

## Department of Precision and Microsystems Engineering

### Design optimization for a kinematic coupling for use in vacuum with minimum particle generation

Jelle Kortman

Report no : 2023.006  
Coach : Ir. W.W.J.P. van de Sande  
Professor : Prof.dr.Ir. J.L. Herder  
Specialisation : Mechatronic System Design  
Type of report : MSc Thesis  
Date : 15 January 2023

# Design optimization for a kinematic coupling for use in vacuum with minimum particle generation

by

Jelle Kortman

to obtain the degree of Master of Science  
in Mechanical Engineering  
at the Delft University of Technology,  
to be defended publicly on Wednesday January 25, 2023 at 9:30 AM.

Student number:	5177103	
Thesis committee:	Prof. dr. ir. J. Herder,	TU Delft, Chair
	Ir. W. W. J. P. van de Sande,	TU Delft, supervisor
	Dr. Ir. J.G. Buijnsters	TU Delft
	Ir. J. van Dalen,	VDL ETG

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

# Preface

This thesis is the final part of the master Mechanical Engineering at the faculty of 3ME of the Technical University of Delft. Where I followed the track High-Tech Engineering with the specialization of Mechatronic System Design, which is part of the Precision and Microsystems Engineering department. This study was performed in cooperation with VDL ETG T&D located in Eindhoven, where the inspiration for the project came from. This study investigates how the design of a kinematic coupling affects particle generation. The first part of this thesis presents an overview of particle generation methods that are applicable to kinematic couplings. The latter part of the thesis is about translating these generation methods into design parameters and are optimized for low particle generation.

*Jelle Kortman  
Delft, January 2023*

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Goal of the study	4
1.2	Study layout	4
<b>2</b>	<b>Particle generation sources</b>	<b>4</b>
2.1	Wear mechanisms	5
2.2	Particle distribution model for low wear rate applications	6
2.3	Particle generation sources and guidelines	6
<b>3</b>	<b>Guideline one: Contact points and freedom of motion</b>	<b>6</b>
3.1	Kinematics	6
3.2	Self-locking	7
3.3	Freedom of motion	8
3.4	Statistical analysis on performance improvement	8
	Monte-Carlo analysis: Input • Monte-Carlo simulation: Results	
<b>4</b>	<b>Guideline two: Indenter shape and dimensions</b>	<b>11</b>
4.1	Indenter shape	11
4.2	Indenter dimensions	12
<b>5</b>	<b>Guideline three: Constraints positions</b>	<b>12</b>
5.1	Objective goals	12
5.2	Optimising repeatability	13
	Calculating objective function • Execution	
5.3	Checking for stability	13
	Contact model • Grasp model • Grasp quality measure	
<b>6</b>	<b>Case study</b>	<b>15</b>
6.1	Maximum nesting force and suspension setup	15
6.2	Pre-processing object	16
6.3	Optimising grasp repeatability	17
6.4	Checking for grasp stability	17
6.5	Indenter dimensions	17
6.6	Suspension setup	18
	Building blocks • Minimum required stroke • Concept example	
<b>7</b>	<b>Discussion</b>	<b>18</b>
7.1	Main findings	18
	Applications fields	
7.2	Limitations	19
7.3	Design recommendations	20
7.4	Experimental recommendations	20
<b>8</b>	<b>Conclusion</b>	<b>20</b>
	<b>References</b>	<b>20</b>
<b>A</b>	<b>Monte-Carlo simulation</b>	<b>23</b>
A.1	Input	23
A.2	Algorithm procedure V-groove/Kelvin-Clamp	25
A.3	Algorithm procedure constraint method	25
A.4	Matlab: MonteCarlo analysis	26
<b>B</b>	<b>Indenter shape</b>	<b>30</b>
B.1	Matlab Code: Stress distribution	32
B.2	Matlab Code: Maximum stress	33

