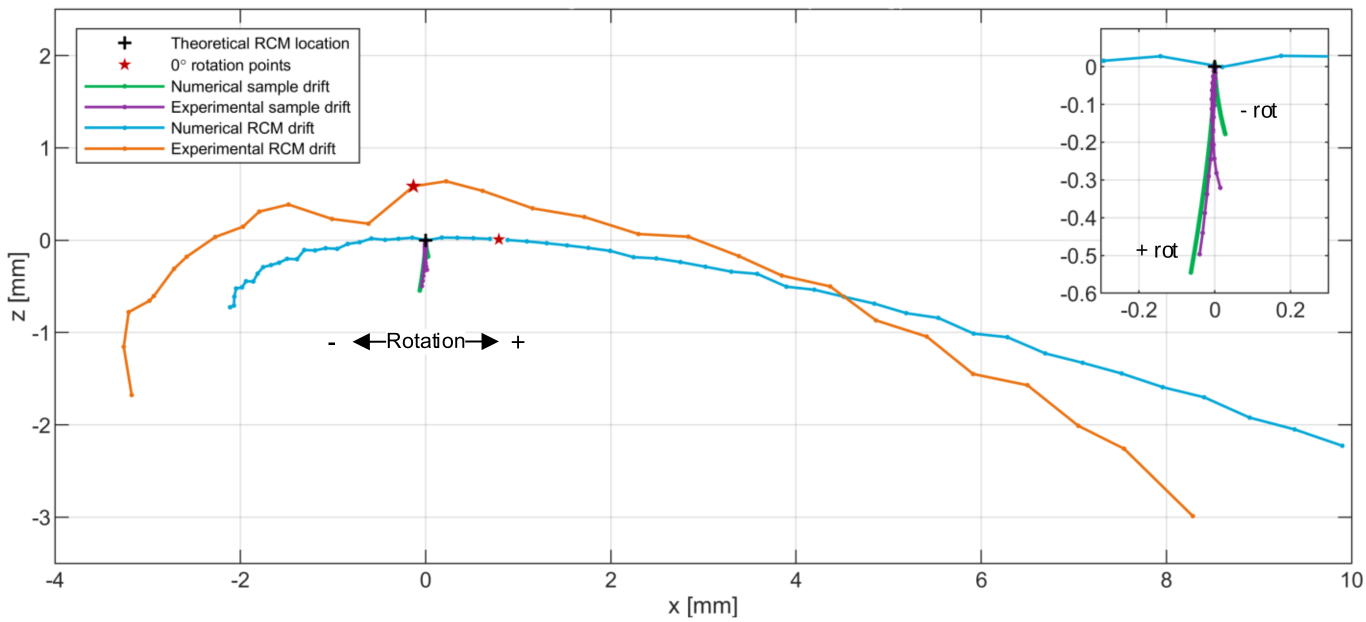


Fig. 19. The offset and drift of the RCM and the sample, for an input link rotation range of ±13°. The sample drift as a result of positive and negative rotation are indicated. For the RCM drift curves, the part to the left of the 0° rotation point is a result of negative rotation and to the right is a result of positive rotation.

rotation.

The shift of the experimental RCM curve can be attributed to a variation in hinge length due to manufacturing errors. The experimental RCM drift curve also has a higher curvature than its numerical counterpart. During experimentation a high sensitivity of the RCM location to actuation forces in z-direction (Fig. 15) was noticed. This can cause the experimental RCM drift curve to drop down more when the minimum and maximum rotation is reached, resulting in a higher curvature.

Concluding, the sample drift is about 0.55 mm. Because the RCM and sample drift scale 1:1 with the size of the mechanism (appendix C), if the mechanism would be scaled down 10 times to get to the size that would match the sample holder, the sample drift for that mechanism will be around 55 μm, which is just above the accuracy requirement of the sample holder of ±50 μm. Of course, scaling down the proof-of-principle mechanism will bring additional challenges, but it can be concluded that this RCM mechanism is a contender (although with some optimization) to be used in the sample holder when it comes to accuracy.

VI. Conclusion

By analyzing the EDS influences and the tilting mechanism of double-tilt sample holders, the need for a RCM mechanism as a tilting mechanism was identified. Based on the RCM mechanism classification and various RCM mechanism design strategies two concept groups were developed; RCM mechanisms that support the MEMS chip from the side and RCM mechanisms that support the MEMS chip from the back. Although these concepts already show a more than sevenfold improvement in EDS performance, for an optimal design a novel RCM mechanism design strategy has been developed that gives more design flexibility in terms of the use of space, design for stiffness and a variable input/output link rotation ratio in comparison with existing strategies. Using this design strategy, a compliant proof-of-principle mechanism was created which is used to characterize the RCM mechanism in terms of accuracy, force/displacement behavior and the input/output link rotation ratio. The results show potential for further investigation of a RCM mechanism based on the novel RCM mechanism design strategy as a tilting mechanism for double-tilt in-situ TEM sample holders.

VII. Recommendations

If a version of the proof-of-principle mechanism would be used for the double-tilt sample holder it needs to be scaled down. To still meet the tilting range requirement, the flexures need to be thinner or longer. Making thinner (<150 μm) flexures is challenging and implementing longer flexures will also be a challenge. Other linear elastic materials can be used, like titanium or beryllium copper, but a better solution to this problem might be using Nitinol [26]–[29]. Nitinol is a shape memory alloy which behaves superelastic above a material specific temperature (the Austenite finish temperature) and when stresses are induced. Therefore, it is an ideal material for compliant mechanisms that have lumped compliance, like the RCM mechanisms in this paper. Nitinol shows nonlinear elastic stress-strain behavior and the influence of this behavior on the accuracy of a compliant RCM mechanism is unknown. Furthermore, the mechanical properties of Nitinol are sensitive to changes in temperature which also influences the performance. Finally, the thermal and mechanical of stability of such mechanisms can be investigated.

VIII. References

[1] P. J. Ferreira, K. Mitsuishi, and E. A. Stach, “In Situ Transmission Electron Microscopy,” *MRS Bull.*, vol. 33, no. 02, pp. 83–90, Feb. 2008.

[2] M. L. Taheri *et al.*, “Current status and future directions for in situ transmission electron microscopy,” *Ultramicroscopy*, vol. 170, pp. 86–95, Nov. 2016.

[3] R. A. Bernal, R. Ramachandramoorthy, and H. D. Espinosa, “Double-tilt in situ TEM holder with multiple electrical contacts and its application in MEMS-based mechanical testing of nanomaterials,” *Ultramicroscopy*, vol. 156, pp. 23–28, 2015.

[4] D. B. Williams and C. B. Carter, *Transmission Electron Microscopy: A Textbook for Materials Science*, vol. 1–4. Boston, MA: Springer US, 2009.

[5] A. Mikajiri, “Specimen tilting device for an electron optical device,” US3778621 A, 1973.

[6] J. S. Jones and P. R. Swann, “Specimen heating holder for electron microscopes,” US4996433 A, 1991.

[7] G. Unter, M. Obus, R. C. Doole, and B. J. Inkson, “Spectroscopic electron tomography,” *Ultramicroscopy*, vol. 965980, no. 61, pp. 433–45146, 2003.

[8] M. A. Verheijen *et al.*, “Transmission electron microscopy specimen holder for simultaneous in situ heating and electrical resistance measurements,” *Rev. Sci. Instrum.*, vol. 75, no. 2, pp. 426–429, 2004.

[9] H. Miyazaki and S. Hata, “Sample holder,” US 7705324 B2, 2010.

[10] X. Han *et al.*, “Double tilt transmission electron microscope sample holder for in-situ measurement of microstructures,” US8569714 B2, 2011.

[11] T. Sato, E. Tochigi, T. Mizoguchi, Y. Ikuhara, and H. Fujita, “An experimental system combined with a micromachine and double-tilt TEM holder,” *MEE*, vol. 164, pp. 43–47, 2016.

[12] R. M. Langford and A. K. Petford-Long, “Preparation of transmission electron microscopy cross-section specimens using focused ion beam milling,” *J. Vac. Sci. Technol. A Vacuum, Surfaces, Film.*, vol. 19, no. 5, pp. 2186–2193, Sep. 2001.

[13] W. Zhou and Z. L. Wang, Eds., *Scanning Microscopy for Nanotechnology*. New York, NY: Springer New York, 2007.

[14] T. Slater, Y. Chen, G. Auton, N. Zaluzec, and S. Haigh, “X-Ray Absorption Correction for Quantitative Scanning Transmission Electron Microscopic Energy-Dispersive X-Ray Spectroscopy of Spherical Nanoparticles,” *Microsc. Microanal.*, vol. 22, pp. 440–447, 2016.

[15] C. E. Lyman, J. I. Goldstein, D. B. Williams, D. W. Ackland, S. Von Harrach, and A. W. NICHOLLSt, “High-performance X-ray detection in a new analytical electron microscope,” *J. Microsc.*, vol. 176, no. 2, pp. 85–98, 1994.

[16] T. J. A. Slater, P. H. C. Camargo, M. G. Burke, N. J. Zaluzec, and S. J. Haigh, “Understanding the limitations of the Super-X energy dispersive x-ray spectrometer as a function of specimen tilt angle for

tomographic data acquisition in the S/TEM," *J. Phys. Conf. Ser.*, vol. 522, no. 1, p. 012025, Jun. 2014.

[17] J. F. Jensen, "Remote center positioner with channel shaped linkage element," US 5800423, 1998.

[18] G. Zong, X. Pei, J. Yu, and S. Bi, "Classification and type synthesis of 1-DOF remote center of motion mechanisms," *Mech. Mach. Theory*, vol. 43, pp. 1585–1595, 2008.

[19] J. L. H. M. Janssen, D. Farhadi Machekposhti, D. Morsink, and N. Tolou, "Classification of Remote-Center-of-Motion Mechanisms," 2018.

[20] J. Li, G. Zhang, A. Müller, and S. Wang, "A Family of Remote Center of Motion Mechanisms Based on Intersecting Motion Planes," *J. Mech. Des.*, vol. 135, no. 9, p. 091009, Jul. 2013.

[21] Y. He, P. Zhang, H. Jin, Y. Hu, and J. Zhang, "Type Synthesis for Remote Center of Motion Mechanisms Based on Coupled Motion of Two Degrees-of-Freedom," *J. Mech. Des.*, vol. 138, no. 12, p. 122301, Sep. 2016.

[22] J. Li, G. Zhang, Y. Xing, H. Liu, and S. Wang, "A Class of 2-Degree-of-Freedom Planar Remote Center-of-Motion Mechanisms Based on Virtual Parallelograms," *J. Mech. Robot.*, vol. 6, no. 3, p. 031014, Jun. 2014.

[23] H. Long, Y. Yang, X. Jingjing, and S. Peng, "Type Synthesis of 1R1T Remote Center of Motion Mechanisms Based on Pantograph Mechanisms," *J. Mech. Des.*, vol. 138, no. 1, p. 014501, Nov. 2015.

[24] S. Liu, C. Chen, B. Chen, and L. Harewood, "Novel Linkage with Remote Centre of Motion," in *Proceeding of the 14th IFToMM World Congress*, 2015, pp. 338–343.

[25] M. Verotti, "Analysis of the center of rotation in primitive flexures: Uniform cantilever beams with constant curvature," *MAMT*, vol. 97, pp. 29–50, 2016.

[26] E. Henderson, D. H. Nash, and W. M. Dempster, "On the experimental testing of fine Nitinol wires for medical devices," *J. Mech. Behav. Biomed. Mater.*, vol. 4, no. 2, pp. 261–268, 2011.

[27] J. Eaton-Evans, J. M. Dulieu-Barton, E. G. Little, and I. A. Brown, "Observations during mechanical testing of Nitinol," *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.*, vol. 222, no. 2, pp. 97–105, 2008.

[28] Y. Liu, "On the Detwinning Mechanism in Shape Memory Alloys," in *IUTAM Symposium on Mechanics of Martensitic Phase Transformation in Solids*, 2002, pp. 37–44.

[29] J. Liu, "Compliant Mechanisms using Superelastic NiTiNOL," Pennsylvania State University, 2012.

4 Conclusion

The current design of the DENSsolutions sample holder is analyzed. The best optimization direction for the sample holder design was identified by mapping the various influences on the amount and quality of the EDS signal. These EDS influences were filtered to only include the influences that are critical for the design of the sample holder. As a result of the analysis of the DENSsolutions double-tilt sample holder the need for a RCM mechanism was identified.

Therefore, the literature on RCM mechanisms was reviewed, and the RCM mechanism classification was updated by removing the isocenters and passive RCM mechanisms class and adding the instantaneous center of rotation and intersecting motion planes class. Various design strategies were described, and a knowledge gap was indicated in the shape of a RCM mechanism design strategy that creates a RCM mechanism with a tunable output/input link rotation ratio. A novel RCM design strategy was created as a result. A schematic overview of the RCM mechanism classes and the accompanying design strategies is introduced as a tool to be used during RCM mechanism design.

Using this overview various concepts were developed and evaluated. Those concepts can be divided into two groups; concepts that support the MEMS chip from the side and concepts that support the MEMS chip from the back. Although the concept that uses two flexures to replace the revolute joints of the current sample holder design meets the requirements, the concepts that support the MEMS chip from the back give more flexibility regarding MEMS chip design and a higher α-tilt range.

Using the novel RCM mechanism design strategy, a concept was developed that supports the MEMS chip from the back. The negative effects of a compliant RCM mechanism are identified, and the accuracy of the design of the concept mechanism is optimized using its kinematic model with implemented hinge shift model and pseudo-rigid-body model (PRBM). A large-scale proof-of-principle mechanism is manufactured which is tested on its force/displacement behavior, output/input link rotation ratio and RCM and sample location offset and drift. These experimental results are compared to analytical and numerical results obtained with the kinematic/PRBM model and COMSOL respectively. The accuracy of the proof-of-principle mechanism is obtained and when scaled down to the size of a sample holder it can be concluded that it meets the accuracy requirement and shows promise to be used as the tilting mechanism in a double-tilt sample holder.

5 Recommendations

There are a few things that I did not get to do during this thesis and it would be nice to see these back in future work. First of all, the proof-of-principle mechanism has to be scaled down to make sure it fits in the sample holder. To still meet the tilt-requirement, the flexures have to be made thinner, longer or a combination of the two. Both options will be difficult; the flexures in the proof-of-principle mechanism are already very thin (150 µm) and fitting in longer flexures will also pose a challenge. Using different linear elastic materials like titanium or beryllium copper are options that might make this process easier.

Another option is using Nitinol, a non-linear elastic material that behaves superelastic when it is stressed. Nitinol can reduce the required size of the hinges, so a variant of the proof-of-principle mechanism can fit in the sample holder. However, Nitinol is a difficult material to model and manufacture. The non-linear elastic behavior of Nitinol will affect the accuracy of the mechanism in a different way than is the case when linear elastic materials are used.

Whether the result is a mechanism that uses a linear elastic or non-linear elastic material, the stability of the mechanism should be evaluated. This stability includes the mechanical stability of the mechanism but also the effect of temperature fluctuations.

In the case of a Nitinol concept, experimental testing might give valuable information on the validation of Nitinol material models and will give accurate results, because Nitinol material properties tend to have high variations.

Finally, it is known that the proof-of-principle mechanism is sensitive to forces in other directions than the actuation force. The design of an actuation system that can actuate the mechanism in the proper way, and that also fits inside the limited space of the sample holder will bring challenges.

A EDS Influences

To achieve the highest accuracy on the identification of the elements in a sample, the amount and quality of the EDS signal has to be optimized. In this chapter the different influences on the EDS signal will be discussed, an overview of these influences is given in Fig. 3. The 6 main influences discussed in the next paragraphs are:

- Detector type/window
- Detector errors
- X-ray absorption
- Desired collection angle
- Undesired collection angle
- Background noise

In this case all influences up to the point that the signal exits the detector and enters the electronics and data processing part is taken into consideration.

A.1. Detector Type/Window

A Si detector, which is the most commonly used detector, can convert the energy of a characteristic X-ray into a measurable quantity. In Fig. 2 a schematic overview of a Si detector is shown, its working is as follows: when an X-ray enters the detector, it will interact with the Si, transferring its energy and creating electron-hole pairs. For every 3.8 eV of energy an electron-hole pair will be created, thus the number of electrons separated from their “hole” is proportional to the energy of the X-ray that entered the detector. Because a characteristic X-ray typically has an energy well above 1 keV a high enough resolution can be achieved with this method to distinguish most elements. By applying a negative bias across the Si the electrons are forced towards the back of the detector. At the rear contact an electron pulse can then be measured. To identify the pulse as an element, it will be converted into a voltage which will be further amplified so it can be matched to a specific X-ray energy.

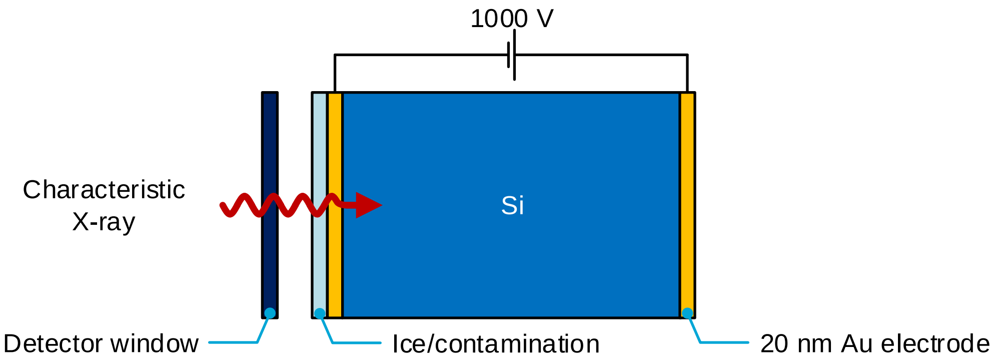


Fig. 2. A schematic overview of a Si detector. Adapted from [2].

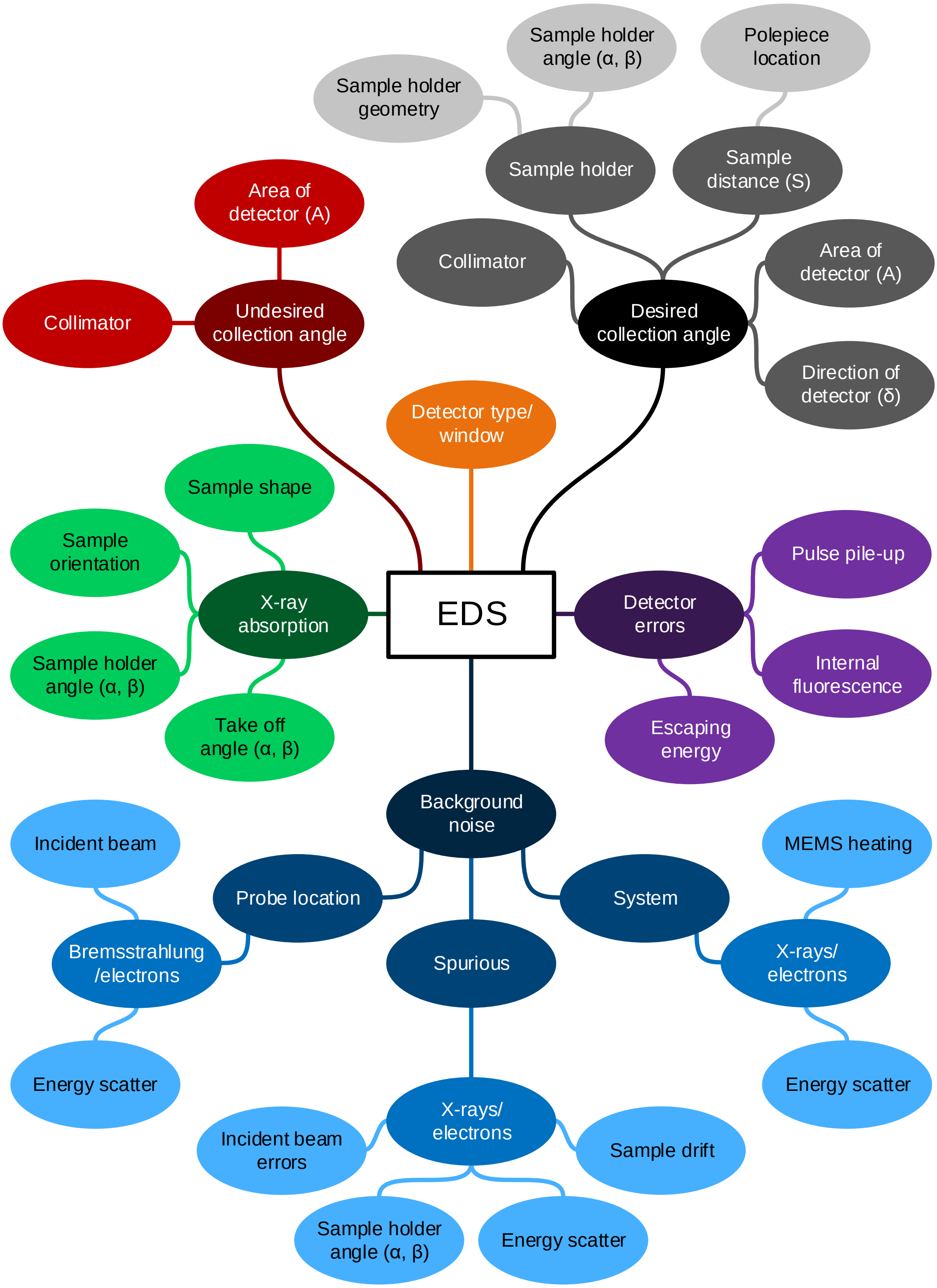


Fig. 3. The different influences on the amount and quality of the EDS signal.