<b>C</b>	<b>Wrenchhull</b>	<b>35</b>
C.1	Example of stability check for a 2D object with friction	35
C.2	Matlab Code: Wrench hull and objective measure - No friction	36
C.3	Matlab Code: Wrench hull and objective measure - With friction	40
<b>D</b>	<b>Casestudy suspension</b>	<b>46</b>
D.1	Suspension setup	46
D.2	Minimum required nesting force	47
D.3	Suspension results	47
D.4	Matlab code: Leaf spring dimensions	48
D.5	Matlab code: folded leaf spring dimensions	49
<b>E</b>	<b>Algorithm execution time optimisation</b>	<b>51</b>
E.1	Run-time optimization	51
E.2	Pointcloud orientation	52
E.3	Shift and rotate pointcloud	53
E.4	Main function to exclude datapoints and lower resolution	54
E.5	Function IterationPointCloudV2: Removes datapoints and lowers resolution	56
<b>F</b>	<b>Matlab: ICR optimisation</b>	<b>57</b>
F.1	Main script: ICR optimization	57
F.2	Function: 'ICR_Optimisation'	59
F.3	Function: Graspstability	62
F.4	Function: TorqueV2S1 to TorqueV2S6	63
<b>G</b>	<b>Matlab: Rotational Offset</b>	<b>79</b>
<b>H</b>	<b>Discussion: Example limitation wrong calculation of normal vector</b>	<b>81</b>
<b>I</b>	<b>Example constraint position optimization with different boundary conditions</b>	<b>82</b>
<b>J</b>	<b>Literature Review: Micro-positioning and fixating of an object in a vacuum environment with restrictive contamination constraints</b>	<b>84</b>

# Design optimization for a kinematic coupling for use in vacuum with minimum particle generation

Jelle Kortman

## Abstract

In the semiconductor industry, integrated circuits with nanometre scale structures are manufactured on silicon wafers. Particle contamination on the topside of the wafer can result in the defectivity of the entire die. In the ever-increasing efforts to create smaller structures onto a wafer particle contamination becomes more restrictive to prevent the defectivity of the die. In order to reduce the risk of particle contamination the wafer is placed in a vacuum environment. But further risks exist, during the grasping and positioning of the wafer new particles are generated and are able to transport to critical surfaces. This study investigates what guidelines need to be followed for the design of a kinematic coupling for use in a vacuum, whilst keeping particle generation as low as possible. The guidelines can be used to determine an optimal kinematic coupling design for any arbitrary object shape, size and mass. First of all, it is determined that abrasive wear, adhesive wear, contact stresses and outgassing are the predominant factors that have an effect on particle generation and influence the design of a kinematic coupling. The first guideline optimizes the constraint method of the kinematic coupling. Here, the guideline states that each constraint needs to be able to translate parallel to the surface of the object, doing so minimizes abrasive and adhesive wear. This can be achieved by suspending the contact points by using, for example, (folded) leaf springs. The second guideline encompasses the contact point shape and size. To equally distribute contact stresses, a spherical indenter shape should be chosen. Here, the radius of the indenter is chosen to be as small as possible to prevent adhesive wear. The third and last guideline optimizes the position of the contact points placed on the object. These positions influence the positioning repeatability, an algorithm is introduced that can calculate the optimal position for the contact points to maximize this repeatability. Also, the stability of the grasp is determined in terms of maximum allowable external disturbance forces.

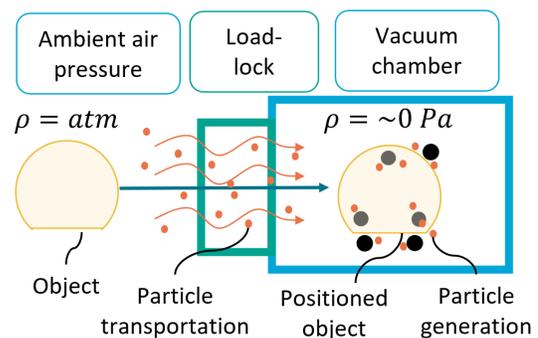
## Keywords

High-precision positioning – Kinematic coupling – Particle generation – Vacuum – Compliant mechanisms – Object grasping