The detector processes the X-rays one by one, keeping track of the identified elements for every X-ray. After a certain time, a spectrum can be plotted that shows how many X-rays are received from every element. This will give an overview of the different materials and how common they are in the sample. An example is shown in Fig. 4, from the image alone the different materials cannot be identified but with the help of EDS and its obtained spectrum it becomes clear that the image shows aluminum that is evaporated onto a silicon substrate. [2], [3]

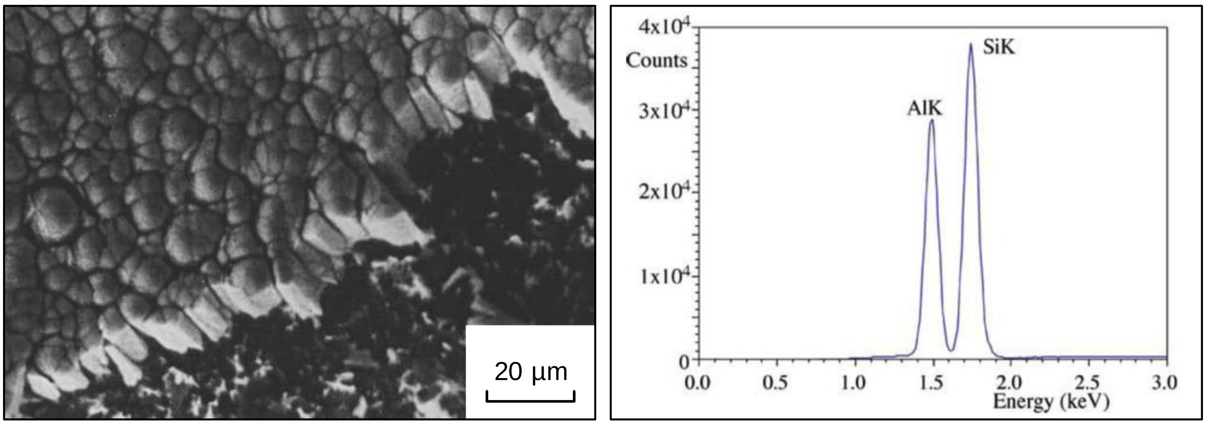


Fig. 4. (left) An image of Al evaporated onto a Si substrate and (right) its spectrum created with EDS. [2]

The type of detector that is used will influence the number of X-rays it can process per second, also called the count rate. Alternatives to the Si detector are the Ge detector or a Si-drift-detector. Next to the detector itself, the type of window of the detector also influences the EDS signal. A Be-, ultra-thin-, or atmospheric-thin window can be used, or even no window, which will result in contamination problems. Because these factors don't influence the amount or quality of the EDS signal that is received from the sample they will not be discussed further here. For more information see chapter 32.4 in [2].

A.2. Detector Errors

The EDS detector itself has some effects that create errors in the spectrum obtained from a sample. First there is pulse-pile up; when the electronics of the detector cannot keep up with the count rate there is a chance that multiple electron pulses will be summed into a single voltage signal. This will often result in a peak in the spectrum at twice the characteristic energy of the most common element.

Occasionally some of the incoming energy will not be used to form electron-hole pairs but will fluoresce a Si X-ray (energy 1.74 keV). Two things can happen with this X-ray; its energy is converted to electron-hole pairs in the detector (internal fluorescence) or the Si X-ray will leave the detector (escaping energy). In the case of absorption, the detector can't distinguish between the internal fluoresced Si X-ray and an actual Si X-ray that might have come from the sample, so this will result in a Si peak in the spectrum. When the Si X-ray leaves the detector, a part of the energy of the original characteristic X-ray will be lost. This will result in a measured X-ray energy that is 1.74 keV lower than the actual energy of the characteristic X-ray. [2]

A.3. X-Ray Absorption

When the incident electron beam produces characteristic X-rays in the sample there is a chance that they are absorbed by the sample before they are emitted from the surface towards the detector. The further the X-ray has to travel through the sample (the absorption distance), the higher the chance that it will be absorbed. The absorption distance can be influenced in two ways (Fig. 5); either by the shape of the sample or by a variation of the angle between the sample and the detector. Variation of the shape and thus a variation in absorption distance is illustrated in Fig. 5a. Variation of the angle between the sample and the detector can be influenced in multiple ways; the sample is rotated with respect to the detector, the sample holder is rotated with respect to the detector or the detector is rotated with respect to the sample (variation in take-off angle, Fig. 5b). All three rotations are equivalent in how they result in different amounts of absorption. In most electron microscopes the take-off angle is generally pretty low (≈18°) because of pole piece restrictions, which increases the X-ray absorption. [2], [4]

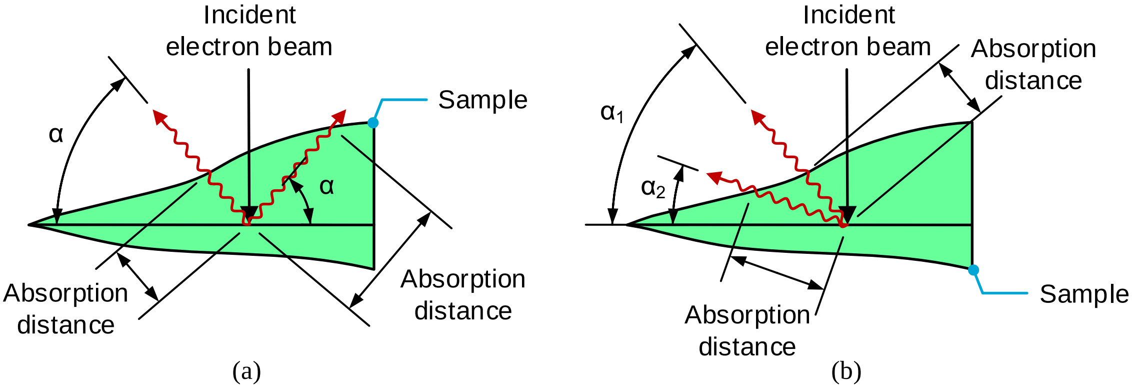


Fig. 5. The influence of (a) sample shape and (b) sample orientation, sample holder angle and take-off angle on the amount of absorption in the sample. Adapted from [2].

A.4. Desired Collection Angle

It is known that when the incident electron beam hits the sample, electrons and photons are emitted. These particles will be scattered throughout the TEM and will hit the different components, e.g. the pole pieces, sample holder or other parts of your sample. The particles will interact with the atoms of these parts, creating additional signals. Basically, every part in a TEM that is near the sample will emit its own electrons and X-rays due to this energy scatter. Preferably, only the signals from the probe area on the sample should reach the detector, the background noise created by other parts in the TEM is undesirable. The amount of desired EDS signal that is received from the sample is defined by the desired X-ray collection angle Ω (Fig. 6, equation (A.1)). It is dependent on the surface area of the detector A, the distance from the sample to the detector S and δ, the angle between the normal to the detector surface and a line from the center of the detector to the sample. In many EDS systems, the detector is tilted towards the sample so $\delta = 0$. A decrease in S, and thus an increase in Ω, is often limited by the geometry of the upper pole piece while A normally lies between 10 to 50 mm^2. In contrast to the theoretical maximum desired collection angle, in practice, this angle is often reduced by the collimator or by the geometry and angle of the sample holder. These components can obstruct the EDS signal, shadowing the detector from the sample X-rays. [2], [5], [6]

$$\Omega = \frac{A \cos \delta}{S^2} \tag{A.1}$$

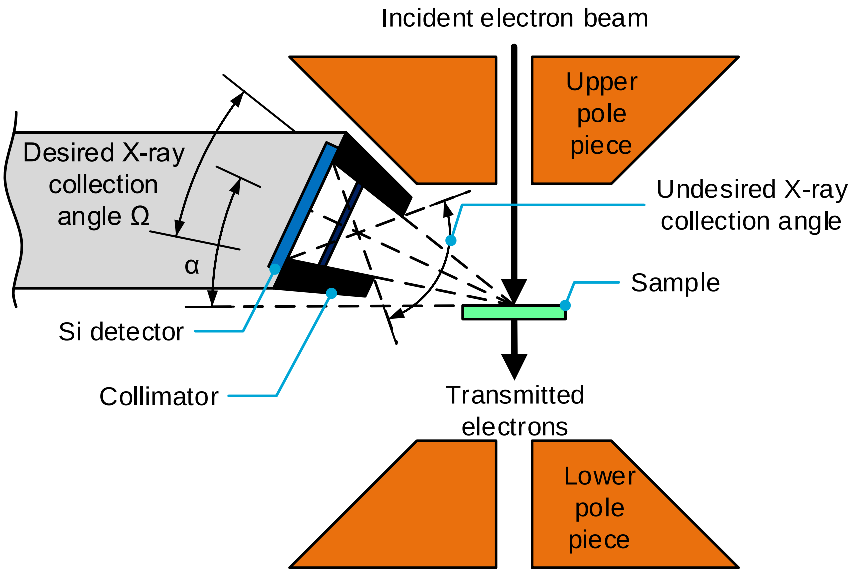


Fig. 6. A schematic drawing of the EDS detector in a TEM. Adapted from [2].

A.5. Undesired Collection Angle

Next to the desired signals there exists the background noise as discussed in the previous paragraph. The collimator of the detector is designed to block as much of the noise as possible by preferring signals from a general direction (Fig. 6). The maximum angle over which signals can be received is called the undesired collection angle. The smaller the undesired collection angle, the lower the amount of noise that can enter the detector. As shown in Fig. 3 the undesired collection angle is dependent on the surface area of the detector A and the geometry of the collimator. [2]

A.6. Background Noise

Background noise is indicated as an undesirable signal in the previous paragraphs but how does it influence the amount and quality of the EDS signal? First of all, three different locations can be specified as the source of the noise. One option is the probe location on the sample itself. Another type of noise comes from spurious X-rays/electrons, which means noise from the sample but away from the probe location. The last type of noise comes from the system, which consists of all parts in the TEM except for the sample.

From the probe location, the characteristic X-rays are desired but other signals are not, like the bremsstrahlung X-rays and emitted electrons. They are created by the incident electron beam and the energy scatter in the TEM and will create noise in the detector.

Spurious X-rays and electrons can be created in 4 different ways; by incident electron beam errors (electrons that don't hit the right spot on the sample), by increasing the sample holder angle and thus manipulating the electron beam spot on the sample, by emission due to energy scatter and because of sample drift.

Finally, the noise from the system. Emitted X-rays and electrons are created, again, by energy scatter. Another phenomenon is the emission of infrared light by the MEMS heating chip when in situ heating is used, this can saturate the detector. [2], [5]

B Concepts

B.1. Requirements

Next to the critical requirements mentioned in chapter 3 there exist additional requirements. The complete list of requirements is listed in Table 1. First of all, the sample holder should be able to fit in the TEM and more specifically, the tilting mechanism should fit in between the pole pieces (Fig. 7).

TABLE 1
REQUIREMENTS

Requirement	Value
Workspace	Ø 6.5 mm x 17 mm
β rotation range	$> \pm 10°$
β rotation resolution	$< 0.1°$
β rotation repeatability	$< 1°$
β rotation speed	< 1 min/20°
Sample drift	$< \pm 50$ µm
MEMS chip stability	< 1 nm/min
Vacuum compatibility	$< 10^{-6}$ Pa

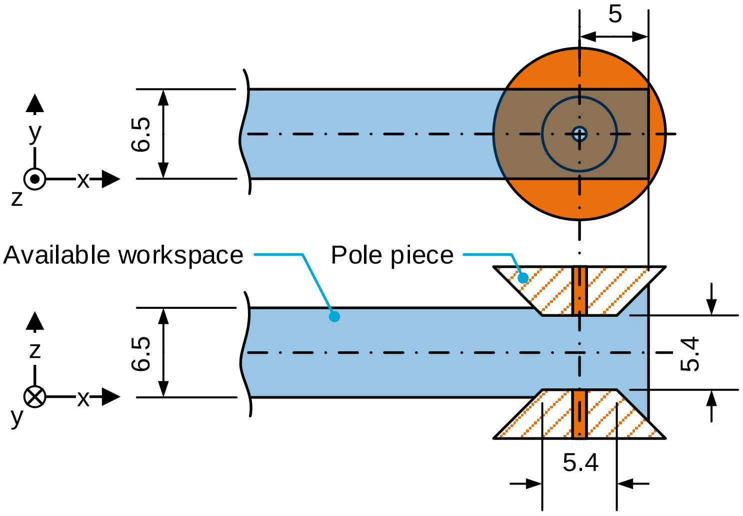


Fig. 7. The space available for the sample holder and tilting mechanism.

The β rotation range requirement is dependent on how the image in a TEM is optimized. According to Dr. Qiang Xu, the quality of the projected image depends on the angle between the crystal structure of the sample and the incident electron beam. In Fig. 8 two projections are shown at different sample angles. It can be seen that when the crystal structure of the sample is aligned with the direction of the incident electron beam, an image with higher contrast is created. Thus, for an optimum image, the alignment between the crystal structure of the sample and the incident electron beam is crucial. If the sample has a cube-shaped crystal structure, there is always one of the optimal crystal structure orientations within 45° of tilt. To also be safe for non-equal crystal structure spreading of the atoms in the sample, a β-tilt range of 50° is advised. However, this requirement is only necessary for the case that there is no control over the orientation of the crystal structure of the sample during sample preparation. If a user can orientate a sample during sample preparation in such a way that the crystal structure is somewhat aligned with the incident electron beam, the β-tilt range requirement can be lowered.

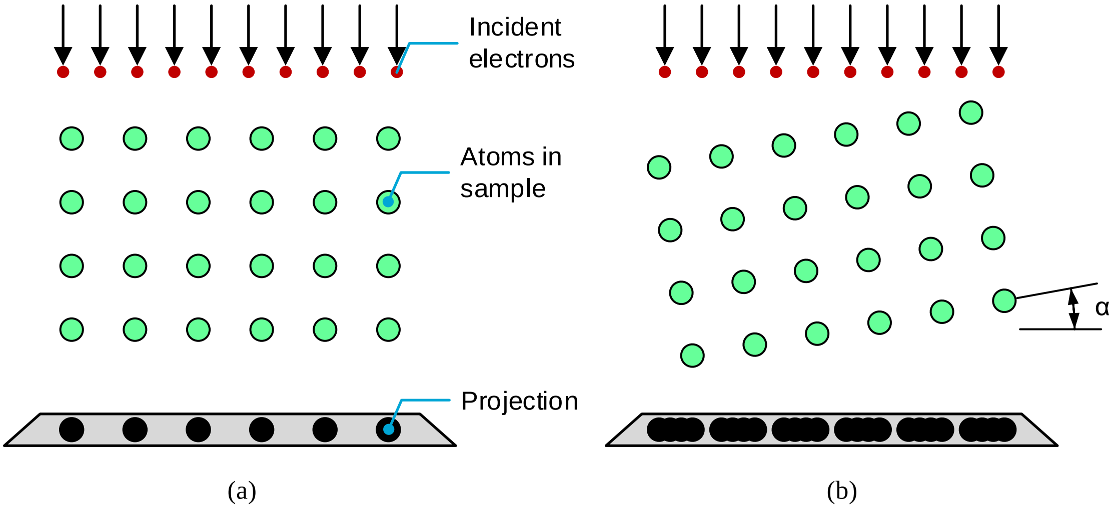


Fig. 8. The effect of rotating the sample with respect to the incident electron beam. (a) The crystal structure of the sample aligned with the incident electron beam. (b) The crystal structure of the sample misaligned with the incident electron beam.

One of the techniques that gives control over the sample orientation and has the highest β-tilt requirement is focused ion beam (FIB) sample preparation [7]. Using FIB sample preparation, the sample can be prepared with a crystal structure orientation error of ±1°. After the sample is pulled out of the bulk material it is positioned at an angle of 8 degrees with respect to the MEMS chip surface resulting in a total required β-tilt of ±9°. The placement of the FIB lamella is restricted to the FIB machine, so it is always positioned in the same way. This also means that only a one-sided β-tilt range is needed when FIB sample preparation is used.

For nanoparticles, nanowires, 2d samples and polycrystalline dimpled samples there are multiple grains that have various orientations. By searching for a grain that is close to the correct grain orientation, only small tilt is necessary (±5°). For monocrystalline dimpled samples the bulk material can mostly be cut at the right angles to get the dimpled sample in the right orientation (<±5°). Concluding, a minimum tilt range of +9/-5 is required. To be safe, also with different lamella orientations because of different FIB preparing methods, ±10° is taken as the main requirement.

The resolution, repeatability and speed requirements are not the most relevant requirements because they are not that strict. A resolution of 0.1° is necessary and this is not difficult to achieve with most mechanisms. Because the alignment of the sample with the electron beam is visually optimized there is practically no need for high repeatability, therefore 1° is taken as the requirement. The speed requirement is also dependent on user friendliness, it shouldn't take long to tilt the stage and interrupt the user's workflow. 1 minute/20° is taken.

The sample drift is defined by the distance the center point of the MEMS chip drifts during rotation. When high drift occurs, the sample can drift outside of the view of the TEM. A range of ±50 μm is acceptable and is on par with the performance of current sample holder designs.

Another type of drift can occur because of thermal or structural effects. Both effects are included in the stability of the MEMS chip. The value of 1 nm/min comes from the current DENSsolutions sample holder. This corresponds to a lowest eigenfrequency of the sample holder of 1650 Hz. This value is valid for the optimum case that the cradle is rigidly connected to the sample holder and there is no play in the hinges. Because the optical column, sample holder and image plane are rigidly connected, a high stiffness connection is required between the sample and the TEM column to make sure that, when vibrations occur, all crucial parts that are used to create the image will vibrate together (Fig. 9). If the new design has its eigenfrequency above 1650 Hz its stability should be sufficient.

Finally, the materials used should not affect the vacuum performance of the TEM. Materials with high outgassing values cannot be used. Also structures that can contain a lot of contamination or are difficult to get to the desired pressure should be avoided.

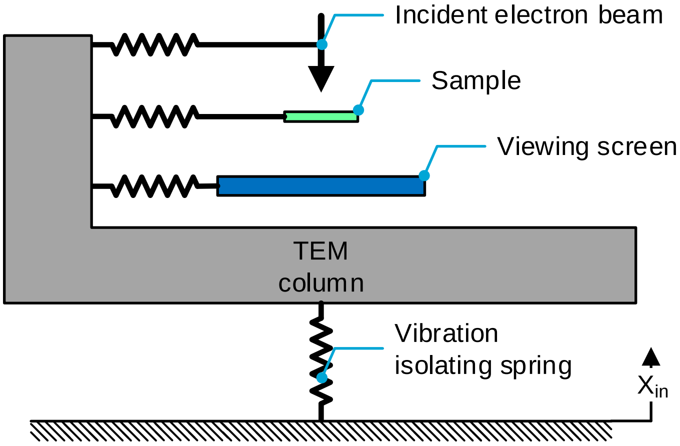


Fig. 9. An illustration of the how the different parts used to create the image are connected to each other.

B.2. Concept Evaluation

Various concepts are developed for the tilting mechanism in the sample holder (Fig. 10). They can be divided into two groups; concepts that support the MEMS chip from the side and concepts that support the MEMS chip from the back. The mechanisms in the first group are based on RCM mechanisms from the spherical linkages class while the mechanisms in the second group are based on RCM mechanisms from the instantaneous center of rotation (ICR) or parallelograms class. The concepts are evaluated on maximum stress, actuation force, actuation moment, sample drift, eigenfrequency and buckling force using COMSOL. The results are shown in Table 2 with the values that do not satisfy the requirements shown in red.

Most concepts are made out of titanium because of its good Young's modulus and high yield strength. Another material that is chosen is Nitinol; a shape memory alloy that shows superelastic behavior at a certain temperature range when it is stressed. It is used in concepts 8-10. Simulations have been done using a shape memory alloy (SMA) material model in COMSOL, but those results should be taken with a grain of salt because this material model is not validated and has a high chance of giving different results than a real world prototype. The benefits and challenges of Nitinol will be discussed in the next chapter.

In concepts 1-4 the traditional revolute joints in the DENSsolutions sample holder are replaced with compliant alternatives. In concept 1 this is done by using two flexures. The compliant elements get rid of play that normally exists in revolute joints and simplify the assembly process. The current design of the DENSsolutions sample holder shows problems because assembling the tiny hinges (Ø 0.7 mm) and removing the play by adding pre-tension is a process that is difficult to do well consistently. The drawback comes in the shape of hinge shift which introduces sample drift; the movement of the sample when the sample is rotated. But as can be seen in Table 2 this is still below the requirement of ±50 µm.

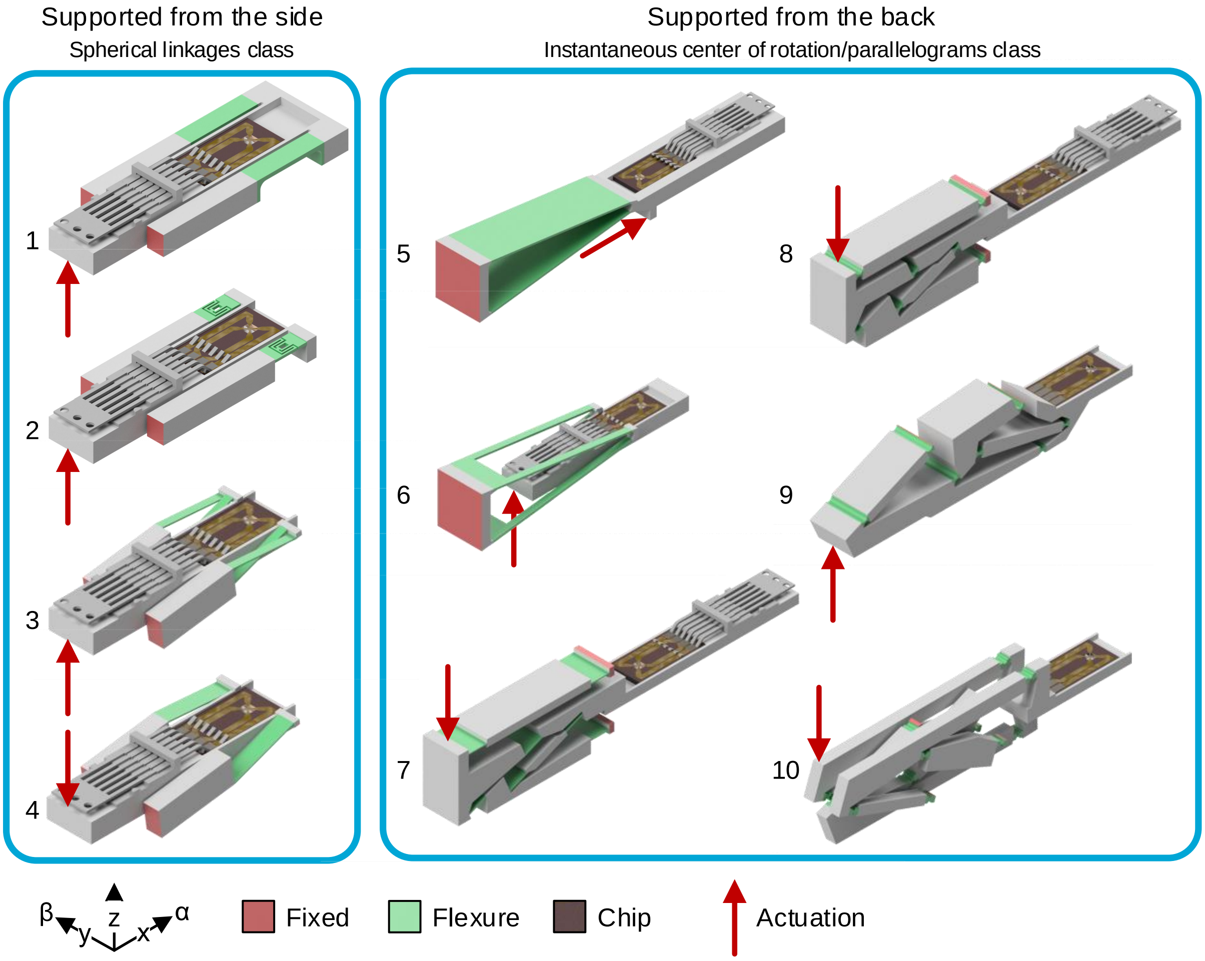


Fig. 10. The various concepts for the tilting mechanism in the DENSsolutions double-tilt sample holder that have been developed and evaluated.

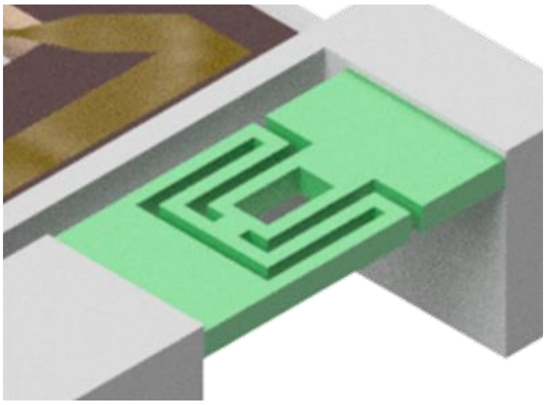


Fig. 11. The inverse lamina emergent torsional (I-LET) joint used in concept 2.

TABLE 2
CONCEPT EVALUATION

#	Name	Stress [MPa] @ 10°	Actuation force [N]	Actuation moment [Nmm]	Sample drift [μm]	Lowest eigenfrequency [Hz]	Buckling force [N]	Thickness of elastic element [mm]
1	Leaf spring hinge	772	1.76	24.64	25	2261	128	0.3
2	I-LEJ	715	0.03	0.42	17	1125	3.3	0.1
3	Cross hinge	800	0.14	1.96	1	2960	2.2	0.1
4	Cross hinge asymmetric	870	0.12	1.68	122	1290	3.6	0.1
5	ICR	815	25.20	50.18	25	1600	5.5	0.2
6	ICR double	850	1.21	16.94	13	600	1.5	0.2
7	Double parallelogram	850	2.01	38.99	29	965	4.4	0.1
8	Double parallelogram N 1	No failure criteria	5.80	109.33	98	958	63	0.15
9	Double parallelogram N 2	No failure criteria	17.21	20.35	90	1735	43.8	0.15
10	3 VC Nitinol	No failure criteria	7.46	10.82	128	638	14	0.1

In concept 2 a lamina emergent torsional (LET) joint (Fig. 11) [8], [9] is used which reduces the sample drift, but the joint will fail on buckling a lot earlier. The I-LET joint will also reduce the lowest eigenfrequency in comparison with concept 2.

Concepts 3 and 4 are based on cross hinges. Concept 3 uses two cross hinges while concept 4 uses a single cross hinge which is split between both sides of the MEMS chip. This type of hinge shows very low sample drift values. Nevertheless, this type of hinge is very sensitive to buckling because the flexures are exposed to axial compressive and tensile forces during rotation.

For the concepts that support the MEMS chip from the back, concepts 5 and 6 belong to the ICR class. This type of mechanism also depends for its rotation not only on bending but also on compression and tension. This, in combination with long thin structures results in high sensitivity to buckling. Concept 5 is in that regard just above the threshold of 5 N but concept 6 will fail because of buckling. Concept 5 satisfies all requirements, but it is difficult to actuate and needs a high actuation force of about 25 N. For a mechanism the size of the tip of your little finger, 25 N is quite a lot.

Concepts 7-10 are all based on the parallelograms RCM class. The kinematics of the mechanisms in concepts 7-9 are identical to the mechanism shown in Fig. 12 but in different configurations (Fig. 13). Concept 7 uses titanium flexures and is fixed on the two, outside links, close to the MEMS chip. This is not an ideal location to connect the mechanism to the sample holder, but it is dictated by the kinematic constraints of the mechanism. Nevertheless, this is not the only problem with the mechanism; the high number of hinges and how they are connected result in a mechanism with a low eigenfrequency.

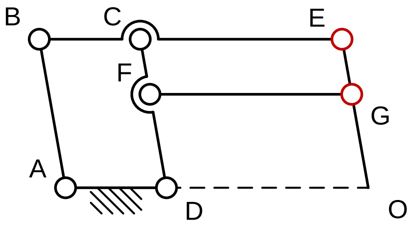


Fig. 12. The kinematics of the mechanisms used in concept 7-9.

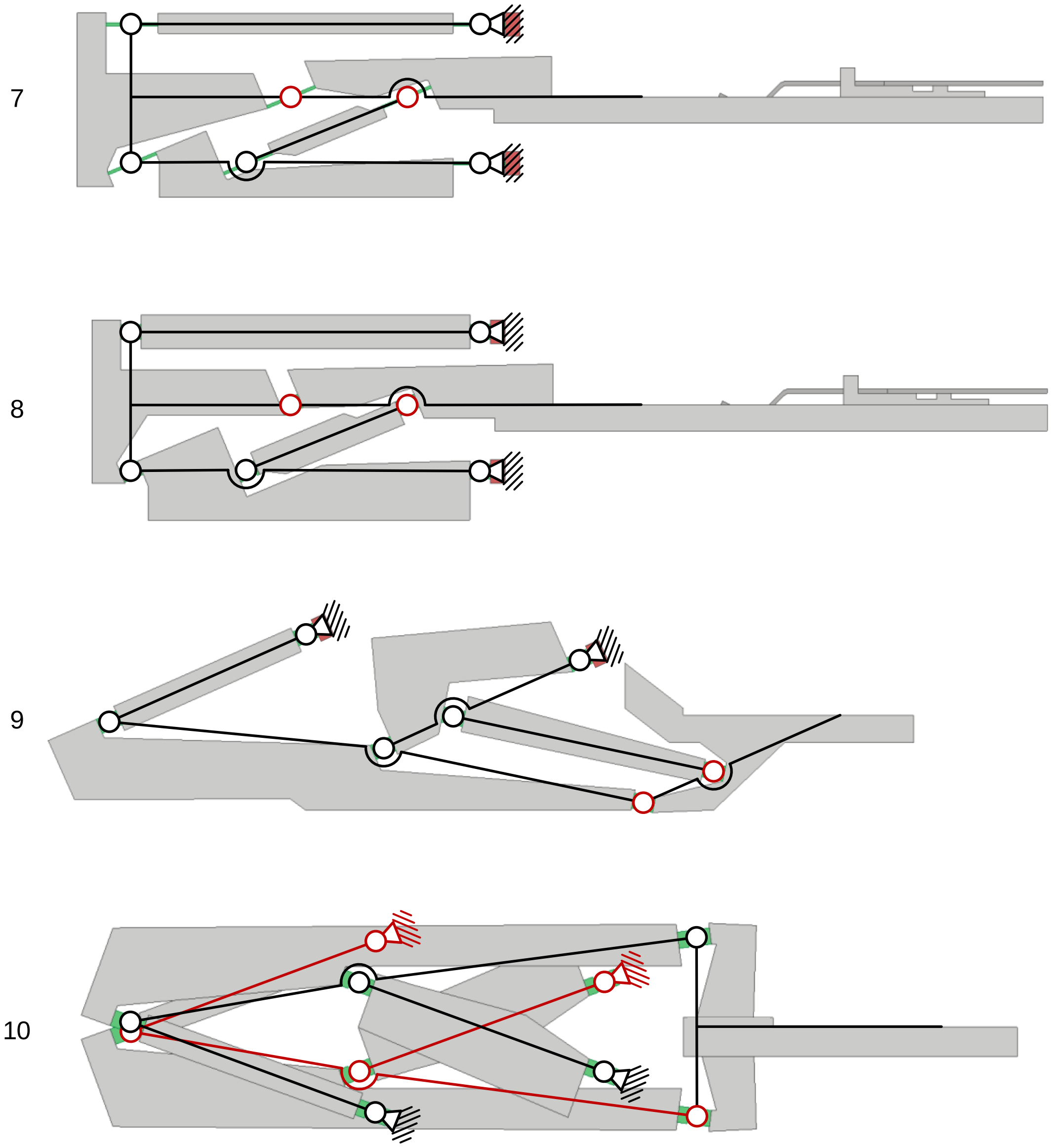


Fig. 13. The rigid body mechanics of concepts 7-10 overlaid on the side view of the concepts.

Concept 8 is identical to concept 7 except for the shape of its hinges. Concept 8 is made out of Nitinol which means that the hinges can be made more compact. This results in a mechanism that gets rid of the buckling problems of the thin and long titanium hinges of concept 7 but gets a higher sample drift in return. This comes from the non-linear elastic behavior of Nitinol.

Concept 9 uses a different configuration of the mechanism shown in Fig. 12 in comparison with concepts 7 and 8. The mechanism is fixed on the two, top links. This configuration results in a higher eigenfrequency, while still maintaining the resistance against buckling by using Nitinol but that also comes with the problem of high sample drift. By investigating the effect of non-linear elastic Nitinol on the sample drift and optimizing the mechanism, the sample drift can be lowered, but to what extent is unknown.

Finally, concept 10. It is based on the design strategy of combining multiple virtual center (VC) mechanisms defined by Zong et al. [10]. The mechanism consists of three identical VC mechanisms like the ones shown in Fig. 14. Although this is an efficient use of the available space it does require six separate connections to the sample holder. But that is not the worst of the problems, the use of Nitinol again results in high sample drift and because of the three separate structures the eigenfrequency is very low.

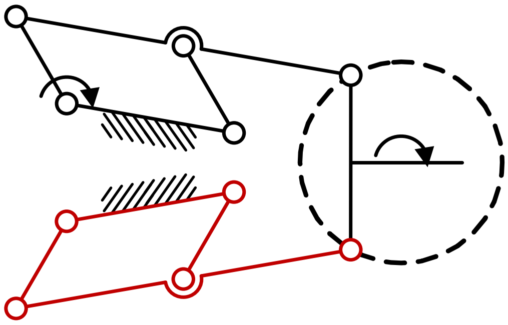


Fig. 14. Two of the three virtual center mechanisms used in concept 10 and depicted in black and red. By connecting the two mechanisms the RCM is defined.

Concluding, from the group that supports the MEMS chip from the back only concept 5 and 9 are viable, although with some optimization. But is hard to recommend any of these more complex concepts when concept 1 is this simplistic in design and manufacturing. Maybe when MEMS chip requirements become higher, the tradeoff between the drawback of a more complex concept and the benefit of more design flexibility of the MEMS chip will shift to the other side.

B.3. Nitinol

B.3.1 Theory

Several concepts use Nitinol [11]–[13] as material for their flexures. Nitinol is a nickel/titanium alloy with a nickel weight percentage of about 53% to 57%. Its name is a contraction of nickel, titanium and the place it was invented; the Naval Ordnance Laboratory. Nitinol is a shape memory alloy (SMA) that has two types of behavior; either it behaves as an alloy that can be plastically deformed which returns to its original shape when it is heated, or it behaves as a superelastic material. Both behaviors are a result of the phase transformation between two crystal structures: martensite and austenite. Which of the two behaviors the alloy shows is dependent on the temperature; when the material is below the so called austenite finish temperature (Af) it will behave like a shape memory alloy, when the Nitinol is above the austenite finish temperature it will behave as a superelasic material. The austenite finish temperature is dependent on the ratio between nickel and titanium in the Nitinol and thus is a material property.

Below the austenite finish temperature, the Nitinol has the martensite crystal structure and the material can be easily deformed. When the temperature is then increased above the austenite finish temperature the Nitinol will transform from the martensite phase into the austenite phase rearranging its atoms and jumping back into its original shape. The original shape can be set by heating the Nitinol far beyond the austenite finish temperature. This a fascinating behavior but is not very relevant for compliant mechanisms.

The superelastic behavior on the other hand, is very valuable in compliant mechanisms. When the temperature of Nitinol is above the austenite finish temperature the Nitinol will be in the austenite crystal structure (Fig. 15). If then the stress on the Nitinol is increased it will show linear elastic stress/strain behavior (Fig. 16) up to a certain stress level, when the loading plateau is reached. At this transformation stress the Nitinol will change phase from the austenite crystal structure to the twinned martensite crystal structure. During this phase change large strains can happen while there is a minimal change in stress increase. At the end of the loading plateau the Nitinol has fully transformed into detwinned martensite. Increasing the stress even further will result in linear elastic deformation of the detwinned martensite. When the Nitinol is then unloaded it will follow a different path based on the austenite start temperature (A_s).

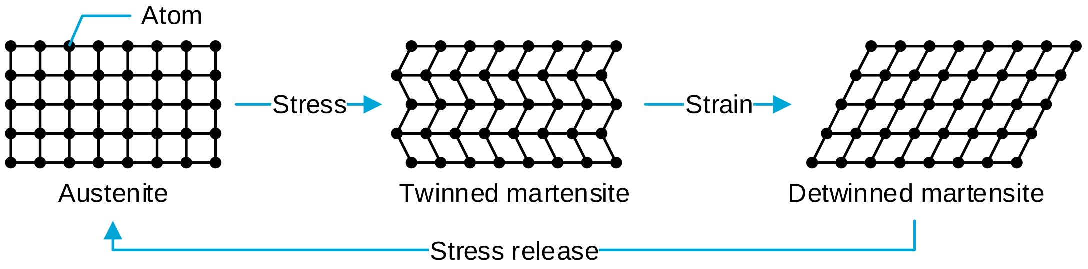


Fig. 15. The phase transformation of superelastic Nitinol as a result of stress and strain.

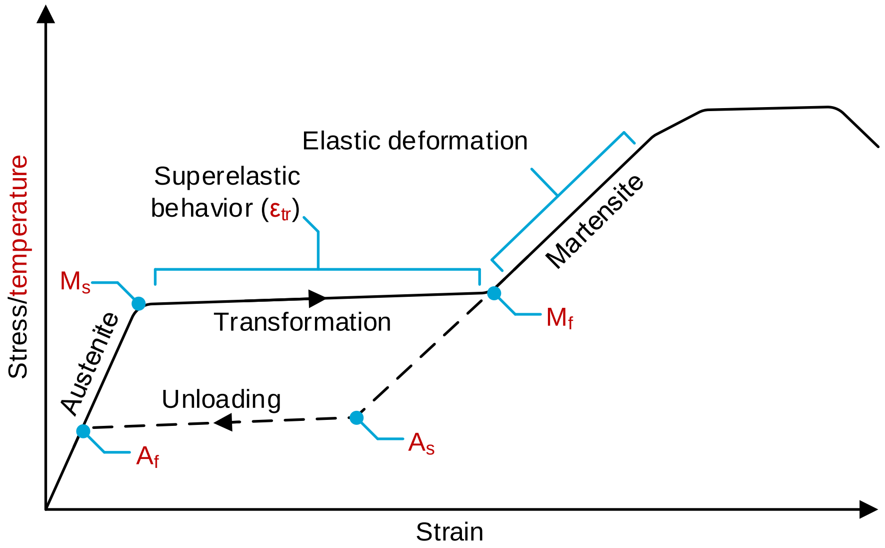


Fig. 16. The stress/strain curve of Nitinol with the transformation temperatures; austenite finish temperature (A_f), austenite start temperature (A_s), martensite start temperature (M_s) and martensite finish temperature (M_f).

In comparison with more common materials used in complaint mechanisms like spring steel or titanium, Nitinol can recover from very high strains (9%-10%). A comparison of strain rates and Young's moduli of different materials is shown in Table 3. Most often an increase in elastic strain will come at the cost of low Young's modulus but for Nitinol this is not the case. Thus, a complaint mechanism using Nitinol can still have decent stiffness, which in the case of the concepts introduced in the previous chapter means a higher lowest eigenfrequency and better stability. Finally, when a Nitinol flexure is deformed it will show lumped compliance because at the location where the stresses are highest the superelastic behavior will first occur. This means that the rotation is much more localized which is beneficial for compliant mechanism that are based on the kinematics of rigid body mechanisms.

TABLE 3
STRAIN RATES [14]

Material	Young's modulus [GPa]	Elastic strain [%]
NiTiNOL	2-55	≈10
PEEK	4	2.43
Beryllium Copper	115	0.94
Ti 6Al 4V (grade5)	114	0.77
Spring steel	183	0.27

B.3.2 Modeling

To model Nitinol in COMSOL first a model that uses a stress dependent Young's modulus was used. This means that at a certain stress level, the Young's modulus of the Nitinol will be lowered, simulating the superelastic behavior. But this way of modeling has the inherent problem that the Nitinol around the location where the Nitinol switches first to its superelastic behavior will see a reduction in stress and thus keep it from phase changing. So, instead of the expected smooth transition in Young's modulus, the beam in Fig. 17 shows a checkerboard pattern, which is the result of localization. The first locations where the Nitinol changes into superelastic material will depend in this case on the mesh.

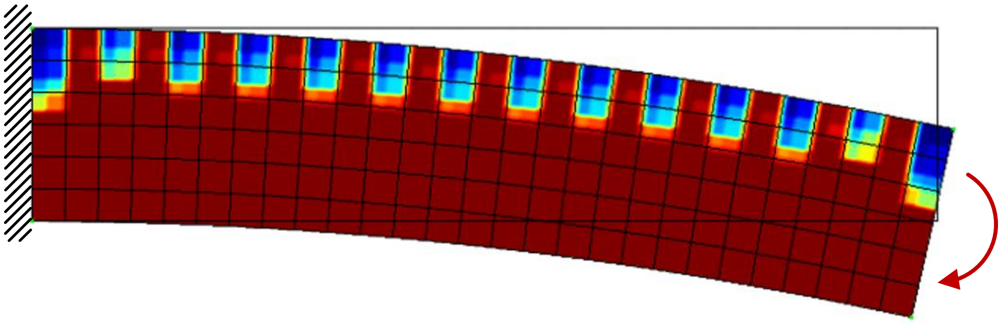


Fig. 17. A plot of the Young's modulus in a Nitinol beam when a moment is applied on the right side. The localization is shown, that occurs due to the local reduction of the stresses when a nearby node switches to the superelastic behavior.

Instead, the shape memory alloy material model in COMSOL is used. This material model has the four red parameters of Fig. 16 as its inputs; the austenite start and finish temperature (A_s, A_f), the martensite start and finish temperature (M_s, M_f), and the transformation strain ϵ_{tr}. The difficulty lies in obtaining the correct parameters. The material properties of Nitinol are very sensitive to the ratio between nickel and titanium and this ratio is difficult to keep consistent during the manufacturing process. The different types of Nitinol are mostly defined and sorted after the production process by determining the nickel/titanium ratio. Therefore, there is a big chance that actual obtained Nitinol will have different properties than the Nitinol simulated using the SMA material model in COMSOL. It is common that when Nitinol is used for a mechanism, mechanical testing will be done to validate the simulation results. The numerical results in Table 2 should thus be taken with a grain of salt. [11]–[13]

C Analytical Modeling

C.1. Rigid-Body Kinematics

At the basis of all the analytical results obtained is the kinematic model of the proof-of-principle mechanism. The Matlab code, named f_mechanismkinematics, of this model can be found in appendix D. In Fig. 18 the proof-of-principle mechanism is shown with the numbering of the hinges and links.