## 1. Introduction

In the semiconductor industry, nanometre-scale structures are manufactured on silicon wafers. It is generally acknowledged that particle contamination on the top side of the wafer has a major influence on the defectivity of the die and is therefore the largest contributor to yield loss in the manufacturing process of integrated circuits [1, 2, 3]. In the ever-increasing efforts to create smaller structures on a wafer, the number and size of particles that can be present become more restrictive [4]. The manufacturing process of the integrated circuits is performed in a vacuum chamber, which is helpful with respect to particle contamination. One of the possible contamination sources is during the object handling of the wafer. Here, the wafer is picked up, transferred through a load lock from ambient air pressure to the vacuum chamber and at last is positioned and fixated. During this procedure, particles are generated at the locations where the contact points of the wafer gripper are applied to the wafer [3]. One of the other important contamination sources is the introduction of new particles during the transport of the wafer through the load-lock. Here, particles are free to move from the outer environment and enter the vacuum environment [5]. Second,

pumping down the load lock from ambient air pressure to a vacuum will introduce newly generated particles [6].



**Figure 1.** Particle contamination due to transportation from ambient air and generation at the points that make contact with the object.

Within the vacuum environment, particles with very small mass are able to move in every direction, even in directions opposite to the gravitational force [3, 7]. Therefore, the particles are able to transport itself to every reachable volume.

Which results in particles generated in non-critical volumes can cause performance-limiting risks if they can travel to critical volumes. Therefore, any particle that has an open path to reach a critical surface should be prevented.

From the industry, there is an increased interest in removing the need for a load-lock during the swapping procedure of objects. Excluding the load-lock will both reduce down-time and particle contamination since no longer a pump-down or opening of the valves is required [8]. What remains is the particle generation during the fixating and positioning of an object. In current practices, a well-known example for high-precision positioning and fixating objects with an arbitrary shape, size and mass is a V-groove of Kelvin-Clamp. This is a compact exact constrained solution that is able to reach precision in the  $1\ \mu\text{m}$  range [9]. The working principle of a V-groove is directly correlated with the disadvantage of this solution. A V-groove resorts to the solution where the pins connected to the object slide over the V-groove's surface into the position with minimal potential energy. Sliding creates friction between the two bodies and therefore will create a large amount of particles [7]. In literature, different kind of kinematic coupling strategies are described, but all resort to the same working principle that the object is pushed and slides over one or multiple surfaces during positioning [9, 10].

In this study guidelines are introduced for the design of a kinematic coupling for use in a vacuum with high positioning repeatability and minimum particle generation. The guidelines and design constraints can be used for a fast approach to design an optimal kinematic coupling for any object's shape, size and mass. First, multiple sources are identified that have a relationship between the inherent working principle of a kinematic coupling and particle generation. The present wear mechanisms are a combination of factors such as abrasive-, adhesive wear and contact stresses. For each wear mechanism, guidelines are developed and optimized to minimize particle generation.

Added benefits of the guidelines introduced in this study include the ability to minimize the correlation between particle generation and gravitational forces. In other words, the mass of the object is no longer a significantly increasing factor in particle generation. This is contrary to a commonly used kinematic coupling like a V-groove, here particle generation increases significantly for higher object masses.

The ability of the kinematic coupling to obtain high repeatability and create minimum particles are correlated with each other. The position of the contact points influences the repeatability. Duenner [10] mentioned that maximizing the restoring moments about the instantaneous center of rotation maximizes the repeatability. An algorithm is introduced that can determine the best position of the constraints for an arbitrary shape and size to restore positioning errors for translation

in  $X$ - and  $Y$ -axis and rotation about the  $Z$ -axis. The stability of the grasp is checked using a second boundary condition function. Here, the boundary function can calculate the maximum allowable magnitude and direction of an external force.

### 1.1 Goal of the study

The goal of this study is to determine design guidelines for the design of a kinematic coupling with high positioning repeatability and low particle generation in a vacuum. The goal includes optimizing the position of the constraints, shape and size of contact points of the kinematic coupling.

### 1.2 Study layout

This study contains eight chapters. First, a theoretical analysis is performed concerning kinematic couplings and particle generation sources. Then, the obtained guidelines are applied for obtaining an optimized kinematic coupling. In chapter 3 the required kinematics are described, including a statistical analysis that provides the expected performance increase. In chapter 4, the optimal indenter shape and dimensions are determined. In chapter 5, the position of the constraints is optimized with respect to repeatability and particle generation using an optimization algorithm. A case study is performed in chapter 7, in order to provide an example and summary of the findings of this study. At last, a discussion and conclusion are given in chapters 7 and 8 respectively.