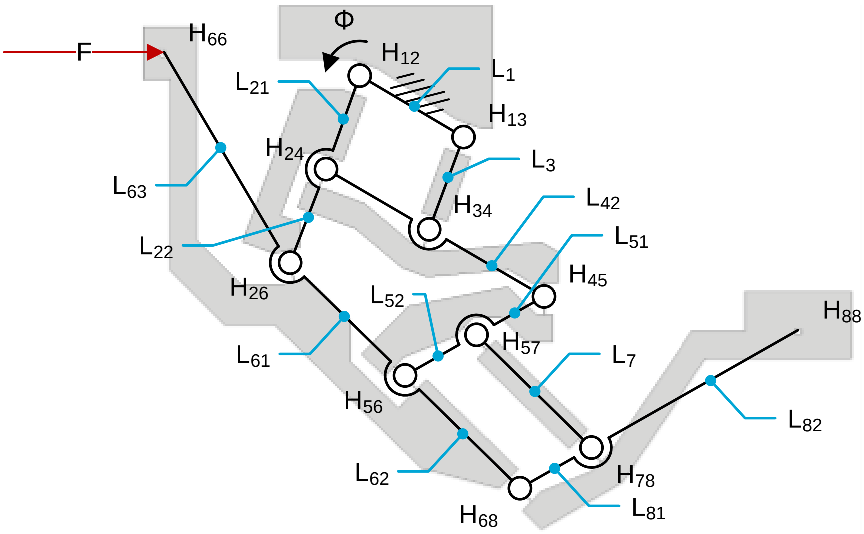


Fig. 18. The numbering of the hinges and links in the proof-of-principle mechanism.

To define the rotated coordinates of the hinges when a rotation of the input link Φ is introduced, three functions are crucial. First of all, a function that calculates the connection point of two links that are fixed on one end. The coordinates of those ends are $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ and the length of the links are l_1 and l_2 (C.1). The function gives two solutions when the links are closer than l_1+l_2 to each other. One of the coordinates of the possible connection points is kept, based on which of the two connection points is closest to $\boldsymbol{h}_3$. In other words; the sign of $\alpha \pm \beta$ is chosen so that $||\boldsymbol{h}_{out} - \boldsymbol{h}_3||$ is minimized.

$$\boldsymbol{h}_{\boldsymbol{out}}(\boldsymbol{h}_1, \boldsymbol{h}_2, l_1, l_2, \boldsymbol{h}_3) = \boldsymbol{h}_1 + \begin{bmatrix} l_1 \cos(\alpha \pm \beta) \\ l_1 \sin(\alpha \pm \beta) \end{bmatrix}$$

$$with: \boldsymbol{h}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}, i = 1, \dots, 3 \qquad (C.1)$$

$$and: \alpha = atan2\frac{y_2 - y_1}{x_2 - x_1} \quad \beta = acos\frac{l_1^2 - l_2^2 + l_3^2}{2 l_1 l_3} \quad l_3 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

The second function (C.2) calculates the angle Θ between two vectors $\boldsymbol{lv_1}$ and $\boldsymbol{lv_2}$.

$$\theta(\boldsymbol{lv_1}, \boldsymbol{lv_2}) = atan2\,\frac{x_1y_2 - y_1x_2}{x_1x_2 + y_1y_2} \qquad (C.2)$$

$$with\colon \boldsymbol{lv_i} = \begin{bmatrix} x_i \\ y_i \end{bmatrix}, i = 1,2$$

The third function (C.3) gives the rotation matrix between those two vectors and uses equation (C.2).

$$\boldsymbol{R}(lv_1, lv_2) = \begin{bmatrix} cos\big(\theta(lv_1, lv_2)\big) & -sin\big(\theta(lv_1, lv_2)\big) \\ sin\big(\theta(lv_1, lv_2)\big) & cos\big(\theta(lv_1, lv_2)\big) \end{bmatrix} \qquad (C.3)$$

Now the three functions $\boldsymbol{h}_{out}(\boldsymbol{h}_1,\boldsymbol{h}_2,l_1,l_2,\boldsymbol{h}_3)$, $\Theta(\boldsymbol{lv_1},\boldsymbol{lv_2})$ and $\mathrm{R}(\boldsymbol{lv_1},\boldsymbol{lv_2})$ are defined the relation between the initial hinge coordinates $\boldsymbol{H}$ and rotated hinge coordinates $\boldsymbol{h}$ can be defined. The 12 initial hinge coordinates of the mechanism are given in (C.4).

$$\begin{aligned}
\boldsymbol{H_{12}} &= \begin{bmatrix} -164.545 \\ 95.000 \end{bmatrix}, \boldsymbol{H_{13}} = \begin{bmatrix} -164.545 \\ 95.000 \end{bmatrix}, \boldsymbol{H_{24}} = \begin{bmatrix} -164.545 \\ 95.000 \end{bmatrix}, \boldsymbol{H_{26}} = \begin{bmatrix} -164.545 \\ 95.000 \end{bmatrix} \\
\boldsymbol{H_{34}} &= \begin{bmatrix} -164.545 \\ 95.000 \end{bmatrix}, \boldsymbol{H_{45}} = \begin{bmatrix} -164.545 \\ 95.000 \end{bmatrix}, \boldsymbol{H_{56}} = \begin{bmatrix} -164.545 \\ 95.000 \end{bmatrix}, \boldsymbol{H_{57}} = \begin{bmatrix} -164.545 \\ 95.000 \end{bmatrix} \qquad (C.4) \\
\boldsymbol{H_{68}} &= \begin{bmatrix} -164.545 \\ 95.000 \end{bmatrix}, \boldsymbol{H_{78}} = \begin{bmatrix} -164.545 \\ 95.000 \end{bmatrix}, \boldsymbol{H_{88}} = \begin{bmatrix} -164.545 \\ 95.000 \end{bmatrix}, \boldsymbol{H_{66}} = \begin{bmatrix} -164.545 \\ 95.000 \end{bmatrix}
\end{aligned}$$

Which give the initial link vectors $\boldsymbol{LV}$ in (C.5).

$$\boldsymbol{LV} = \begin{bmatrix} \boldsymbol{LV_1} \\ \boldsymbol{LV_{21}} \\ \boldsymbol{LV_{22}} \\ \boldsymbol{LV_3} \\ \boldsymbol{LV_{41}} \\ \boldsymbol{LV_{42}} \\ \boldsymbol{LV_5} \\ \boldsymbol{LV_{51}} \\ \boldsymbol{LV_{52}} \\ \boldsymbol{LV_{61}} \\ \boldsymbol{LV_{62}} \\ \boldsymbol{LV_7} \\ \boldsymbol{LV_{81}} \\ \boldsymbol{LV_{82}} \\ \boldsymbol{LV_{63}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{H_{13}} - \boldsymbol{H_{12}} \\ \boldsymbol{H_{12}} - \boldsymbol{H_{24}} \\ \boldsymbol{H_{24}} - \boldsymbol{H_{26}} \\ \boldsymbol{H_{13}} - \boldsymbol{H_{34}} \\ \boldsymbol{H_{34}} - \boldsymbol{H_{24}} \\ \boldsymbol{H_{45}} - \boldsymbol{H_{34}} \\ \boldsymbol{H_{45}} - \boldsymbol{H_{56}} \\ \boldsymbol{H_{45}} - \boldsymbol{H_{57}} \\ \boldsymbol{H_{57}} - \boldsymbol{H_{56}} \\ \boldsymbol{H_{56}} - \boldsymbol{H_{26}} \\ \boldsymbol{H_{68}} - \boldsymbol{H_{56}} \\ \boldsymbol{H_{78}} - \boldsymbol{H_{57}} \\ \boldsymbol{H_{78}} - \boldsymbol{H_{68}} \\ \boldsymbol{H_{88}} - \boldsymbol{H_{78}} \\ \boldsymbol{H_{66}} - \boldsymbol{H_{26}} \end{bmatrix} \qquad (C.5)$$

The angles between the links in the initial configuration $\boldsymbol{AH}$ are given in (C.6). The function $\Theta(\boldsymbol{lv_1},\boldsymbol{lv_2})$ is used to calculate these angles.

$$\boldsymbol{AH} = \begin{bmatrix} AH_{12} \\ AH_{13} \\ AH_{24} \\ AH_{26} \\ AH_{34} \\ AH_{45} \\ AH_{56} \\ AH_{57} \\ AH_{68} \\ AH_{78} \end{bmatrix} = \begin{bmatrix} \theta(\boldsymbol{LV_1}, \boldsymbol{LV_{21}}) \\ \theta(\boldsymbol{LV_1}, \boldsymbol{LV_3}) \\ \theta(\boldsymbol{LV_{21}}, \boldsymbol{LV_{41}}) \\ \theta(\boldsymbol{LV_{21}}, \boldsymbol{LV_{61}}) \\ \theta(\boldsymbol{LV_3}, \boldsymbol{LV_{41}}) \\ \theta(\boldsymbol{LV_{41}}, \boldsymbol{LV_5}) \\ \theta(\boldsymbol{LV_5}, \boldsymbol{LV_{61}}) \\ \theta(\boldsymbol{LV_5}, \boldsymbol{LV_7}) \\ \theta(\boldsymbol{LV_{61}}, \boldsymbol{LV_{81}}) \\ \theta(\boldsymbol{LV_7}, \boldsymbol{LV_{81}}) \end{bmatrix} \tag{C.6}$$

Now the initial state of the mechanism is fully defined. To get to the rotated hinge coordinates $\boldsymbol{h}$ as a function of the rotation Φ of L_{21} with respect to L_1 the equations in (C.7) are used. This is a stepwise process as the coordinates of hinges further away from the base are dependent on the coordinates of hinges that are closer to the base. Thus (C.7) should be evaluated from top to bottom.

$$\boldsymbol{h} = \begin{bmatrix} \boldsymbol{h_{12}} \\ \boldsymbol{h_{13}} \\ \boldsymbol{h_{24}} \\ \boldsymbol{h_{26}} \\ \boldsymbol{h_{34}} \\ \boldsymbol{h_{45}} \\ \boldsymbol{h_{56}} \\ \boldsymbol{h_{57}} \\ \boldsymbol{h_{68}} \\ \boldsymbol{h_{78}} \\ \boldsymbol{h_{88}} \\ \boldsymbol{h_{66}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{H_{12}} \\ \boldsymbol{H_{13}} \\ \boldsymbol{h_{12}} - \begin{bmatrix} cos\,\Phi & -sin\,\Phi \\ sin\,\Phi & cos\,\Phi \end{bmatrix} \boldsymbol{LV_{21}} \\ \boldsymbol{h_{24}} - \begin{bmatrix} cos\,\Phi & -sin\,\Phi \\ sin\,\Phi & cos\,\Phi \end{bmatrix} \boldsymbol{LV_{22}} \\ \boldsymbol{h_{out}}(\boldsymbol{h_{13}}, \boldsymbol{h_{24}}, \|\boldsymbol{LV_3}\|, \|\boldsymbol{LV_{41}}\|, \boldsymbol{H_{34}}) \\ \boldsymbol{h_{34}} + \boldsymbol{R}(\boldsymbol{LV_{41}}, \boldsymbol{h_{34}} - \boldsymbol{h_{24}})\boldsymbol{LV_{42}} \\ \boldsymbol{h_{out}}(\boldsymbol{h_{26}}, \boldsymbol{h_{45}}, \|\boldsymbol{L_{61}}\|, \|\boldsymbol{L_5}\|, \boldsymbol{H_{56}}) \\ \boldsymbol{h_{45}} + \boldsymbol{R}(\boldsymbol{LV_5}, \boldsymbol{h_{45}} - \boldsymbol{h_{56}})\boldsymbol{LV_{51}} \\ \boldsymbol{h_{45}} + \boldsymbol{R}(\boldsymbol{LV_{61}}, \boldsymbol{h_{56}} - \boldsymbol{h_{26}})\boldsymbol{LV_{62}} \\ \boldsymbol{h_{out}}(\boldsymbol{h_{57}}, \boldsymbol{h_{68}}, \|\boldsymbol{L_7}\|, \|\boldsymbol{L_{81}}\|, \boldsymbol{H_{78}}) \\ \boldsymbol{h_{78}} + \boldsymbol{R}(\boldsymbol{LV_{81}}, \boldsymbol{h_{78}} - \boldsymbol{h_{68}})\boldsymbol{LV_{82}} \\ \boldsymbol{h_{26}} + \boldsymbol{R}(\boldsymbol{LV_{61}}, \boldsymbol{h_{56}} - \boldsymbol{h_{26}})\boldsymbol{LV_{63}} \end{bmatrix} \tag{C.7}$$

Using the rotated hinge coordinates $\boldsymbol{h}$ the rotated link vectors $\boldsymbol{lv}$ can be calculated (C.8).

$$\boldsymbol{lv} = \begin{bmatrix} \boldsymbol{lv_1} \\ \boldsymbol{lv_{21}} \\ \boldsymbol{lv_{22}} \\ \boldsymbol{lv_3} \\ \boldsymbol{lv_{41}} \\ \boldsymbol{lv_{42}} \\ \boldsymbol{lv_5} \\ \boldsymbol{lv_{51}} \\ \boldsymbol{lv_{52}} \\ \boldsymbol{lv_{61}} \\ \boldsymbol{lv_{62}} \\ \boldsymbol{lv_7} \\ \boldsymbol{lv_{81}} \\ \boldsymbol{lv_{82}} \\ \boldsymbol{lv_{63}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{h_{13}} - \boldsymbol{h_{12}} \\ \boldsymbol{h_{12}} - \boldsymbol{h_{24}} \\ \boldsymbol{h_{24}} - \boldsymbol{h_{26}} \\ \boldsymbol{h_{13}} - \boldsymbol{h_{34}} \\ \boldsymbol{h_{34}} - \boldsymbol{h_{24}} \\ \boldsymbol{h_{45}} - \boldsymbol{h_{34}} \\ \boldsymbol{h_{45}} - \boldsymbol{h_{56}} \\ \boldsymbol{h_{45}} - \boldsymbol{h_{57}} \\ \boldsymbol{h_{57}} - \boldsymbol{h_{56}} \\ \boldsymbol{h_{56}} - \boldsymbol{h_{26}} \\ \boldsymbol{h_{68}} - \boldsymbol{h_{56}} \\ \boldsymbol{h_{78}} - \boldsymbol{h_{57}} \\ \boldsymbol{h_{78}} - \boldsymbol{h_{68}} \\ \boldsymbol{h_{88}} - \boldsymbol{h_{78}} \\ \boldsymbol{h_{66}} - \boldsymbol{h_{26}} \end{bmatrix} \tag{C.8}$$

And then the link angles in the rotated configuration $\boldsymbol{ah}$ can be calculated

$$\boldsymbol{ah} = \begin{bmatrix} ah_{12} \\ ah_{13} \\ ah_{24} \\ ah_{26} \\ ah_{34} \\ ah_{45} \\ ah_{56} \\ ah_{57} \\ ah_{68} \\ ah_{78} \end{bmatrix} = \begin{bmatrix} \theta(\boldsymbol{lv_1}, \boldsymbol{lv_{21}}) \\ \theta(\boldsymbol{lv_1}, \boldsymbol{lv_3}) \\ \theta(\boldsymbol{lv_{21}}, \boldsymbol{lv_{41}}) \\ \theta(\boldsymbol{lv_{21}}, \boldsymbol{lv_{61}}) \\ \theta(\boldsymbol{lv_3}, \boldsymbol{lv_{41}}) \\ \theta(\boldsymbol{lv_{41}}, \boldsymbol{lv_5}) \\ \theta(\boldsymbol{lv_5}, \boldsymbol{lv_{61}}) \\ \theta(\boldsymbol{lv_5}, \boldsymbol{lv_7}) \\ \theta(\boldsymbol{lv_{61}}, \boldsymbol{lv_{81}}) \\ \theta(\boldsymbol{lv_7}, \boldsymbol{lv_{81}}) \end{bmatrix} \tag{C.9}$$

The difference between the initial hinge angles $\boldsymbol{AH}$ and the rotated hinge angles $\boldsymbol{ah}$ gives a vector with the rotation of every hinge due to the input rotation Φ (C.10).

$$\boldsymbol{rh} = \boldsymbol{ah} - \boldsymbol{AH} \tag{C.10}$$

C.2. Hinge Shift Model

Hinge shift will affect the accuracy of the RCM of the proof-of-principle mechanism. There is no control over the amount of hinge shift that will occur for a defined flexure hinge. But there is control over the direction of the hinge shift by defining the orientation of the hinges in the proof-of-principle mechanism. Therefore, a hinge shift model is implemented in the kinematic model of the proof-of-principle mechanism. The hinge shift model is based on Verroti's [15] large deflection analysis of constant curvature beams. In Fig. 19 a graphical representation is shown of how the instantaneous center of rotation (ICR) of a flexure moves during deflection. The hinge shift model can be found in Matlab file f_ICRflexure in appendix D. For every step the location of the ICR is calculated by finding the intersection point of the initial and deflected symmetry lines. By adding the movement of the ICR to the hinge locations in the kinematic model of the proof-of-principle mechanism, the hinge shift can be simulated and its effect on the accuracy of the RCM (or H_{88} in Fig. 18) is calculated and optimized. The effect of hinge shift on the kinematics of the mechanism is shown in Fig. 20. The hinge shift is exaggerated to give a better view of its workings. The conclusion of this graph is that although the hinge shift is relatively large, the optimized orientations of the hinges makes sure that the combination of hinge shift in all hinges cancel each other out and give a relatively small error at the RCM location. Using the hinge shift model f_ICRflexure the orientations of the flexure hinges is optimized.

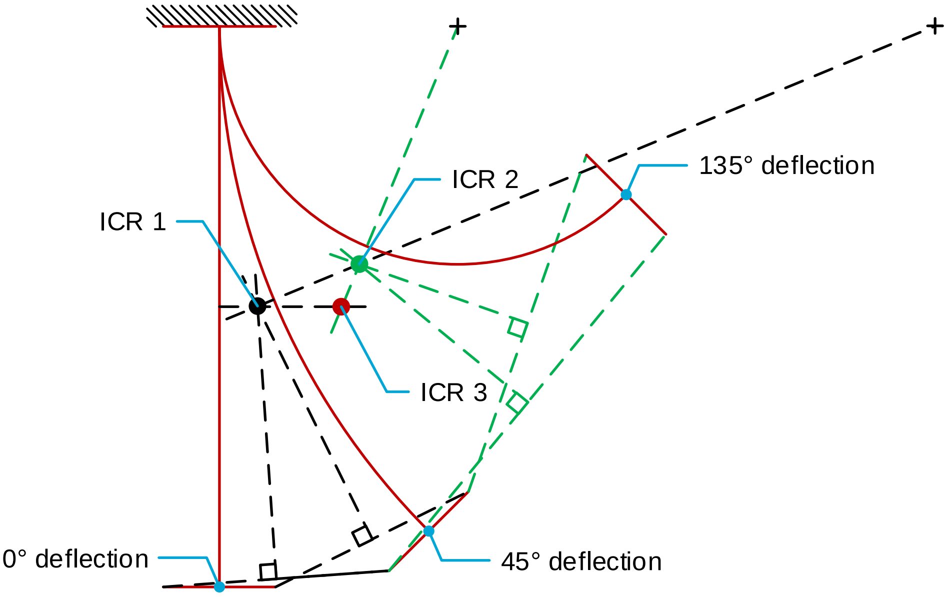


Fig. 19. A graphical representation of the hinge shift that occurs when a flexure is deflected. Three deflection configurations are shown; 0°, 45° and 135° of deflection. For a deflection of 0° to 45° the ICR is defined as ICR 1 and its location is defined by the intersection of the two bisectors, or the intersection of the two symmetry lines from the 0° and 45° deflections. For the deflection step from 45° to 135°, the ICR is defined in the same way and lies at ICR 2. ICR 3 is the instantaneous center of rotation when the flexure is deflected from 0° to 135°, which is the sum of ICR 1 and ICR 2.

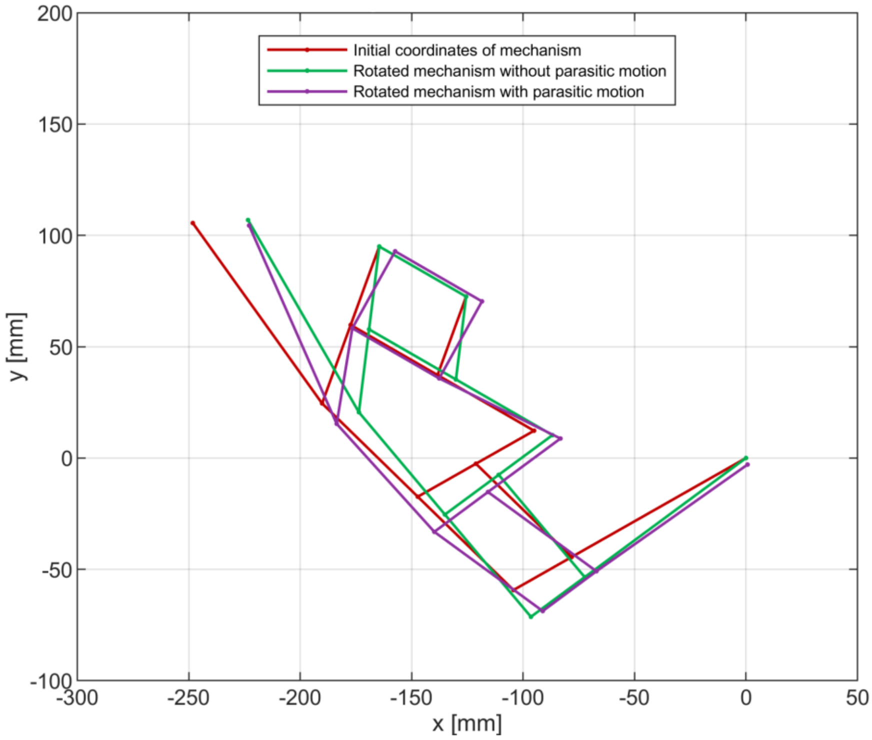


Fig. 20. The effect of hinge shift on the kinematics of the mechanism. The hinge shift is exaggerated for illustrative purposes, in reality the hinge shift is 10 times as small. It can be seen that the hinge shift is quite large everywhere, but the effect on the RCM and the actuation location is marginal. The hinges in the mechanism are orientated in such a way that their effects cancel each other out.

C.3. Pseudo-Rigid-Body Model

To simulate the force deflection behavior a pseudo-rigid-body model [16] is used as an add-on to the kinematic model of the proof-of-principle mechanism. The stiffness in the flexure hinges is simulated by torsional springs. The potential energy in these springs when the mechanism is rotated comes from the work done by the actuation force. Because of the equality between the energy in the hinges (C.11) and the work done by the actuation force (C.12), the actuation force as a function of the mechanism rotation and actuation x-displacement is calculated.

$$U = \frac{1}{2}k_{rot}\theta^2 \tag{C.11}$$

$$W = \frac{F_{max}}{2}x_{act} \tag{C.12}$$

C.4. Scaling of Proof-of-Principle Mechanism

To make sure the compliant proof-of-principle mechanism fits in the sample holder, it has to be scaled down. The effect of this scaling on the sample drift is investigated in this paragraph.

C.4.1 Hinge Shift

The first effect that has influence on the sample drift is hinge shift. The amount of hinge shift is dependent on the length of the flexure and the amount of deflection. When the proof-of-principle mechanism is scaled down 10 times, the flexure length will also scale down 10 times, but the amount of deflection will stay equal. Because the hinge shift with respect to the size of the mechanism will stay the same, the sample drift will scale 1:1 with the size of the proof-of-principle mechanism.

C.4.2 Stiffness in Hinges and Actuation

As discussed in chapter 3 the stiffness in the hinges will result in parasitic rotations of the hinges. In the free body diagram in Fig. 21 this principle is illustrated. There is an equilibrium between the moment M_{rot} as result of the rotational stiffness k_{rot} of the hinges and the moment M_{tr} created by the transverse stiffness k_{tr} of the hinges. The equations of M_{rot} and M_{tr} are given in (C.13) and (C.14) respectively. M_{rot} is dependent on the Young's modulus E, width w, height h and length l of the flexure and the rotation of the flexure Θ. M_{tr} is dependent on the same values but instead of Θ it is dependent on the parasitic displacement u at the end of the link. If the equality between M_{rot} and M_{tr} is written in terms of u the equation of (C.15) is obtained. Now it can be seen that when the proof-of-principle mechanism is scaled down 10 times, thus scaling l 10 times down, u will also scale down 10 times. When the scaling of the mechanism and the parasitic motion of the hinges is identical, this will result in a 1:1 scaling between the sample drift and the size of the mechanism, because the sample drift is a result of the parasitic hinge motion u.

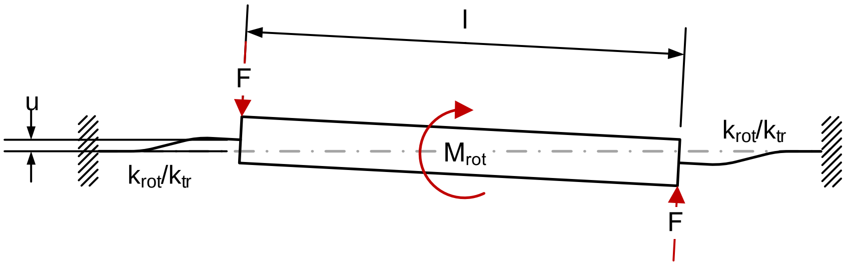

l
F
u
k_{rot}/k_{tr}
M_{rot}
k_{rot}/k_{tr}
F

Fig. 21. The free-body diagram of a link that is connected to two flexures with an equilibrium between the moment M induced by the rotational stiffness in the hinges and the forces F as a result of the transverse stiffness in the hinges.

$$\begin{gathered} M_{rot} = 2k_{rot}\theta \\ with: k_{rot} = \frac{Ewh^3}{12l} \end{gathered} \tag{C.13}$$

$$\begin{gathered} M_{tr} = Fl \\ with: F = k_{tr}u \\ with: k_{tr} = \frac{Ewh^3}{l^3} \end{gathered} \tag{C.14}$$

$$\begin{gathered} M_{rot} = M_{tr} \\ \frac{2\theta Ewh^3}{12l} = \frac{Ewh^3 l}{l^3}u \\ u = \frac{\theta}{6}l \end{gathered} \tag{C.15}$$

The actuation force and location have identical effects on the parasitic displacement in the hinges as the stiffness in the hinges, except for how the moment on the links is created. The actuation force is a result of the stiffness in the hinges and in turn this actuation force induces moments on the hinges. Therefore, the same scaling applies as for the effect of stiffness in the hinges: 1:1.

To summarize, all negative effects of the compliant proof-of-principle mechanism scale 1:1 with the size of the mechanism thus when the mechanism is scaled down, the sample drift will scale down with an equal amount.

D Matlab Code

Several Matlab functions were created to calculate the analytical result and to process data from COMSOL and the experiments (Fig. 22). The main file is RCM_error, it will run the three functions f_RCManalytical, f_RCMnumerical and f_RCMexperimental. The analytical function runs f_nohingeshift to calculate the force deflection behavior and the output/input hinge rotation ratio. f_hingeshift is run to calculate the drift of the RCM as a function of the orientation of the hinges. f_RCMnumerical processes the data from COMSOL. The force deflection behavior, the output/input hinge rotation ratio and the RCM offset and drift is calculated. The experimental results, that are obtained by running image_acquisition, are processed by f_RCMexperimental. f_image_processing_webcam and f_image_processing_micro extract the coordinates of the tracked points from the images from the webcam and microscope respectively. f_RCMcalculatorgrid calculates the RCM offset and drift from these coordinates.

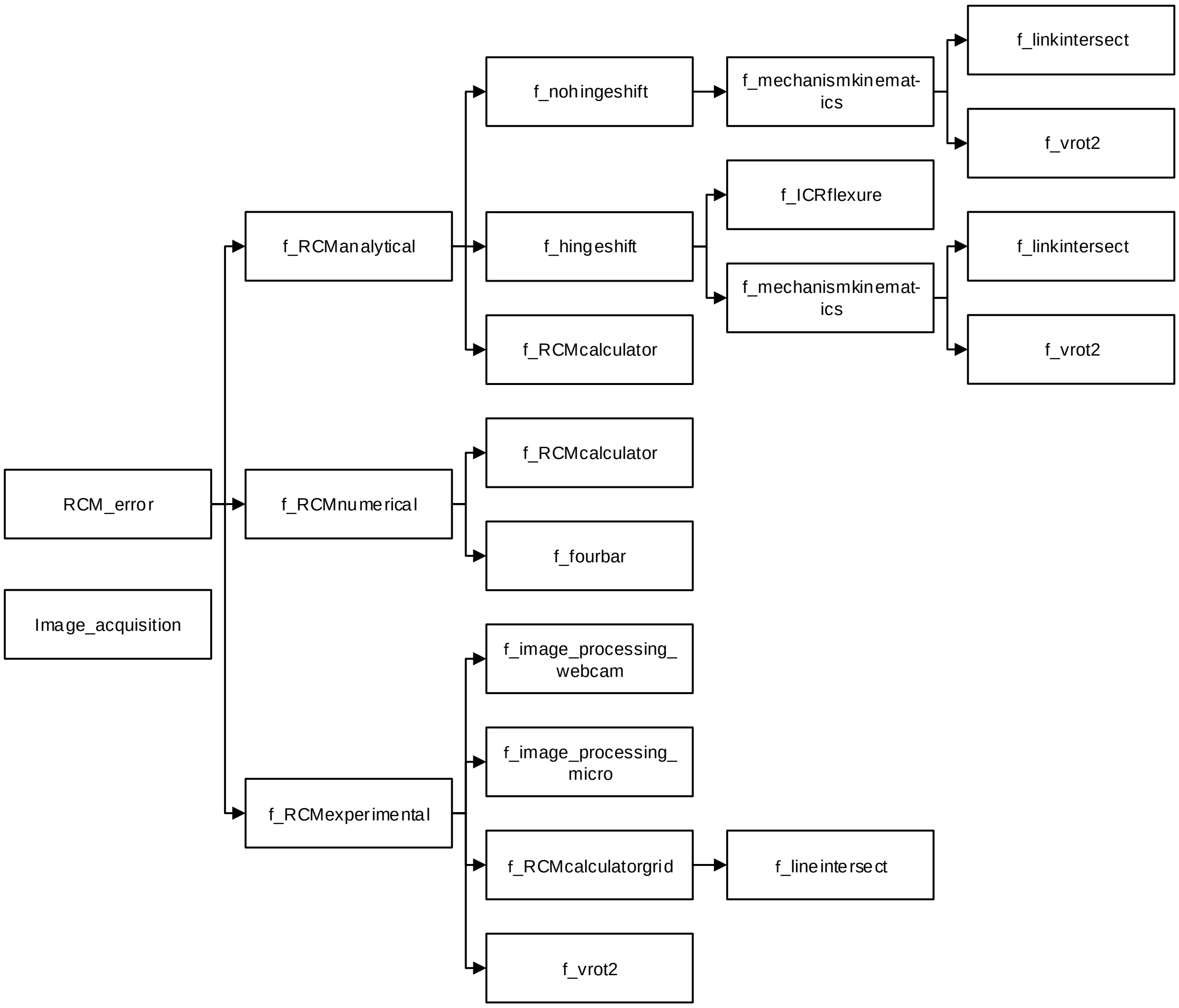


Fig. 22. An overview of the Matlab functions used to calculate and process results.

D.1. RCM_error

```
clear all
close all
clc

global i c1 c2 c3 c4 c5 c6 c7 c8 pc dataset plots
i = 0;
c1 = [192 0 0]/255;
c2 = [0 176 80]/255;
c4 = [0 166 214]/255;
c3 = [142 48 160]/255;
c5 = [0 0 0]/255;
c6 = [234 112 14]/255;
c7 = [255 192 0]/255;
c8 = [0 255 204]/255;

%% Controls
pc = 1;
plots = 1;
solutions = 1;
titlecolor = 'w';

%% Load figures
figure(1)
set(gcf,'Position',[0 600 480 400])
hold on

figure(2)
set(gcf,'Position',[480 600 480 400])
hold on

figure(3)
set(gcf,'Position',[960 600 480 400])
hold on

figure(4)
set(gcf,'Position',[1440 600 480 400])
hold on

figure(5)
set(gcf,'Position',[0 50 480 400])
hold on

%% Inputs
%hinge coordinates
h12 = [-164.545 95]';
h13 = [-125.574 72.500]';
h24 = [-177.371 59.762]';
h26 = [-190.196 24.523]';
h34 = [-138.399 37.262]';
h45 = [-95.098 12.262]';
h56 = [-147.261 -17.388]';
h57 = [-121.179 -2.563]';
h68 = [-104.325 -59.299]';
h78 = [-78.244 -44.474]';
h88 = [0 0]';
h66 = [-248.196 105.523]';
ihingecoordinates = [h12 h13 h24 h26 h34 h45 h56 h57 h68 h78 h88 h66];
ihingeorientations = [-20 -20 160 192.84 160 0.2 225.69 225.69 45.69 45.69]'; %hinge
orientations four bar 7/8

%% Analytical
[drift, magnitude] = f_RCManalytical(ihingecoordinates,ihingeorientations); %run analytical
model

%% Numerical data 1
coordinates = csvread('RCMdata25.csv',5,1); %load coordinates of measured points
rotation =   csvread('Rotation25.csv',5,1); %load rotations of links
force =         csvread('Force25.csv',5,0); %load force
[offset, drift] = f_RCMnumerical(ihingecoordinates, coordinates, rotation, force) %run
numerical model                                                %run RCM plotter function

%% Numerical data 2
if solutions == 2
i = 0;
coordinates = csvread('RCMdata26.csv',5,1); %load coordinates of measured points
rotation =   csvread('Rotation26.csv',5,1); %load rotations of links
force =         csvread('Force26.csv',5,0); %load force
[offset, drift] = f_RCMnumerical(ihingecoordinates, coordinates, rotation, force) %run
numerical model                                            %run RCM plotter function
end

%% Experimental data
numberofsteps = 100; %number of steps taken during measurement
numberofimages = 10; %number of images per step
```

```
dataset = '7'; %measurement number

nametemplate1 = strcat(dataset,'_m_0%d.tif');  %name pattern microscope images
nametemplate2 = strcat(dataset,'_w_0%d.tif');  %name pattern webcam images
filename = strcat(dataset,'.xlsx'); %filename of force deflection xlsx
f_RCMexperimental(numberofsteps,numberofimages,nametemplate1,nametemplate2,filename) %run
experimental calculations

%% Figure properties
figure(2)
title('Location of RCM (13 deg)')
xlabel('x [mm]')
ylabel('y [mm]')
h = findobj(gca,'Type','line');
L = length(h);
legend([h(L+1-1) h(L+1-4) h(L+1-2) h(L+1-8) h(L+1-11) h(L+1-19) h(L+1-20) h(L+1-21) h(L+1-
22)],{'Theoretical RCM location','0\circ rotation points', 'Numerical coordinates', 'Numerical
RCM line', 'Experimental coordinates', 'Experimental RCM line 1', 'Experimental RCM line 2',
'Experimental RCM line 3', 'Experimental RCM line 4'},'Location','NorthWest');
grid on
axis equal
axis([-5 10 -8 15])

figure(3)
title('Hingeshift in mechanism')
xlabel('x [mm]')
ylabel('y [mm]')
legend('Initial coordinates of mechanism','Rotated mechanism without parasitic
motion','Rotated mechanism with parasitic motion','Location','north')
grid on
axis([-300 50 -100 200])
set(gca,'DataAspectRatio',[1 1 1])

figure(4)
title('Force displacement (13 deg)')
xlabel('Actuation x-displacement [mm]')
ylabel('Actuation force [N]')
h = findobj(gca,'Type','line');
L = length(h);
legend([h(L+1-1) h(L+1-3) h(L+1-
5)],{'Analytical','Numerical','Experimental'},'Location','NorthWest')
grid on

figure(5)
title('Rotation ratio (13 deg)')
xlabel('Actuation x-displacement [mm]')
ylabel('Rotation ratio \theta_o_u_t/\theta_i_n')
legend('Theoretical','Theoretical','COMSOL','Experimental')
h = findobj(gca,'Type','line');
L = length(h);
legend([h(L+1-1) h(L+1-3) h(L+1-4)],{'Analytical','Numerical','Experimental'})
grid on
axis([-30 30 0.5 0.7])

%% Print figures

fh=findobj(0,'type','figure');
for n=2:length(fh)-3
    figure(n)
    fig = gcf;
    ax = gca;
    lines = get(ax, 'Children');
    set(lines, 'LineWidth', 1);
    set(fig, 'InvertHardCopy', 'off');
    set(fig,'color','w');
    T = get(gca,'Title');
    T.Color = titlecolor;
    set(ax, 'FontName', 'Arial')
    set(ax, 'fontsize', 8)
    set(ax, 'LooseInset', get(ax, 'TightInset'));
    L = get(gca,'Legend');
    L.FontSize = 6;
    box on
    fig.PaperUnits = 'centimeters';
    fig.PaperPosition = [0 0 8.9 7];  %paper 1 column
%     fig.PaperPosition = [1.4129 0 13 10]; %report block
%     fig.PaperPosition = [0 0 15.8258 7];  %report wide
%     fig.PaperPosition = [0 0 10 7];  %Powerpoint
    print('-dpng','-r500',get(get(gca,'title'),'string'))
end
```

D.2. image_acquisition

```
close all
clear all
```

```
clc

%% Settings
folder = 'C:\Users\Janeau\Google Drive\Master Thesis\Matlab\Proof of principle measurements';
%the folder
nametemplate1 = '11_w_0%d.tif';  %name pattern
nametemplate2 = '11_m_0%d.tif';  %name pattern
intervaltime = 11;
waittime = 6;
numberofsteps = 30;
numberofimages = 5;

%% Webcam

% Connect to webcam
c1 = webcam('Logitech HD Webcam C525');
c1.Resolution = '1600x896';
% c1.AvailableResolutions
c1.Brightness = 128; %default
c1.Contrast = 32; %default
c1.Saturation = 32; %default
c1.Exposure = -4;
% c1.Focus = 0; %default
c1.Focus = 50;
c1

% Setup preview window
fig = figure('NumberTitle', 'off', 'MenuBar', 'none');
fig.Name = 'Webcam';
ax = axes(fig);
frame = snapshot(c1);
im = image(ax, zeros(size(frame), 'uint8'));
axis(ax, 'image');

% Start preview
preview(c1, im);
setappdata(fig, 'cam', c1);
fig.CloseRequestFcn = @closePreviewWindow_Callback;
set(gcf,'Position',[0 500 960 500])

%% Microscope

% Connect to microscope
c2 = webcam('Digital Microscope');
c2.Resolution = '2592x1944';
% c2.AvailableResolutions

% Default settings
% c2.Saturation = 64; %0-128
% c2.Hue = 0; %-40-40
% c2.Contrast = 64; %0-64
% c2.Brightness = 0; %-64-64
% c2.Sharpness = 6; %1-6
% c2.Gamma = 100; %48-500
% c2.BacklightCompensation = 1; %0-2

% Setup preview window
fig = figure('NumberTitle', 'off', 'MenuBar', 'none');
fig.Name = 'Microscope';
ax = axes(fig);
frame = snapshot(c2);
im = image(ax, zeros(size(frame), 'uint8'));
axis(ax, 'image');

% Start preview
preview(c2, im);
setappdata(fig, 'cam', c2);
fig.CloseRequestFcn = @closePreviewWindow_Callback;
set(gcf,'Position',[960 300 960 700])

%% Take images
imnum = 0;        %starting image number
waitforbuttonpress
pause(waittime)
for K = 1 : numberofsteps+1   %if you want to do this 50 times
    tic
    disp('Pictures start')
    for i = 1:numberofimages
        imnum = imnum + 1;
        YourImage1 = snapshot(c1); %capture the image
        YourImage2 = snapshot(c2); %capture the image
        thisfile1 = sprintf(nametemplate1, imnum);  %create filename
        thisfile2 = sprintf(nametemplate2, imnum);  %create filename
        fullname1 = fullfile(folder, thisfile1);  %folder
        fullname2 = fullfile(folder, thisfile2);  %folder
        imwrite( YourImage1, fullname1);  %write the image there as tif
        imwrite( YourImage2, fullname2);  %write the image there as tif
        i
    end
    disp('Pictures taken')
    time = toc;
    K
    pause(intervaltime-time)
end


%% Local functions
function closePreviewWindow_Callback(obj, ~)
c = getappdata(obj, 'cam');
closePreview(c)
delete(obj)
end
```

D.3. f_RCManalytical

```
function [drift,offset] = f_RCManalytical(ihingecoordinates,ihingeorientations)
%This function calculates and plots the analytical RCM line, the force
%deflection and the rotation ratio. Two subfunctions, no hingeshift and
%hingeshift are run; the kinematics without and with parasitic hingeshift.

global c1 c2 c3 c4 c5 c6 force1 force2 forcetotal1 forcetotal2 rotationratio1 rotationratio2

%% Inputs
number = 30; %number of steps
stepsize = 13/number; %rotation per step
hingelength = 200; %length of hinges [mm]

%% Run hingeshift and calculate RCM
[outputcoordinates1] =
f_hingeshift(ihingecoordinates,number,stepsize,hingelength,ihingeorientations); %positive
rotation
[arcmx1, arcmy1] = f_RCMcalculator(outputcoordinates1);

[outputcoordinates2] = f_hingeshift(ihingecoordinates,number,-
stepsize,hingelength,ihingeorientations); %negative rotation
[arcmx2, arcmy2] = f_RCMcalculator(outputcoordinates2);

% Save rcm curve data to base workspace
assignin('base','arcmx1',arcmx1);
assignin('base','arcmy1',arcmy1);
assignin('base','arcmx2',arcmx2);
assignin('base','arcmy2',arcmy2);

%% Run nohingeshift and calculate force and rotation ratio
[outputcoordinates, force1, forcetotal1, rotationratio1] =
f_nohingeshift(ihingecoordinates,number,stepsize,hingelength,ihingeorientations); %positive
rotation

[outputcoordinates, force2, forcetotal2, rotationratio2] =
f_nohingeshift(ihingecoordinates,number,-stepsize,hingelength,ihingeorientations); %negative
rotation

%% Calculate drift and offset
drift = norm([arcmx2(length(arcmx2)); arcmy2(length(arcmy2))] - [arcmx1(length(arcmx1));
arcmy1(length(arcmy1))]);
for m = 1 : length(arcmx1)-1
    off1(m) = norm([arcmx1(m+1)-arcmx1(m); arcmy1(m+1)-arcmy1(m)]);
    off2(m) = norm([arcmx2(m+1)-arcmx2(m); arcmy2(m+1)-arcmy2(m)]);
end
offset = sum(off1)+sum(off2);

%% Plots
% RCM
figure(2)
plot(0, 0, '+','markers',5,'Color',c5) %plot theoretical rcm
plot(outputcoordinates1(:,1),outputcoordinates1(:,3),'.-
',outputcoordinates1(:,2),outputcoordinates1(:,4),'.-','Color',c3)   %plot measured points
lines
plot(arcmx1,arcmy1,'.-','Color',c3)
scatter(arcmx1,arcmy1,5,linspace(1,10,length(arcmx1)))

plot(outputcoordinates2(:,1),outputcoordinates2(:,3),'.-
',outputcoordinates2(:,2),outputcoordinates2(:,4),'.-','Color',c4)   %plot measured points
lines
plot(arcmx2,arcmy2,'.-','Color',c4)
scatter(arcmx2,arcmy2,5,linspace(1,10,length(arcmx2)))
end
```