## 2. Particle generation sources

Understanding the process how particles are generated and what link can be made with respect to kinematic couplings is essential for developing guidelines and optimizing its design. Particle contamination in a vacuum environment can be found in many different forms, but can generally be divided into two categories, molecular and particle contamination. Various molecular and particle sources are visualized in figure 2. Molecular sources can be any organic, ions, molecules, non-volatile residue or inorganic surface contaminants. Particle contamination consists of fibers or any metal particle [7].

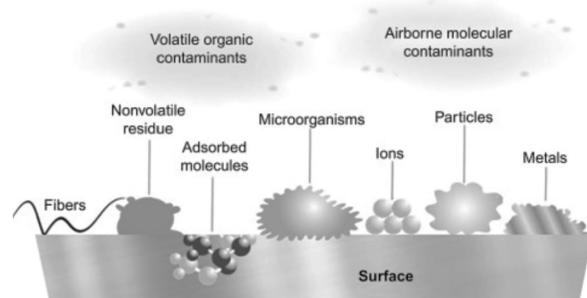


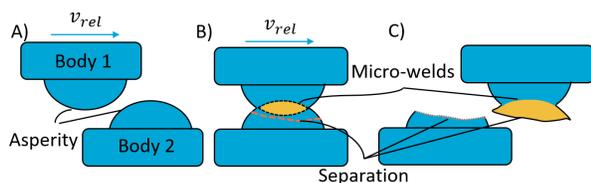
Figure 2. Types of contamination [11]

## 2.1 Wear mechanisms

For the design of a kinematic coupling, particle contamination in the form of material particles originating from the two bodies that come in contact is most important. Due to the interaction between the kinematic coupling and the object, friction is present at the contact points. The friction between two surfaces can result in wear of the contacting surfaces, which results in a large amount of newly introduced particles into the vacuum system [7]. The relationship between particle generation and wear volume can be divided into multiple categories, the ones that are applicable to kinematic couplings are the following wear mechanisms and boundary conditions:

### Adhesive wear:

Adhesive wear refers to a sticking effect of asperities of two surfaces. In reality, the contact pressure is carried by a contact surface consisting of asperity contacts. At asperity level, the contact pressure exceeds the elastic limit of the material. This results in the welding of the two asperities to each other. This phenomenon can not be noticed at a larger macroscopic scale, but is only observed at asperity level contacts [12]. At the instance when the surface starts to slide, the welded asperities will break [13].



**Figure 3.** When small peaks of two surfaces (asperities) come in contact, locally these peaks will weld together eventually leading to separation by any further motion.

Parameters that affect the wear volume for adhesive wear can be expressed by the Archard-Holmes equation (equation 1). Here,  $k$  is an experimentally determined wear rate factor. The  $k$  factor is dependent on material properties like roughness, lubricant application and geometry [12]. What the Archard-Holmes equation does not directly state is the influence of contact surface area. A larger contact surface area between two bodies results in an increase of adhesive wear [5, 14].

$$V = \frac{kFs}{\sigma} \quad (1)$$

Where:

- $V$  = Wear volume
- $k$  = Wear rate, differs for abrasive and adhesive wear
- $F$  = Normal load
- $s$  = Sliding distance
- $\sigma$  = Hardness of softer material

### Abrasive wear:

Abrasive wear is the tearing of the material when friction is present. Here, the peaks of the surface or asperities can penetrate into the softer material and tear material away from the softer surface [12].

Parameters that affect the wear volume for abrasive wear can be expressed by the Archard equation, which is almost identical to the Archard-Holmes equation (equation 1). The difference lays in a different experimentally determined  $k$  factor.

### Local contact pressure:

In an interview with a contamination expert K. van den Broek [5], a boundary condition was discussed that stated that the maximum Hertz contact pressure should not exceed 80% of the lowest yield strength of the two contacting materials. Exceeding this value will drastically increase particle generation.