D.4. f_RCMnumerical

```
function [offset, drift] = f_RCMnumerical(ihingecoordinates, coordinates,rotation,force)
% This function adjusts the dataset to the range of +-13 degrees, runs f_RCMcalculator,
% calculates the offset and drift of the RCM and plots this together with the force deflection
behavior
% annd the rotation ratio.

global i c1 c2 c3 c4 c5 c6 c7 c8 force1 force2 forcetotal1 forcetotal2 rotationratio1
rotationratio2

%% Adjust the datasets
keep = find(rotation(:,1) >= -13.2 & rotation(:,1) <= 13.2); %find the row numbers of the
rotations from -13 to 13 degrees
maxrotation = max(rotation(keep,1)) %calculate max rotation of input link
minrotation = min(rotation(keep,1)) %calculate min rotation of input link
if maxrotation <= 13 || minrotation >= -13 %show warning when rotation not reached
    msgbox('Input rotation of 13 degrees not reached');
    return
end
coordinates = coordinates(keep,:); %adjust the dataset to only contain -12 to 12 degrees of
motion
coordinatesend = find(rotation(keep,1) == minrotation(1)); %find the end of the first part of
the dataset
coordinatesstart = coordinatesend + 1;

rotation = rotation(keep,:); %adjust rotation dataset
force = force(keep,:); %adjust force dataset
minactuation = min(force(:,1))
maxactuation = max(force(:,1))

% Assign numerical coordinates data in workspace
assignin('base','coordinates',coordinates);
assignin('base','coordinatesend',coordinatesend);
assignin('base','coordinatesstart',coordinatesstart);

%% Run RCMcalculator
[nrcmx1, nrcmy1] = f_RCMcalculator(coordinates(1:coordinatesend,:));
[nrcmx2, nrcmy2] = f_RCMcalculator(coordinates(coordinatesstart:length(coordinates),:));

% Assign numerical RCM curve to workspace
assignin('base','nrcmx1',nrcmx1);
assignin('base','nrcmy1',nrcmy1);
assignin('base','nrcmx2',nrcmx2);
assignin('base','nrcmy2',nrcmy2);

%% Calculate errors
% Offset
offsetoptions(1,1) = (nrcmx1(1)^2 + nrcmy1(1)^2)^0.5;
offsetoptions(1,2) = (nrcmx2(1)^2 + nrcmy2(1)^2)^0.5;
offset = max(offsetoptions);

% Drift
RCMmax1 = [nrcmx1(length(nrcmx1)); nrcmy1(length(nrcmy1))];
RCMmax2 = [nrcmx2(length(nrcmx2)); nrcmy2(length(nrcmy2))];
drift = norm(RCMmax1 - RCMmax2)/2;

%% Calculate rotation ratio
ratio = rotation(:,7)./rotation(:,1);

%% Path of center point
distancetocenter1 = norm(coordinates(1,[1 3]));
distancetocenter2 = norm(coordinates(1,[2 4]));
prevcenter = [0; 0];
for k = 1 : length(coordinates)
    center(k,:) = f_linkintersect(coordinates(k,[1 3]),coordinates(k,[2
4]),distancetocenter1,distancetocenter2,prevcenter);
    prevcenter = center(k,:)';
end

assignin('base','centerN',center);

%% Plot coordinates and RCM line
figure(2)
hold on
rectangle('Position',[-20 -15 40 30]) %plot end link body

plot(coordinates(1:coordinatesend,1),coordinates(1:coordinatesend,3),'.-','Color',c1) %plot
measured points lines
plot(coordinates(coordinatesstart:length(coordinates),1),coordinates(coordinatesstart:length(c
oordinates),3),'.-','Color',c1)
plot(coordinates(1,1),coordinates(1,3),'.','Color',c8) %plot center point

plot(coordinates(1:coordinatesend,2),coordinates(1:coordinatesend,4),'.-','Color',c1) %plot
measured points lines
```

```
plot(coordinates(coordinatesstart:length(coordinates),2),coordinates(coordinatesstart:length(c
oordinates),4),'.-','Color',c1)
plot(coordinates(1,2),coordinates(1,4),'.','Color',c8) %plot center point

plot(nrcmx1,nrcmy1,'.-','Color',c2) %plot rcm locations line 1
plot(nrcmx2,nrcmy2,'.-','Color',c2) %plot rcm locations line 2
plot((nrcmx1(1)+nrcmx2(1))/2,(nrcmy1(1)+ nrcmy2(1))/2,'.','Color',c8) %plot center point
plot(center(1:coordinatesend,1),center(1:coordinatesend,2),'.-','Color',c6)
plot(center(coordinatesstart:end,1),center(coordinatesstart:end,2),'.-','Color',c7)

%% Run fourbar and calculate and plot theoretical angles
figure(1)
i = i + 1;
if i == 1
    theta1max = rotation(length(rotation),1);
    [rotationpoints] = imread('Rotation points.png');
    image(rotationpoints)
    axis ij
end

% Hinge coordinates
h12 = ihingecoordinates(:,1);
h24 = ihingecoordinates(:,4);
h34 = ihingecoordinates(:,9);
h13 = [0 0]';

hingecoordinates = [h12 h13 h24 h34];
theta = theta1max;

[hingecoordinates, linkrotations] = f_fourbar(hingecoordinates,theta);

figure(1)
text(52,1048+100*(i-1), strcat(num2str(rotation(length(rotation),1)),'\circ'))
text(1723,884+100*(i-1), strcat(num2str(rotation(length(rotation),2)),'\circ'))
text(1605+220*(i-1),1271, strcat(num2str(rotation(length(rotation),3)),'\circ'))
text(439,1663+100*(i-1), strcat(num2str(rotation(length(rotation),4)),'\circ'))
text(1384+220*(i-1),715, strcat(num2str(rotation(length(rotation),5)),'\circ'))
text(1530+220*(i-1),1345, strcat(num2str(rotation(length(rotation),6)),'\circ'))
text(2090,1506+100*(i-1), strcat(num2str(rotation(length(rotation),7)),'\circ'))

figure(1)
text(52, 100,strcat('Theoretical:\newlinetheta1 = ',num2str(linkrotations(2)),'\circ'))
text(52, 100,strcat('\newline\newline\newlinetheta2 = ',num2str(linkrotations(3)),'\circ'))
text(52, 100,strcat('\newline\newline\newline\newline\newlinegamma = -
',num2str(linkrotations(4)),'\circ'))

%% Plot force deflection
figure(4)
deadzone = 6;

% Stretched analytical data force
force1 = interp1(linspace(0, 1, length(force1)), force1, linspace(0,1,length(force)-
coordinatesend-deadzone)); %scale data
force2 = interp1(linspace(0, 1, length(force2)), force2, linspace(0,1,coordinatesend-
deadzone)); %scale data

plot(force(deadzone+1:coordinatesend,1),force2,'.-
',force(coordinatesstart+deadzone:length(force),1),force1,'.-','Color',c1)

% Numerical data
plot(force(1:coordinatesend,1),force(1:coordinatesend,2),'.-
',force(coordinatesstart:length(force),1),force(coordinatesstart:length(force),2),'.-
','Color',c2)

%% Plot rotation ratio
figure(5)
% Stretched analytical data
middleforce = round(length(force)/2);
rotationratio1 = interp1(linspace(0, 1, length(rotationratio1)), rotationratio1,
linspace(0,1,length(force)-middleforce-deadzone)); %scale data
rotationratio2 = interp1(linspace(0, 1, length(rotationratio2)), rotationratio2,
linspace(0,1,middleforce-deadzone)); %scale data

plot(force(deadzone+1:middleforce,1),rotationratio2,'.-
',force(deadzone+middleforce+1:end,1),rotationratio1,'.-','Color',c1)

% Numerical data
plot(force(:,1),ratio(:,1),'.-','Color',c2)

end
```

D.5. f_RCMexperimental

```
function [output] =
f_RCMexperimental(numberofsteps,numberofimages,nametemplate1,nametemplate2,filename)
```

```
%Analyzes the images from the microscope and the webcam and plots the RCM
%offset and drift which is calculated using f_RCMcalculatorgrid.
global c1 c2 c3 c4 c5 c6 c7 c8 pc dataset plots

if pc == 1
    addpath('C:\Users\User\Google Drive\Master Thesis\Matlab\Proof of principle
measurements');
    addpath('C:\Users\User\Google Drive\Master Thesis\Matlab\Proof of principle
measurements\Concept 1 measurements');
    folder = 'C:\Users\User\Google Drive\Master Thesis\Matlab\Proof of principle
measurements';  %the folder
else
    addpath('C:\Users\Janeau\Google Drive\Master Thesis\Matlab\Proof of principle
measurements');
    addpath('C:\Users\Janeau\Google Drive\Master Thesis\Matlab\Proof of principle
measurements\Concept 1 measurements');
    folder = 'C:\Users\Janeau\Google Drive\Master Thesis\Matlab\Proof of principle
measurements';  %the folder
end

%% 0 degree images

imnum = 1;         %image number at middle
nametemplate = strcat(dataset,'overview0%d.tif');  %name pattern webcam images
thisfile = sprintf(nametemplate, imnum);  %create filename
fullname = fullfile(folder, thisfile);  %folder and all
RGB = imread(fullname); %Read first image
RGB = flipud(RGB); %change image data to correct axis
figure(6)
imshow(RGB);
axis xy

imnum = 1;        %image number at middle
nametemplate = strcat(dataset,'closeup0%d.tif');  %name pattern microscope images
thisfile = sprintf(nametemplate, imnum);  %create filename
fullname = fullfile(folder, thisfile);  %folder and all
RGB = imread(fullname); %Read first image
RGB = flipud(RGB); %change image data to correct axis
figure(7)
imshow(RGB);
axis xy

%% Force deflection
fullname = fullfile(folder, filename);
dispforce = xlsread(fullname); %load force deflection data
dispforce(:,2) = dispforce(:,2)./1000-24; % m to mm and set from -24 to +26 mm
dispforce(:,4) = dispforce(:,4)*-1;

%% Image processing webcam

% Coordinates needed for scale and shift
pointexperiment = [811 729]; %fixed world
pointinventor = [-114.545 76.428]; %coordinates of same point in inventor
centerexperiment = [1573 228]; % left bottom corner of square
centerinventor = [-1,-1];
scale1 = [348 820];
scale2 = [196 461];
lengthexperiment = norm(scale2 - scale1); %length of link 7
lengthinventor = 61.211; %mm
scalefactor = lengthinventor/lengthexperiment;
shift = pointinventor - pointexperiment.*scalefactor; %shift to get 0,0 at 0,0 of inventor
centershift = (centerexperiment.*scalefactor + shift) - centerinventor %additional error
created by production/alignment inaccuraccy

centershift = [0 0];

imnum = 0;
for k = 1 : (numberofsteps+1)*numberofimages
    imnum = imnum + 1;
    thisfile = sprintf(nametemplate2, imnum);  %create filename
    fullname = fullfile(folder, thisfile);  %folder and all
    RGB = imread(fullname); %Read image
    RGB = flipud(RGB); %change image data to correct axis
    [input1(k,:),input2(k,:),input1scale(k,:),input2scale(k,:)] =
f_image_processing_webcam(RGB, shift, scalefactor, k);
    if plots == 1
        figure(7)
        plot(input1(:,1),input1(:,2),'k.-')
        figure(8)
        plot(input2(:,1),input2(:,2),'k.-')
    end
end

% Average images
input1scale = reshape(mean(reshape(input1scale,numberofimages,[]),1),[],2);
input2scale = reshape(mean(reshape(input2scale,numberofimages,[]),1),[],2);
inputvec = input2scale - input1scale;
```

```
%% Image processing microscope

% Coordinates needed for scale and shift
if dataset == '1'
    pointexperiment = [1058 662]; %point in image
    scale2 = [1592 665];
end
if dataset == '2'
    pointexperiment = [680 662]; %point in image
    scale2 = [1190 665];
end
if dataset == '3'
    pointexperiment = [695 833]; %point in image
    scale2 = [941 830];
end
if dataset == '4'
    pointexperiment = [695 833]; %point in image
    scale2 = [941 830];
end
if dataset == '5'
    pointexperiment = [695 833]; %point in image
    scale2 = [929 833];
end
if dataset == '6'
    pointexperiment = [683 848]; %point in image
    scale2 = [929 842];
end
if dataset == '7'
    pointexperiment = [686 848]; %point in image
    scale2 = [932 842];
end
if dataset == '8'
    pointexperiment = [929 779]; %point in image
    scale2 = [1121 779];
end
if dataset == '9'
    pointexperiment = [839 836]; %point in image
    scale2 = [1031 833];
end
if dataset == '10'
    pointexperiment = [839 836]; %point in image
    scale2 = [1031 833];
end
if dataset == '11'
    pointexperiment = [938 755]; %point in image
    scale2 = [1133 755];
end

pointinventor = [-1 -1]; %coordinates of same point in inventor
scale1 = pointexperiment;
lengthexperiment = norm(scale2 - scale1); %width of center square
lengthinventor = 2; %mm
scalefactor = lengthinventor/lengthexperiment;
shift = pointinventor - pointexperiment.*scalefactor; %shift to get 0,0 at 0,0 of inventor

% Red color
color = 'r';
imnum = 0;
for k = 1 : (numberofsteps+1)*numberofimages
    imnum = imnum + 1;
    thisfile = sprintf(nametemplate1, imnum);  %create filename
    fullname = fullfile(folder, thisfile);  %folder and all
    RGB = imread(fullname); %Read image
    RGB = flipud(RGB); %change image data to correct axis
    [output1(k,:),output2(k,:),output1scale(k,:),output2scale(k,:)] =
f_image_processing_micro(RGB, shift, centershift, scalefactor, color, k);
    if plots == 1
        figure(7)
        plot(output1(:,1),output1(:,2),'k.-')
        figure(8)
        plot(output2(:,1),output2(:,2),'k.-')
    end
end

% Blue color
color = 'b';
imnum = 0;
for k = 1 : (numberofsteps+1)*numberofimages
    imnum = imnum + 1;
    thisfile = sprintf(nametemplate1, imnum);  %create filename
    fullname = fullfile(folder, thisfile);  %folder and all
    RGB = imread(fullname); %Read image
    RGB = flipud(RGB); %change image data to correct axis
    [output3(k,:),output4(k,:),output3scale(k,:),output4scale(k,:)] =
f_image_processing_micro(RGB, shift, centershift, scalefactor, color, k);
    if plots == 1
```

```
        figure(7)
        plot(output3(:,1),output3(:,2),'k.-')
        figure(8)
        plot(output4(:,1),output4(:,2),'k.-')
    end
end

% Average over multiple images
output1scale = reshape(mean(reshape(output1scale,numberofimages,[]),1),[],2);
output2scale = reshape(mean(reshape(output2scale,numberofimages,[]),1),[],2);
output3scale = reshape(mean(reshape(output3scale,numberofimages,[]),1),[],2);
output4scale = reshape(mean(reshape(output4scale,numberofimages,[]),1),[],2);
outputvec = output2scale - output1scale;

% Path of center
distancetocenter1 = norm(output1scale(numberofsteps/2+1,:));
distancetocenter2 = norm(output2scale(numberofsteps/2+1,:));
prevcenter = [0; 0];
for k = 1 : numberofsteps+1
    center(k,:) =
f_linkintersect(output1scale(k,:),output2scale(k,:),distancetocenter1,distancetocenter2,prevce
nter);
    prevcenter = center(k,:)';
end

% Assign locations of tracked points and center in workspace
assignin('base','output1scale',output1scale);
assignin('base','output2scale',output2scale);
assignin('base','output3scale',output3scale);
assignin('base','output4scale',output4scale);
assignin('base','centershift',centershift);
assignin('base','centerE',center);

% Save locations of tracked points
filename = strcat(dataset,'trackedpoints.xlsx');
trackedpoints = [output1scale output2scale output3scale output4scale];
xlswrite(filename,trackedpoints)

%% RCM line calculation
coordinates1 = [output1scale(:,1) output2scale(:,1) output1scale(:,2) output2scale(:,2)];
coordinates2 = [output3scale(:,1) output4scale(:,1) output3scale(:,2) output4scale(:,2)];
coordinates3 = [output1scale(:,1) output3scale(:,1) output1scale(:,2) output3scale(:,2)];
coordinates4 = [output2scale(:,1) output4scale(:,1) output2scale(:,2) output4scale(:,2)];

[rcmx1, rcmy1] = f_RCMcalculatorgrid(coordinates1); % RCM line
[rcmx2, rcmy2] = f_RCMcalculatorgrid(coordinates2); % RCM line
[rcmx3, rcmy3] = f_RCMcalculatorgrid(coordinates3); % RCM line
[rcmx4, rcmy4] = f_RCMcalculatorgrid(coordinates4); % RCM line

%% Rotation ratio
for k = 1 : numberofsteps+1
    inoutputvec = output2scale(numberofsteps/2+1,:) - output1scale(numberofsteps/2+1,:);
    outputangle(k,:) = abs(f_vrot2(inoutputvec, outputvec(k,:)));

    ininputvec = input2scale(numberofsteps/2+1,:) - input1scale(numberofsteps/2+1,:);
    inputangle(k,:) = abs(f_vrot2(ininputvec, inputvec(k,:)));

    ratio(k) = outputangle(k,:)/inputangle(k,:);
end
ratio = ratio(1,2:end);

%% Plots
figure(2)
rectangle('Position',[centershift(1)-1 centershift(2)-1 2 2])

% Point data
plot(output1scale(:,1),output1scale(:,2),'.-','Color',c3)
plot(output2scale(:,1),output2scale(:,2),'.-','Color',c3)
plot(output3scale(:,1),output3scale(:,2),'.-','Color',c3)
plot(output4scale(:,1),output4scale(:,2),'.-','Color',c3)

% Point center
pcenter = round(length(output1scale)/2);
plot(output1scale(pcenter,1),output1scale(pcenter,2),'.','Color',c8)
plot(output2scale(pcenter,1),output2scale(pcenter,2),'.','Color',c8)
plot(output3scale(pcenter,1),output3scale(pcenter,2),'.','Color',c8)
plot(output4scale(pcenter,1),output4scale(pcenter,2),'.','Color',c8)

% RCM data
plot(rcmx1,rcmy1,'.-','Color',c4)
plot(rcmx2,rcmy2,'.-','Color',c5)
plot(rcmx3,rcmy3,'.-','Color',c6)
plot(rcmx4,rcmy4,'.-','Color',c7)

% RCM center
rcmcenter = round(length(rcmx1)/2);
plot(rcmx1(rcmcenter),rcmy1(rcmcenter),'.','Color',c8)
plot(rcmx2(rcmcenter),rcmy2(rcmcenter),'.','Color',c8)
plot(rcmx3(rcmcenter),rcmy3(rcmcenter),'.','Color',c8)
plot(rcmx4(rcmcenter),rcmy4(rcmcenter),'.','Color',c8)

plot(centershift(1),centershift(2),'+','Color',c5)
plot(center(:,1),center(:,2),'.-','Color',c6)

% plot(rcmx1,rcmy1,'r.-')

figure(4)
plot(dispforce(:,2),dispforce(:,4),'.-','Color',c3)

figure(5)
plot(dispforce(:,2),ratio(1,:),'.-','Color',c3)
end
```