### Outgassing:

Outgassing is present when a chamber is pumped down to vacuum pressures, during this phase outgassing can be divided into two categories [15]:

1. Outgassing from materials: Molecules of the material itself diffuse through the volume of the material and enter its surface, eventually desorbing from it.
2. Previously adsorbed molecules: Molecules that have most likely entered during the venting of the system are absorbed by the surface of the material. In a vacuum, these molecules desorb again from the surface.

The outgassing rate can be approximated with formula 2 [16]. Outgassing rates are high for porous materials like rubber or silicone, kinematic couplings that require these types of materials for their mechanical working principle are therefore not suitable for use in particle-critical applications. Second, outgassing can be minimized by reducing the object's surface area.

$$\dot{Q} = \sum \frac{\alpha_{1h} A}{\left(\frac{t}{1h}\right)^\alpha} \quad (2)$$

Where:

- $\dot{Q}$  = Outgassing flow rate
- $\alpha_{1h}$  = Outgassing rate at 1 hour of the material
- $\alpha$  = Decay constant of the material
- $A$  = Surface
- $t$  = Time

Friction between two surfaces can also introduce outgassing, this type of friction induced outgassing is also referred to as mechanically affected surfaces (MSO) [17, 18]. Repa et al. [17] describe the parameters that determine the outgassing rates due to MSO, he states that friction speed, normal load, and the concentration of earlier dissolved gasses in the materials have an influence.

### 2.2 Particle distribution model for low wear rate applications

The Archard equation (equation 1) is not specifically developed for low wear-rate applications where particle diameters in the micron range are considered. Therefore, it can not be said with full certainty that the Archard equation is valid for predicting low-wear rates.

VDL ETG [11] has performed an experiment to verify a particle distribution model for particle diameters ranging from the results are displayed in figure 4. In this experiment, the occurrence of a particle with a particular diameter is measured in three situations. The experiment is performed for a baseline measurement with a normal force equal to and a sliding distance equal to. These results are compared to a test setup where normal force and sliding distance were doubled. The results of the experiment can be fitted by equation 3.

Figure 4 shows the results of the experiment, the obtained data can be fitted with equation 3. Using the results of the performed experiment, it can be concluded that the linear relationship of normal force and sliding distance versus wear volume holds for low wear and small particle diameters situations. Therefore, the relationship with sliding distance and applied normal force as reasoned by the Archard equation (equation 1) also holds for very low wear rates.

### 2.3 Particle generation sources and guidelines

In table 5 an overview is given of the in this chapter presented particle generation sources, and the reduction methods that follow to minimize particle generation.

Contamination source	Parameters	Reduction methods
<b>Abrasive wear</b>	$V = \frac{k_{abr} F_N}{\sigma_0} x$	1) Minimize normal force 2) Minimize sliding distance
<b>Adhesive wear</b>	$V = \frac{k_{adh} F_N}{\sigma_0} x$	1) Minimize normal force 2) Minimize sliding distance 3) Minimize contact surface area
<b>Contact stresses</b>	$\sigma < 0.8\sigma_{yield}$	1) Peak stresses at contact points should not exceed $0.8\sigma_{yield}$
<b>Outgassing</b>	$\dot{Q} = \sum \left( \frac{\alpha_{1h} A}{1h} \right)^\alpha$	1) Porous materials like silicone or rubber have high outgassing rates 2) Minimize friction speed 3) Minimize sliding distance 4) Minimize surface area

Figure 5. Overview particle generation sources, parameters and guidelines.

## 3. Guideline one: Contact points and freedom of motion

Well-known kinematic couplings, like a V-groove or Kelvin Clamp, position an object using a ball that slides into a position that has minimal potential energy. During the positioning, the object is not yet fully constrained and therefore able to translate into its nested position, where it will be exactly constrained. The first guideline states that abrasive and adhesive wear can be respectively eliminated or minimized by removing sliding between the contact points and the object. This section describes the kinematics that is preferred to establish a kinematic coupling where the positioning is performed without sliding of contact points over a surface.