D.6. f_nohingeshift

```
function [outputcoordinates, force, forcetotal, rotationratio] =
f_nohingeshift(ihingecoordinates,number,stepsize,hingelength,ihingeorientations)
%This functiom calculates the rotated mechanism when hinge shift is not taken into
%consideration. Number is the number of iterations, stepsize the size of
%each iteration in degrees.

global c1 c2 c3 c4 c5 c6
%% Plot coordinates without parasitic motion and get initial hinge orientation
th = number*stepsize;
[nphingecoordinates, nphingeangles, ilinkvectors ] =
f_mechanismkinematics(ihingecoordinates,th);

o1 = ihingeorientations(1,1); %hinge orientation angles
o2 = ihingeorientations(2,1);
o3 = ihingeorientations(3,1);
o4 = ihingeorientations(4,1);
o5 = ihingeorientations(5,1);
o6 = ihingeorientations(6,1);
o7 = ihingeorientations(7,1);
o8 = ihingeorientations(8,1);
o9 = ihingeorientations(9,1);
o10 = ihingeorientations(10,1);

R1 = [cosd(o1) -sind(o1) ;sind(o1) cosd(o1)];
R2 = [cosd(o2) -sind(o2) ;sind(o2) cosd(o2)];
R3 = [cosd(o3) -sind(o3) ;sind(o3) cosd(o3)];
R4 = [cosd(o4) -sind(o4) ;sind(o4) cosd(o4)];
R5 = [cosd(o5) -sind(o5) ;sind(o5) cosd(o5)];
R6 = [cosd(o6) -sind(o6) ;sind(o6) cosd(o6)];
R7 = [cosd(o7) -sind(o7) ;sind(o7) cosd(o7)];
R8 = [cosd(o8) -sind(o8) ;sind(o8) cosd(o8)];
R9 = [cosd(o9) -sind(o9) ;sind(o9) cosd(o9)];
R10 = [cosd(o10) -sind(o10) ;sind(o10) cosd(o10)];

ioh12 = R1*[1; 0]; %initial hinge orientation
ioh13 = R2*[1; 0];
ioh24 = R3*[1; 0];
ioh26 = R4*[1; 0];
ioh34 = R5*[1; 0];
ioh45 = R6*[1; 0];
ioh56 = R7*[1; 0];
ioh57 = R8*[1; 0];
ioh68 = R9*[1; 0];
ioh78 = R10*[1; 0];

ihingeorientation = [ioh12 ioh13 ioh24 ioh26 ioh34 ioh45 ioh56 ioh57 ioh68 ioh78];
ihingeorientation = ihingeorientation./vecnorm(ihingeorientation);

%For negative rotation
if stepsize <= 0
    ihingeorientation = ihingeorientation*-1;
end

% Plot initial hinge coordinates and rotated coordinates without parasitic motion
figure(3)
% plot(ihingecoordinates(1,[5 2 1 3 5 6 8 7 9 10 11 10 8 7 4 12 4 3]),ihingecoordinates(2,[5 2
1 3 5 6 8 7 9 10 11 10 8 7 4 12 4 3]),'.-','Color',c1)
hold on
% plot(nphingecoordinates(1,[5 2 1 3 5 6 8 7 9 10 11 10 8 7 4 12 4 3]),nphingecoordinates(2,[5
2 1 3 5 6 8 7 9 10 11 10 8 7 4 12 4 3]),'.-','Color',c2)

%% Loop
prevhingerotation = zeros(1,length(ihingeorientation)); %set initial hinge rotations to zero
prevrotpoint = zeros(2,length(ihingeorientation)); %set initial rotation points to zero
motionvsave = zeros(2,length(ihingeorientation)); %reset initial parasitic motion save vector
starthingecoordinates = ihingecoordinates; %rename initial hinge coordiantes for loop
```

```
for j = 1:number %number of steps
%% Calculate initial hinge angles
th = 0; %rotation angle
[starthingecoordinates, ihingeangles ] = f_mechanismkinematics(starthingecoordinates,th);

%% Calculate hinge coordinates and hinge angles of rotated mechanism
th = stepsize; %angle stepsize
[hingecoordinates, hingeangles, ilinkvectorsstep, linkvectors] =
f_mechanismkinematics(starthingecoordinates,th);

% plot(ihingecoordinates(1,:),ihingecoordinates(2,:),'.')
% plot(hingecoordinates(1,:),hingecoordinates(2,:),'.')

%% Calculate hinge rotation
rotationstep = abs(hingeangles - ihingeangles)
hingerotation = prevhingerotation + rotationstep

%% Calculate input and output link rotation
inputrot(j) = f_vrot2(ilinkvectors(:,2),linkvectors(:,2));
outputrot(j) = f_vrot2(ilinkvectors(:,13),linkvectors(:,13));


%% Calculate parasitic motion of icr
for k = 1:length(ihingeorientation) %calculate motion vector for every hinge
[motion(k), rotpoint(:,k)] =
f_ICRflexure(prevrotpoint(:,k),prevhingerotation(k),hingerotation(k),hingelength); %calculate
motion along symmetric axis of hinge
rot = prevhingerotation(k)/2; %rotation of symmetric axis
R = [cosd(rot) -sind(rot) ;sind(rot) cosd(rot)] ; %rotation matrix
hingeorientation(:,k) = R*ihingeorientation(:,k); %current hinge orientation
motionv(:,k) = motion(k).*hingeorientation(:,k); %parasitic motion of hinge
end

motionvsave(2*j-1:2*j,:) = motionv; %save all parasitic displacements

xsum68 = sum(motionvsave(1:2:end,9));
ysum68 = sum(motionvsave(2:2:end,9));
xsum78 = sum(motionvsave(1:2:end,10));
ysum78 = sum(motionvsave(2:2:end,10));
totalparmotion = [xsum68 xsum78; ysum68 ysum78];

shiftedstarthingecoordinates = starthingecoordinates;
shiftedstarthingecoordinates(:,1:10) = starthingecoordinates(:,1:10); %calculate new hinge
coordinates (with parasitic motion)
outputcoordinates(j,[1 2]) = shiftedstarthingecoordinates(1,[9 10]) - totalparmotion(1,:); %x
values
outputcoordinates(j,[3 4]) = shiftedstarthingecoordinates(2,[9 10]) - totalparmotion(2,:); %y
values

% plot(shiftedstarthingecoordinates(1,:),shiftedstarthingecoordinates(2,:),'.','Color',[0/255,
100/255, 100/255])

%% Run shifted coordinates of rotated mechanism
[rotatedhingecoordinates, hingeangles] =
f_mechanismkinematics(shiftedstarthingecoordinates,th);
% plot(rotatedhingecoordinates(1,:),rotatedhingecoordinates(2,:),'.','Color',[142/255, 48/255,
160/255])

% outputcoordinates(j,[1 2]) = rotatedhingecoordinates(1,[9 10]) - totalparmotion(1,:); %x
values
% outputcoordinates(j,[3 4]) = rotatedhingecoordinates(2,[9 10]) - totalparmotion(2,:); %y
values

%% Calculate potential energy in mechanism and actuation force

% Leaf sprinng characteristics
w = 8*10^-3; %mm*10^-3
h = 0.15*10^-3;
Iy = 1/12*w*h^3;
E = 183*10^9;
l = 12*10^-3;

stiff = E*Iy/l;
dxactuator(j) = rotatedhingecoordinates(1,12)-starthingecoordinates(1,12); %%mm
xactuator(j) = rotatedhingecoordinates(1,12)-ihingecoordinates(1,12); %%mm

forcestep(j) = 2*sum(1/2*stiff.*(deg2rad(rotationstep)).^2)/(dxactuator(j)/1000);
force(j) = sum(forcestep);

forcetotal(j) =
2*sum(stiff*1/2*deg2rad(hingerotation).*deg2rad(hingerotation))/(xactuator(j)/1000);

% potenergystep(j) = sum(stiff*deg2rad(hingerotation).*deg2rad(rotationstep));
% forcetotalstep(j) = 2*potenergystep(j)/(dxactuator(j)/1000)
% forcetotal(j) = sum(forcetotalstep);

%% Update values for new loop
```

```
prevrotpoint = rotpoint;
prevhingerotation = hingerotation;
starthingecoordinates = rotatedhingecoordinates;

end
rotationratio = outputrot./inputrot;

%% Plot rotated mechanism with parasitic motion
figure(3)
% plot(outputcoordinates(:,1),outputcoordinates(:,3),'g.')
% plot(rotatedhingecoordinates(1,[5 2 1 3 5 6 8 7 9 10 11 10 8 7 4 12 4
3]),rotatedhingecoordinates(2,[5 2 1 3 5 6 8 7 9 10 11 10 8 7 4 12 4 3]),'.-','Color',c3)
end
```

D.7. f_hingeshift

```
function [outputcoordinates, force, dxactuatormiddle, xactuator, rotationratio] =
f_hingeshift(ihingecoordinates,number,stepsize,hingelength,ihingeorientations)
%This function calculates the rotated mechanism when hinge shift is taken into
%consideration. Number is the number of iterations, stepsize the size of
%each iteration in degrees.

global c1 c2 c3 c4 c5 c6
%% Plot coordinates without parasitic motion and get initial hinge orientation
th = number*stepsize;
[nphingecoordinates, nphingeangles, ilinkvectors ] =
f_mechanismkinematics(ihingecoordinates,th);

o1 = ihingeorientations(1,1); %hinge orientation angles
o2 = ihingeorientations(2,1);
o3 = ihingeorientations(3,1);
o4 = ihingeorientations(4,1);
o5 = ihingeorientations(5,1);
o6 = ihingeorientations(6,1);
o7 = ihingeorientations(7,1);
o8 = ihingeorientations(8,1);
o9 = ihingeorientations(9,1);
o10 = ihingeorientations(10,1);

R1 = [cosd(o1) -sind(o1) ;sind(o1) cosd(o1)];
R2 = [cosd(o2) -sind(o2) ;sind(o2) cosd(o2)];
R3 = [cosd(o3) -sind(o3) ;sind(o3) cosd(o3)];
R4 = [cosd(o4) -sind(o4) ;sind(o4) cosd(o4)];
R5 = [cosd(o5) -sind(o5) ;sind(o5) cosd(o5)];
R6 = [cosd(o6) -sind(o6) ;sind(o6) cosd(o6)];
R7 = [cosd(o7) -sind(o7) ;sind(o7) cosd(o7)];
R8 = [cosd(o8) -sind(o8) ;sind(o8) cosd(o8)];
R9 = [cosd(o9) -sind(o9) ;sind(o9) cosd(o9)];
R10 = [cosd(o10) -sind(o10) ;sind(o10) cosd(o10)];

ioh12 = R1*[1; 0]; %initial hinge orientation
ioh13 = R2*[1; 0];
ioh24 = R3*[1; 0];
ioh26 = R4*[1; 0];
ioh34 = R5*[1; 0];
ioh45 = R6*[1; 0];
ioh56 = R7*[1; 0];
ioh57 = R8*[1; 0];
ioh68 = R9*[1; 0];
ioh78 = R10*[1; 0];

ihingeorientation = [ioh12 ioh13 ioh24 ioh26 ioh34 ioh45 ioh56 ioh57 ioh68 ioh78];
ihingeorientation = ihingeorientation./vecnorm(ihingeorientation);

% For negative rotation
if stepsize <= 0
    ihingeorientation = ihingeorientation*-1;
end

% Plot initial hinge coordinates and rotated coordinates without parasitic motion
figure(3)
plot(ihingecoordinates(1,[5 2 1 3 5 6 8 7 9 10 11 10 8 7 4 12 4 3]),ihingecoordinates(2,[5 2 1
3 5 6 8 7 9 10 11 10 8 7 4 12 4 3]),'.-','Color',c1)
hold on
plot(nphingecoordinates(1,[5 2 1 3 5 6 8 7 9 10 11 10 8 7 4 12 4 3]),nphingecoordinates(2,[5 2
1 3 5 6 8 7 9 10 11 10 8 7 4 12 4 3]),'.-','Color',c2)

%% Loop
prevhingerotation = zeros(1,length(ihingeorientation)); %set initial hinge rotations to zero
prevrotpoint = zeros(2,length(ihingeorientation)); %set initial rotation points to zero
motionvsave = zeros(2,length(ihingeorientation)); %reset initial parasitic motion save vector
starthingecoordinates = ihingecoordinates; %rename initial hinge coordiantes for loop

for j = 1:number %number of steps
%% Calculate initial hinge angles
```

```
th = 0; %rotation angle
[starthingecoordinates, ihingeangles ] = f_mechanismkinematics(starthingecoordinates,th);

%% Calculate hinge coordinates and hinge angles of rotated mechanism
th = stepsize; %angle stepsize
[hingecoordinates, hingeangles, ilinkvectorsstep, linkvectors] =
f_mechanismkinematics(starthingecoordinates,th);

% plot(ihingecoordinates(1,:),ihingecoordinates(2,:),'.')
% plot(hingecoordinates(1,:),hingecoordinates(2,:),'.')

%% Calculate hinge rotation
hingerotation = prevhingerotation + abs(hingeangles - ihingeangles);

%% Calculate input and output link rotation
inputrot(j) = f_vrot2(ilinkvectors(:,2),linkvectors(:,2));
outputrot(j) = f_vrot2(ilinkvectors(:,13),linkvectors(:,13));


%% Calculate parasitic motion of icr
for k = 1:length(ihingeorientation) %calculate motion vector for every hinge
[motion(k), rotpoint(:,k)] =
f_ICRflexure(prevrotpoint(:,k),prevhingerotation(k),hingerotation(k),hingelength); %calculate
motion along symmetric axis of hinge
rot = prevhingerotation(k)/2; %rotation of symmetric axis
R = [cosd(rot) -sind(rot) ;sind(rot) cosd(rot)] ; %rotation matrix
hingeorientation(:,k) = R*ihingeorientation(:,k); %current hinge orientation
motionv(:,k) = motion(k).*hingeorientation(:,k); %parasitic motion of hinge
end

motionvsave(2*j-1:2*j,:) = motionv; %save all parasitic displacements

xsum68 = sum(motionvsave(1:2:end,9));
ysum68 = sum(motionvsave(2:2:end,9));
xsum78 = sum(motionvsave(1:2:end,10));
ysum78 = sum(motionvsave(2:2:end,10));
totalparmotion = [xsum68 xsum78; ysum68 ysum78];

shiftedstarthingecoordinates = starthingecoordinates;
shiftedstarthingecoordinates(:,1:10) = starthingecoordinates(:,1:10) + motionv; %calculate new
hinge coordinates (with parasitic motion)
outputcoordinates(j,[1 2]) = shiftedstarthingecoordinates(1,[9 10]) - totalparmotion(1,:); %x
values
outputcoordinates(j,[3 4]) = shiftedstarthingecoordinates(2,[9 10]) - totalparmotion(2,:); %y
values

% plot(shiftedstarthingecoordinates(1,:),shiftedstarthingecoordinates(2,:),'.','Color',[0/255,
100/255, 100/255])

%% Run shifted coordinates of rotated mechanism
[rotatedhingecoordinates, hingeangles] =
f_mechanismkinematics(shiftedstarthingecoordinates,th);
% plot(rotatedhingecoordinates(1,:),rotatedhingecoordinates(2,:),'.','Color',[142/255, 48/255,
160/255])

% outputcoordinates(j,[1 2]) = rotatedhingecoordinates(1,[9 10]) - totalparmotion(1,:); %x
values
% outputcoordinates(j,[3 4]) = rotatedhingecoordinates(2,[9 10]) - totalparmotion(2,:); %y
values

%% Calculate potential energy in mechanism and actuation force

% Leaf spring characteristics
w = 8*10^-3; %mm*10^-3
h = 0.15*10^-3;
Iy = 1/12*w*h^3;
E = 183*10^9;
l = 12*10^-3;

stiff = E*Iy/l;
potenergyincrease = sum(0.5*stiff.*(deg2rad(hingeangles - ihingeangles)).^2); %potential
energy of step
totalpotenergy = sum(0.5*stiff.*(deg2rad(hingeangles)).^2); %total potential energy

dxactuator(j) = rotatedhingecoordinates(1,12)-starthingecoordinates(1,12); %%mm
xactuator(j) = rotatedhingecoordinates(1,12)-ihingecoordinates(1,12); %%mm
dxactuatormiddle(j) = xactuator(j) - dxactuator(j)/2; %mm

forceincrease(j) = potenergyincrease/(dxactuator(j)/1000);
force(j) = sum(forceincrease);
forcetotal(j) = totalpotenergy/(xactuator(j)/1000);
forcetotal1 = stiff.*deg2rad(hingeangles).*deg2rad(hingeangles -
ihingeangles)/(dxactuator(j)/1000);

%% Update values for new loop
prevrotpoint = rotpoint;
prevhingerotation = hingerotation;
```

```
starthingecoordinates = rotatedhingecoordinates;

end
rotationratio = outputrot./inputrot;

%% Plot rotated mechanism with parasitic motion
figure(3)
% plot(outputcoordinates(:,1),outputcoordinates(:,3),'g.')
plot(rotatedhingecoordinates(1,[5 2 1 3 5 6 8 7 9 10 11 10 8 7 4 12 4
3]),rotatedhingecoordinates(2,[5 2 1 3 5 6 8 7 9 10 11 10 8 7 4 12 4 3]),'.-','Color',c3)
end
```

D.8. f_RCMcalculator

```
function [rcmx, rcmy] = f_RCMcalculator(coordinates)
%Calculates the location of the RCM by using the data of the movement of
%two points on the final link.

coordinates = coordinates(1:round(length(coordinates)/30):end,:);

th = 90; %rotation angle
R = [cosd(th) -sind(th) ;sind(th) cosd(th)]; %rotation matrix
for i = 1:length(coordinates)-1
    dx1(i,1) = (coordinates(i+1,1)-coordinates(i,1)); %x movement
    dy1(i,1) = (coordinates(i+1,3)-coordinates(i,3)); %x movement
    slopevector1(i,1:2) = (R*[dx1(i,1); dy1(i,1)])'; %calculate slope vector
    slope1(i,1) = slopevector1(i,2)/slopevector1(i,1); %calculate slope
    center1(i,:) = [coordinates(i,1)+dx1(i,1)/2; coordinates(i,3)+dy1(i,1)/2]; %calculate
midpoint on displacement vector
    height1(i,1) = center1(i,2)-slope1(i,1)*center1(i,1); %calculate b value in ax+b line

    dx2(i,1) = (coordinates(i+1,2)-coordinates(i,2));
    dy2(i,1) = (coordinates(i+1,4)-coordinates(i,4));
    slopevector2(i,1:2) = (R*[dx2(i,1); dy2(i,1)])';
    slope2(i,1) = slopevector2(i,2)/slopevector2(i,1);
    center2(i,:) = [coordinates(i,2)+dx2(i,1)/2; coordinates(i,4)+dy2(i,1)/2];
    height2(i,1) = center2(i,2)-slope2(i,1)*center2(i,1);

    rcmx(i,1) = (height2(i,1)-height1(i,1))/(slope1(i,1)-slope2(i,1)); %x coordinate of RCM
    rcmy(i,1) = slope1(i,1)*rcmx(i,1)+height1(i,1); %y coordinate of RCM
end
end
```

D.9. f_fourbar

```
function [hingecoordinates, linkrotations] = f_fourbar(hingecoordinates,rotationangle)
%Calculation of coordinates of hinge points of four bar mechanism when the
%input link is rotated
h12 = hingecoordinates(:,1); %hinge coordinates
h13 = hingecoordinates(:,2);
h24 = hingecoordinates(:,3);
h34 = hingecoordinates(:,4);

th = rotationangle; %rotation angle

ilv1 = h13 - h12; %link vectors
ilv2 = h12 - h24;
ilv3 = h13 - h34;
ilv4 = h34 - h24;

ill1 = norm(ilv1); %link lengths
ill2 = norm(ilv2);
ill3 = norm(ilv3);
ill4 = norm(ilv4);

% Calculate h24
R = [cosd(th) -sind(th) ;sind(th) cosd(th)]; %rotation matrix
h24 = h12 - R*ilv2;

% Calculate h34
h34 = f_linkintersect(h24,h13,ill4,ill3,h34);

hingecoordinates = [h12 h13 h24 h34];

% Calculate link vectors
lv1 = h13 - h12; %link vectors
lv2 = h12 - h24;
lv3 = h13 - h34;
lv4 = h34 - h24;

% Calculate link rotations
rl1 = f_vrot(ilv1,lv1);
rl2 = f_vrot(ilv2,lv2);
```

```
rl3 = f_vrot(ilv3,lv3);
rl4 = f_vrot(ilv4,lv4);

linkrotations = [rl1 rl2 rl3 rl4]';
end
```

D.10.f_image_processing_webcam

```
function [input1,input2,input1scale,input2scale] = f_image_processing_webcam(RGB, shift,
scalefactor, k)
%Analyze RGB webcam image data and output the location of the dots.

global irangex3 irangey3 irangex4 irangey4 plots
% Search settings
filt1 = 0.3; %Value threshold of first filter
filt2 = 0.3; %saturation threshold of second filter
filt3 = 25/360; %saturation threshold of third filter
filt4 = 0.60; %saturation threshold of fourth filter
searcharea = 25; %size of search area for second filter
plotarea = 150; %size of additional area around searchearea to plot

HSV = rgb2hsv(RGB);                                      %Convert to hsv
GRY = rgb2gray(RGB);%Convert to grayscale
GRY = im2double(GRY);                                     %Convert to double


%% Analysis
H = HSV(:,:,1);
S = HSV(:,:,2);
V = HSV(:,:,3);

% Coordinate matrix
M = zeros(size(H));
M(400:end,150:650) = 1; %select area

%% Remove pixels with low value

M(V <= filt1) = 0;
[yf2, xf2] = find(M == 1);

%% Get pixels with high saturation
M(S <= filt2) = 0;
[yf3, xf3] = find(M == 1);

%% Get pixels that have a certain color
%remove pixels that are within 20/350 bounds of red hue (or 160/260 for blue, 80/160 for
green)

M(H >= filt3) = 0; %red
[yf4, xf4] = find(M == 1);

% M(H < (160/360)) = 0; %blue
% M(H > (260/360)) = 0;

% M(H < (60/360)) = 0; %green
% M(H > (180/360)) = 0;

%% Seperate input1 and input2 link pixels

% Average recognized point locations
meanx1 = mean(xf4);
meany1 = mean(yf4);

% Split point data between input and input link
xinput1 = xf4(yf4 <= meany1);
xinput2 = xf4(yf4 >= meany1);
yinput1 = yf4(yf4 <= meany1);
yinput2 = yf4(yf4 >= meany1);

%% Average of two point collections
% Combine x and y data
input1 = [xinput1 yinput1];
input2 = [xinput2 yinput2];

% Calculate center of points
input1 = mean(input1);
input2 = mean(input2);

%% Increase accuracy of input1

rangex3 = round(input1(1,1))-searcharea:round(input1(1,1))+searcharea;
rangey3 = round(input1(1,2))-searcharea:round(input1(1,2))+searcharea;

if k == 1
    irangex3 = rangex3;
```

```matlab
    irangey3 = rangey3;
end

% New coordiantes matrix
N = zeros(size(H));
N(rangey3,rangex3) = 1;

% Find high saturation points
N(S <= filt4) = 0;
[yinput1, xinput1] = find(N == 1);
input1 = [xinput1 yinput1];
input1 = mean(input1);

%% Increase accuracy of input2

rangex4 = round(input2(1))-searcharea:round(input2(1))+searcharea;
rangey4 = round(input2(2))-searcharea:round(input2(2))+searcharea;

if k == 1
    irangex4 = rangex4;
    irangey4 = rangey4;
end

% New coordinates matrix
N = zeros(size(H));
N(rangey4,rangex4) = 1;

% Find high saturation points
N(S <= filt4) = 0;
[yinput2, xinput2] = find(N == 1);
input2 = [xinput2 yinput2];
input2 = mean(input2);

%% Plots
if plots == 1
figure(6)
hold off
imshow(RGB);
hold on
set(gcf,'Position',[0 500 780 500])
axis xy
axis on
axis([150,650,400,900])
rectangle('Position',[rangex3(1) rangey3(1) searcharea*2 searcharea*2])
rectangle('Position',[rangex4(1) rangey4(1) searcharea*2 searcharea*2])

plot(xf4,yf4,'c.')

plot(input1(1), input1(2),'w.')
plot(input2(1), input2(2),'w.')
plot([meanx1 meanx1+500],[meany1 meany1], 'r.-')

figure(7)
hold off
imshow(RGB)
hold on
set(gcf,'Position',[960 50 480 400])
axis xy
axis on
axis([irangex3(1)-plotarea irangex3(end)+plotarea irangey3(1)-plotarea
irangey3(end)+plotarea])

h1 = plot(xinput1, yinput1,'c.');
h2 = plot(input1(1), input1(2),'k.');

figure(8)
hold off
imshow(RGB);
hold on
set(gcf,'Position',[1440 50 480 400])
axis xy
axis on
axis([irangex4(1)-plotarea irangex4(end)+plotarea irangey4(1)-plotarea
irangey4(end)+plotarea])

h3 = plot(xinput2, yinput2,'c.');
h4 = plot(input2(1), input2(2),'k.');
end

%% Adjust coordinates for correct axis
input1scale = input1.*scalefactor + shift;
input2scale = input2.*scalefactor + shift;
end
```