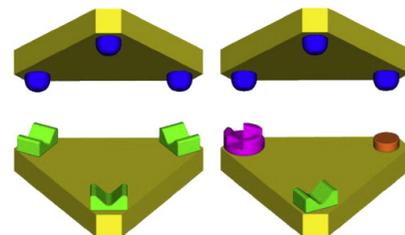


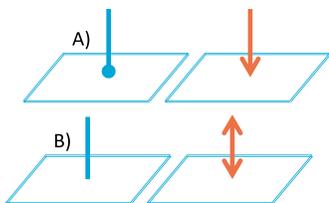
Figure 6. Commonly used kinematic coupling A) V-groove B) Kelvin Clamp [9].

### 3.1 Kinematics

A set of contact points that can fully constrain an object is also referred to as a grasp of the object. Nguyen [19] describes that a grasp can be in force- or form-closure. A force-closure grasp ability to maintain the grasp is limited by the magnitude of the forces that the contact points can apply to the object. In the situation where the magnitude of the applied forces of the

contact points are independent of the ability to hold the grasp, then form closure is achieved. A form closure grasp could also be seen as a grasp where only frictionless contacts can be used. A grasp using contact points with friction enables it to use the tangential friction forces to constrain an object, this enables the grasp to use fewer contact points.

Sliding between two surfaces is present when the tangential force exciting on the surface exceeds the static friction force [20]. Eliminating tangential forces at the contact points reduces the risk of slip between two surfaces to occur. Therefore, it is preferred to create a grasp that is form-closure, eliminating the need for tangential friction forces to maintain the grasp. Reuleaux [21] proved that for a 2D and 3D grasp respectively, a minimum of four and seven contact points are needed for form-closure. What can be noticed is that for both 2D and 3D form-closure grasps, there is an extra constraint needed than the number of degrees of freedom. This is due to the assumption that the constraint is modeled as a point contact, which can dislodge from the object. Therefore, it can only provide a reaction force in one direction (figure 7).

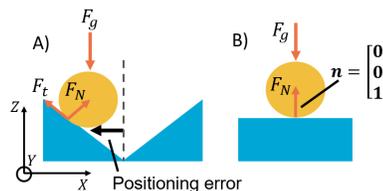


**Figure 7.** Contact models A) Point contact that is able to dislodge, can provide a reaction force in one direction. B) Fixed contact point model that is able to provide a reaction force in two directions.

It is assumed that the object is placed on the kinematic coupling by lowering the object in the Z-axis direction and has a positioning error in x and y-axis and rotational error around the z-axis. The object is first placed and released onto the kinematic coupling, thereafter the object’s positioning error is restored. When the object is released onto the kinematic coupling it should be able to constrain translation in z and rotation about X and Y axis ( $\delta_Z, r_x$  and  $r_y$ ). To obtain a set of contact points that are able to constrain the previously mentioned translation and rotations, a minimum of three contact points are required to constrain translation in z and rotation about X and Y axis ( $\delta_Z, r_x$  and  $r_y$ ) (figure 9A). The contact point is modeled as a point that is able to dislodge from the object. Therefore, motion in the positive Z-axis is still possible. To restrict motion in the positive Z-axis it is assumed that the gravitational force originating from the mass of the object delivers enough force to constrain any motion in the positive z-axis.

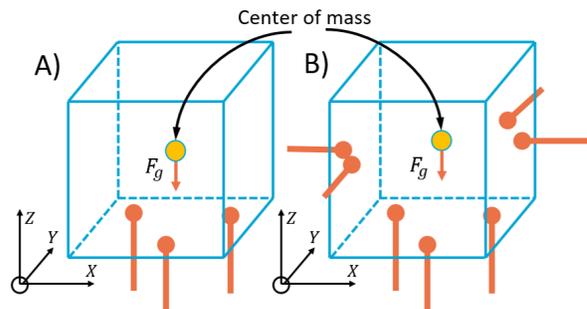
For the three contact points constraining  $\delta_Z, r_x$  and  $r_y$  a boundary condition has been set. For a V-groove or Kelvin clamp, the object is positioned by sliding of the object, which

is possible due to the sloped surfaces where the pins of the object makes contact. To eliminate this unwanted behavior, the three contact points constraining  $\delta_Z, r_x$  and  $r_y$  need to deliver their reaction force  $F_N$  in the positive Z-axis ( $\mathbf{n} = [0 \ 0 \ 1]$ ). Doing so, no reaction forces tangential to the surface are created that are able to actuate a sliding motion of the object (figure 8).