D.11.f_image_processing_micro

```
function [output1,output2,output1scale,output2scale] = f_image_processing_micro(RGB, shift,
centershift, scalefactor, color, k)
%Analyze RGB microsope image data and output the location of the dots.

global irangex3 irangey3 irangex4 irangey4 dataset plots
% Search settings
filt1 = 0.35; %Value threshold of first filter
filt2 = 0.7; %saturation threshold of second filter
filt4 = 0.70; %saturation threshold of fourth filter

if dataset == '4'
    searcharea = 100; %size of search area for second filter
    plotarea = 150; %size of additional area around searcharea to plot
elseif dataset == '5'
    searcharea = 100; %size of search area for second filter
    plotarea = 150; %size of additional area around searcharea to plot
elseif dataset == '6'
    searcharea = 100; %size of search area for second filter
    plotarea = 150; %size of additional area around searcharea to plot
elseif dataset == '7'
    searcharea = 100; %size of search area for second filter
    plotarea = 150; %size of additional area around searcharea to plot
elseif dataset == '10'
    searcharea = 110; %size of search area for second filter
    plotarea = 150; %size of additional area around searcharea to plot
elseif dataset == '11'
    searcharea = 110; %size of search area for second filter
    plotarea = 150; %size of additional area around searcharea to plot
else
    searcharea = 200; %size of search area for second filter
    plotarea = 150; %size of additional area around searcharea to plot
end

HSV = rgb2hsv(RGB);                                      %Convert to hsv
GRY = rgb2gray(RGB);%Convert to grayscale
GRY = im2double(GRY);                                     %Convert to double

%% Analysis
H = HSV(:,:,1);
S = HSV(:,:,2);
V = HSV(:,:,3);

% Coordinate matrix
M = ones(size(H));

%% Remove pixels with low value
% Remove blue line at bottom
M(1,:) = 0;

M(V <= filt1) = 0;
[yf2, xf2] = find(M == 1);

%% Get pixels with high saturation
M(S <= filt2) = 0;
[yf3, xf3] = find(M == 1);

%% Get pixels that have a certain color
%remove pixels that are within 20/350 bounds of red hue (or 160/260 for blue, 80/160 for
green)

if color == 'r'

    M(H >= 10/360) = 0; %red
else
    M(H <= (160/360)) = 0; %blue
    M(H >= (260/360)) = 0;
end

[yf4, xf4] = find(M == 1);

% M(H < (60/360)) = 0; %green
% M(H > (180/360)) = 0;

%% Seperate output1 and output2 link pixels

% Average recognized point locations
meanx1 = mean(xf4);
meany1 = mean(yf4);

% Split point data between input and output link (xsplit)
xoutput1 = xf4(xf4 <= meanx1);
xoutput2 = xf4(xf4 >= meanx1);
youtput1 = yf4(xf4 <= meanx1);
youtput2 = yf4(xf4 >= meanx1);
```

```
%% Average of two point collections
% Combine x and y data
output1 = [xoutput1 youtput1];
output2 = [xoutput2 youtput2];

% Calculate center of points
output1 = mean(output1);
output2 = mean(output2);

%% Increase accuracy of output1

rangex3 = round(output1(1))-searcharea:round(output1(1))+searcharea;
rangey3 = round(output1(2))-searcharea:round(output1(2))+searcharea;

if k == 1
    irangex3 = rangex3;
    irangey3 = rangey3;
end

% New coordiantes matrix
N = zeros(size(H));
N(rangey3,rangex3) = 1;

% Find high saturation points
N(S <= filt4) = 0;
[youtput1, xoutput1] = find(N == 1);
output1 = [xoutput1 youtput1];
output1 = mean(output1);

%% Increase accuracy of output2

rangex4 = round(output2(1))-searcharea:round(output2(1))+searcharea;
rangey4 = round(output2(2))-searcharea:round(output2(2))+searcharea;

if k == 1
    irangex4 = rangex4;
    irangey4 = rangey4;
end

% New coordinates matrix
N = zeros(size(H));
N(rangey4,rangex4) = 1;

% Find high saturation points
N(S <= filt4) = 0;
[youtput2, xoutput2] = find(N == 1);
output2 = [xoutput2 youtput2];
output2 = mean(output2);

%% Plots
if plots == 1
figure(6)
hold off
imshow(RGB);
hold on
set(gcf,'Position',[0 500 780 500])
axis xy
axis on

plot(xf3,yf3,'g.')
plot(xf4,yf4,'c.')

rectangle('Position',[rangex3(1) rangey3(1) searcharea*2 searcharea*2])
rectangle('Position',[rangex4(1) rangey4(1) searcharea*2 searcharea*2])

plot(output1(1), output1(2),'w.')
plot(output2(1), output2(2),'w.')
plot([meanx1 meanx1],[meany1 meany1+500], 'r.-')

figure(7)
hold off
imshow(RGB)
hold on
set(gcf,'Position',[960 50 480 400])
axis xy
axis on
axis([irangex3(1)-plotarea irangex3(end)+plotarea irangey3(1)-plotarea
irangey3(end)+plotarea])

h1 = plot(xoutput1, youtput1,'c.');
h2 = plot(output1(1), output1(2),'k.');

figure(8)
hold off
imshow(RGB);
hold on
```

```
set(gcf,'Position',[1440 50 480 400])
axis xy
axis on
axis([irangex4(1)-plotarea irangex4(end)+plotarea irangey4(1)-plotarea
irangey4(end)+plotarea])

h3 = plot(xoutput2, youtput2,'c.');
h4 = plot(output2(1), output2(2),'k.');
end

%% Adjust coordinates for correct axis
output1scale = output1.*scalefactor + shift + centershift;
output2scale = output2.*scalefactor + shift + centershift;
end
```

D.12.f_RCMcalculatorgrid

```
function [xpoints, ypoints, gridrcm] = f_RCMcalculatorgrid(saveddata)
%Calculates the location of the RCM by using the data of the movement of
%four points on the final link.

% saveddata = saveddata(1:round(length(saveddata)/20):end,:);
gridsize = 500;

out1 = saveddata(:,1:2);
out2 = saveddata(:,3:4);
out3 = saveddata(:,5:6);
out4 = saveddata(:,7:8);

vec12 = out2-out1;
vec34 = out4-out3;
vec13 = out3-out1;
vec24 = out4-out2;

%% Calculate grid for every step
for i = 1:length(saveddata)
    %Create next grid
    for j = 1:gridsize-1
        point12(j,:) = out1(i,:) + j*vec12(i,:)/gridsize;
        point34(j,:) = out3(i,:) + j*vec34(i,:)/gridsize;
        point13(j,:) = out1(i,:) + j*vec13(i,:)/gridsize;
        point24(j,:) = out2(i,:) + j*vec24(i,:)/gridsize;
    end

    for k = 1:length(point12)
        for l = 1:length(point13)
            [xpoints(l,k,i),ypoints(l,k,i)] = f_lineintersect(point12(k,:), point34(k,:),
point13(l,:), point24(l,:));
        end
    end
end

%% Find RCM
for m = 1:size(saveddata,1)-1
    xdisp = xpoints(:,:,m+1)-xpoints(:,:,m);
    ydisp = ypoints(:,:,m+1)-ypoints(:,:,m);
    xdisp = reshape(xdisp,1,[]); %reshape to 1 row
    ydisp = reshape(ydisp,1,[]); %reshape to 1 row
    disp = [xdisp; ydisp];
    normdisp = vecnorm(disp);
    normdisp = reshape(normdisp,  gridsize-1, gridsize-1);
    normdisp(:,:,m) = normdisp;
    minvalue = min(min(normdisp(:,:,m)));
    [row, col] = find(normdisp(:,:,m)==minvalue);
    gridrcm(m,:) = [xpoints(row,col,m) ypoints(row,col,m)];
end
```

D.13.f_vrot2

```
function [angled] = f_vrot2(a,b)
%Calculates the angle between two vectors.
x1 = a(1);
y1 = a(2);
x2 = b(1);
y2 = b(2);
angled = atan2d(x1*y2-y1*x2,x1*x2+y1*y2);
end
```

D.14.f_mechanismkinematics

```
function [hingecoordinates, hingeangles, ilinkvectors, linkvectors ] =
f_mechanismkinematics(hingecoordinates,rotationangle)
%This is the kinematic model of the RCM mechanism. It calculates the new hinge
%coordinates as a function of the rotation angle.

ih12 = hingecoordinates(:,1); %hinge coordinates
ih13 = hingecoordinates(:,2);
ih24 = hingecoordinates(:,3);
ih26 = hingecoordinates(:,4);
ih34 = hingecoordinates(:,5);
ih45 = hingecoordinates(:,6);
ih56 = hingecoordinates(:,7);
ih57 = hingecoordinates(:,8);
ih68 = hingecoordinates(:,9);
ih78 = hingecoordinates(:,10);
ih88 = hingecoordinates(:,11);
ih66 = hingecoordinates(:,12);

th = rotationangle;
%rotation angle

%% Initial conditions
ilv1 = ih13 - ih12; %link vectors
ilv21 = ih12 - ih24;
ilv22 = ih24 - ih26;
ilv3 = ih13 - ih34;
ilv41 = ih34 - ih24;
ilv42 = ih45 - ih34;
ilv5 = ih45 - ih56;
ilv51 = ih45 - ih57;
ilv52 = ih57 - ih56;
ilv61 = ih56 - ih26;
ilv62 = ih68 - ih56;
ilv7 = ih78 - ih57;
ilv81 = ih78 - ih68;
ilv82 = ih88 - ih78;
ilv63 = ih66 - ih26;

ilinkvectors = [ilv1 ilv21 ilv22 ilv3 ilv41 ilv42 ilv5 ilv51 ilv52 ilv61 ilv62 ilv7 ilv81
ilv82 ilv63];

il13 = norm(ilv3); %link lengths
il141 = norm(ilv41);
il15 = norm(ilv5);
il161 = norm(ilv61);
il17 = norm(ilv7);
il181 = norm(ilv81);

%% Calculate h12 and h13
h12 = ih12;
h13 = ih13;

%% Calculate h24 and h26

R = [cosd(th) -sind(th) ;sind(th) cosd(th)]; %rotation matrix
h24 = h12 - R*ilv21;

h26 = h24 - R*ilv22;

%% Calculate h34 and h45

h34 = f_linkintersect(h13,h24,il13,il141,ih34);

lv41 = h34 - h24; %new link vector
th = f_vrot2(ilv41,lv41);
R = [cosd(th) -sind(th) ;sind(th) cosd(th)]; %rotation matrix
lv42 = R*ilv42; %new link vector
h45 = h34 + lv42;

%% Calculate h56 and h57

h56 = f_linkintersect(h26,h45,il161,il15,ih56);

lv5 = h45 - h56; %new link vector
th = f_vrot2(ilv5,lv5);
R = [cosd(th) -sind(th) ;sind(th) cosd(th)]; %rotation matrix
lv51 = R*ilv51; %new link vector
h57 = h45 - lv51;

%% Calculate h68

lv61 = h56 - h26; %new link vector
th = f_vrot2(ilv61,lv61);
R = [cosd(th) -sind(th) ;sind(th) cosd(th)]; %rotation matrix
```

```
lv62 = R*ilv62; %new link vector
h68 = h56 + lv62;

%% Calculate h78 and h88

h78 = f_linkintersect(h57,h68,ill7,ill81,ih78);

lv81 = h78 - h68; %new link vector
th = f_vrot2(ilv81,lv81);
R = [cosd(th) -sind(th) ;sind(th) cosd(th)]; %rotation matrix
lv82 = R*ilv82; %new link vector
h88 = h78 + lv82;

%% Calculate h66

lv61 = h56 - h26; %new link vector
th = f_vrot2(ilv61,lv61);
R = [cosd(th) -sind(th) ;sind(th) cosd(th)]; %rotation matrix
lv63 = R*ilv63; %new link vector
h66 = h26 + lv63;

%% Calculate new link vectors
lv1 = h13 - h12; %link vectors
lv21 = h12 - h24;
lv22 = h24 - h26;
lv3 = h13 - h34;
lv41 = h34 - h24;
lv42 = h45 - h34;
lv5 = h45 - h56;
lv51 = h45 - h57;
lv52 = h57 - h56;
lv61 = h56 - h26;
lv62 = h68 - h56;
lv7 = h78 - h57;
lv81 = h78 - h68;
lv82 = h88 - h78;
lv63 = h66 - h26;

linkvectors = [lv1 lv21 lv22 lv3 lv41 lv42 lv5 lv51 lv52 lv61 lv62 lv7 lv81 lv82 lv63];

ll1 = norm(lv1); %link lengths
ll21 = norm(lv21);
ll22 = norm(lv22);
ll3 = norm(lv3);
ll41 = norm(lv41);
ll42 = norm(lv42);
ll5 = norm(lv5);
ll51 = norm(lv51);
ll52 = norm(lv52);
ll61 = norm(lv61);
ll62 = norm(lv62);
ll7 = norm(lv7);
ll81 = norm(lv81);
ll82 = norm(lv82);
ll63 = norm(lv63);

%% Calculate hinge angles

ah12 = f_vrot2(lv1, lv21);
ah13 = f_vrot2(lv1, lv3);
ah24 = f_vrot2(lv21, lv41);
ah26 = f_vrot2(lv21, lv61);
ah34 = f_vrot2(lv3, lv41);
ah45 = f_vrot2(lv41, lv5);
ah56 = f_vrot2(lv5, lv61);
ah57 = f_vrot2(lv5, lv7);
ah68 = f_vrot2(lv61, lv81);
ah78 = f_vrot2(lv7, lv81);

%% Outputs
hingecoordinates = [h12 h13 h24 h26 h34 h45 h56 h57 h68 h78 h88 h66];
hingeangles = [ah12 ah13 ah24 ah26 ah34 ah45 ah56 ah57 ah68 ah78]; %save hinge angles
end
```

D.15.f_ICRflexure

```
function [motion, rotpoint] =
f_ICRflexure(prevrotpoint,prevhingerotation,hingerotation,hingelength)
%This function gives the parasitic motion along the symmetry line of the initial
%step. Inputs are the location of the rotation point of the previous step,
%the start rotation of the step and the end rotation, and the hinge length.

if prevhingerotation == 0
    arcradius = hingelength/(hingerotation/360*2*pi);
    arccenter = [arcradius hingelength/2];
    arcmiddlex = arcradius - arcradius*cosd(hingerotation/2); % middle of leaf spring is (0,0)
    arcmiddley = hingelength/2 - arcradius*sind(hingerotation/2);

    slope = (arccenter(2)-arcmiddley)/(arccenter(1)-arcmiddlex);
    height = arcmiddley-slope*arcmiddlex;

    rotax = -height/slope;
    rotay = slope*rotax+height;

    rotpoint = [rotax; rotay];
    %distance from previous rotation point to this one
    motion = norm(rotpoint - prevrotpoint);
else
    %initial rotation symmetry line
    arcradius = hingelength/(prevhingerotation/360*2*pi);
    arccenter = [arcradius hingelength/2];

    arcmiddlex = arcradius - arcradius*cosd(prevhingerotation/2);
    arcmiddley = hingelength/2 - arcradius*sind(prevhingerotation/2);

    islope = (arccenter(2)-arcmiddley)/(arccenter(1)-arcmiddlex);
    iheight = arcmiddley-islope*arcmiddlex;

    %new rotation symmetry line
    arcradius = hingelength/(hingerotation/360*2*pi);
    arccenter = [arcradius hingelength/2];

    arcmiddlex = arcradius - arcradius*cosd(hingerotation/2);
    arcmiddley = hingelength/2 - arcradius*sind(hingerotation/2);

    slope = (arccenter(2)-arcmiddley)/(arccenter(1)-arcmiddlex);
    height = arcmiddley-slope*arcmiddlex;

    %intersection of two symmetry lines
    rotax = (height - iheight)/(islope - slope);
    rotay = slope*rotax+height;

    rotpoint = [rotax; rotay];

    %distance from previous rotation point to this one
    motion = norm(rotpoint - prevrotpoint);

    %% Plot
    % figure
    % plot([0 0],[-hingelength/2 hingelength/2],'-o')
    % hold on
    % plot(arccenter(1),arccenter(2),'o')
    %
    % plot(arcmiddlex,arcmiddley,'o')
    % fplot(@(x) islope*x+iheight)
    % fplot(@(x) slope*x+height)
    % plot(rotpoint(1),rotpoint(2),'o')
    % plot(rotpoint(1),rotpoint(2),'o')

end
end
```

D.16.f_lineintersect

```
function [xintersect,yintersect] = f_lineintersect(point1, point2, point3, point4)
%Calculates the intersection point of two lines that are defined by
%point1-point4

% Line 1
line1 = point2-point1;
slope1 = line1(:,2)/line1(:,1);
height1 = point1(:,2)-slope1*point1(:,1);

% Line 2
line2 = point4-point3;
slope2 = line2(:,2)/line2(:,1);
height2 = point3(:,2)-slope2*point3(:,1);

% Intersect
xintersect = (height2-height1)/(slope1-slope2);
yintersect = slope1*xintersect+height1;
end
```

D.17.f_linkintersect

```
function [intersect] = f_linkintersect(loc1,loc2,link1,link2,prevloc)
%Finds the intersection point of two links which is closest to its previous
```

```
%location. input x; y
x1 = loc1(1);
y1 = loc1(2);
x2 = loc2(1);
y2 = loc2(2);

[xout,yout] = circcirc(x1,y1,link1,x2,y2,link2);
options = [xout; yout];
size = vecnorm(abs(options - [prevloc prevloc]))'; %find closest option
intersect = options(:,find(size == min(size)));
end
```

References

[1] C. Liu *et al.*, "In-situ STEM imaging of growth and phase change of individual CuAlX precipitates in Al alloy," *Sci. Rep.*, vol. 7, no. 1, p. 2184, Dec. 2017.

[2] D. B. Williams and C. B. Carter, *Transmission Electron Microscopy: A Textbook for Materials Science*, vol. 1–4. Boston, MA: Springer US, 2009.

[3] W. Zhou and Z. L. Wang, Eds., *Scanning Microscopy for Nanotechnology*. New York, NY: Springer New York, 2007.

[4] T. Slater, Y. Chen, G. Auton, N. Zaluzec, and S. Haigh, "X-Ray Absorption Correction for Quantitative Scanning Transmission Electron Microscopic Energy-Dispersive X-Ray Spectroscopy of Spherical Nanoparticles," *Microsc. Microanal.*, vol. 22, pp. 440–447, 2016.

[5] C. E. Lyman, J. I. Goldstein, D. B. Williams, D. W. Ackland, S. Von Harrach, and A. W. NICHOLLSt, "High-performance X-ray detection in a new analytical electron microscope," *J. Microsc.*, vol. 176, no. 2, pp. 85–98, 1994.

[6] T. J. A. Slater, P. H. C. Camargo, M. G. Burke, N. J. Zaluzec, and S. J. Haigh, "Understanding the limitations of the Super-X energy dispersive x-ray spectrometer as a function of specimen tilt angle for tomographic data acquisition in the S/TEM," *J. Phys. Conf. Ser.*, vol. 522, no. 1, p. 012025, Jun. 2014.

[7] R. M. Langford and A. K. Petford-Long, "Preparation of transmission electron microscopy cross-section specimens using focused ion beam milling," *J. Vac. Sci. Technol. A Vacuum, Surfaces, Film.*, vol. 19, no. 5, pp. 2186–2193, Sep. 2001.

[8] I. L. Delimont, S. P. Magleby, and L. L. Howell, "Evaluating Compliant Hinge Geometries for Origami-Inspired Mechanisms," *J. Mech. Robot.*, vol. 7, no. 1, p. 011009, Feb. 2015.

[9] J. O. Jacobsen, G. Chen, L. L. Howell, and S. P. Magleby, "Lamina Emergent Torsional (LET) Joint," *Mech. Mach. Theory*, vol. 44, pp. 2098–2109, 2009.

[10] G. Zong, X. Pei, J. Yu, and S. Bi, "Classification and type synthesis of 1-DOF remote center of motion mechanisms," *Mech. Mach. Theory*, vol. 43, pp. 1585–1595, 2008.

[11] E. Henderson, D. H. Nash, and W. M. Dempster, "On the experimental testing of fine Nitinol wires for medical devices," *J. Mech. Behav. Biomed. Mater.*, vol. 4, no. 2, pp. 261–268, 2011.

[12] J. Eaton-Evans, J. M. Dulieu-Barton, E. G. Little, and I. A. Brown, "Observations during mechanical testing of Nitinol," *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.*, vol. 222, no. 2, pp. 97–105, 2008.

[13] Y. Liu, "On the Detwinning Mechanism in Shape Memory Alloys," in *IUTAM Symposium on Mechanics of Martensitic Phase Transformation in Solids*, 2002, pp. 37–44.

[14] "JPE - Precision Point: Free engineering knowledge." [Online]. Available: https://www.janssenprecisionengineering.com/precisionpoint/. [Accessed: 12-Jan-2018].

[15] M. Verotti, "Analysis of the center of rotation in primitive flexures: Uniform cantilever beams with constant curvature," *MAMT*, vol. 97, pp. 29–50, 2016.

[16] Y.-Q. Yu, L. L. Howell, C. Lusk, Y. Yue, and M.-G. He, "Dynamic Modeling of Compliant Mechanisms Based on the Pseudo-Rigid-Body Model," *J. Mech. Des.*, vol. 127, no. 4, p. 760, Jul. 2005.