**Figure 8.** A) The pin of the object will slide into correct position. B) The direction of the reaction force is in the opposite direction of the forces exerted by the pin, no sliding between the surface and pin is possible.

The kinematic coupling needs to correct the planer positioning error in the XY-plane and angular positioning error about the Z-axis. The three contact points mentioned above (constraining  $\delta_Z, r_x$  and  $r_y$ ) do not contribute to the set of restoring forces required to nest the object and eliminate the positioning error. This is because the before mentioned three contact points only act in the positive Z axis ( $\mathbf{n} = [0 \ 0 \ 1]$ ). A further minimum of four contact points are required to restore the positioning error and constrain the remaining free translations X and Y axis and rotation about Z (figure 9B).



**Figure 9.** A) Three contact points and a body force  $F_g$  constraining  $\delta_Z, r_x$  and  $r_y$ . B) seven contact points and a body force  $F_g$  which together obtain a form-closure grasp

Overall, seven contact points and a body force originating from the mass of the object are required to fully constrain and restore the positioning error of a 3D object. This results in a total of eight body forces, including the body force originating from the gravitational force in the negative Z direction.

### 3.2 Self-locking

While positioning an object not all contact points make immediate contact, rather each contact point is established in chronological order. In the case that the contact points are

rigid, the object needs to be able to slide over these rigid contact points. This not only introduces abrasive and adhesive wear, but second, a failure mode called self-locking can appear. Self-locking is apparent when the tangential force can not exceed the static friction force at a contact point, therefore preventing sliding over the contact points. Or explained by:

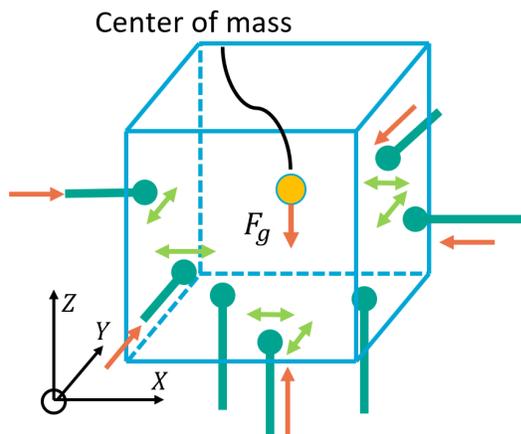
$$F_T < \mu_s F_N \quad (4)$$

In an ultra-high vacuum, the static friction coefficient drastically increases in an ultra-high vacuum. For example, the static friction coefficient for a common material interaction like aluminum-aluminum increases from 0.8 to 2.2 [22]. This makes this failure mode even more predominant.

### 3.3 Freedom of motion

The first guideline aims to minimize abrasive and adhesive wear by eliminating sliding between the contact points and the object. Removing the need for sliding to nest an object would also discard the self-locking failure mode. To fully constrain the object and nest the object without sliding, each contact point only needs to provide a reaction force normal to the surface. Tangential forces located at the contact point should be prevented since these do not contribute to the nesting or constraining and can introduce sliding.

Figure 10 shows the preferred freedom of motion for each contact point. Here, each contact point should be rigid normal to the object's surface and should be able to translate parallel to the object's surface. By allowing the contact point to translate parallel to the object's surface the sliding distance can be eliminated. In chapter 6 an example of a suspension setup is given that is able to obtain the preferred freedom of motion for each of the contact points.



**Figure 10.** Overview of applied contact points on an object to create a form-closure grasp. Each contact point constrains in one direction and is able to translate parallel to the surface. The gravitational force restricts motion in the positive Z-axis.

### 3.4 Statistical analysis on performance improvement

A statistical analysis has been performed to quantify the expected performance improvement of the proposed guideline to minimize abrasive and adhesive wear by eliminating sliding. A Monte-Carlo simulation approach is used to setup the statistical analysis. A Monte-Carlo is constructed by first setting up an objective function. The used parameters in this objective function are defined by a probability density function and are calculated using a standard deviation. Finally, the objective function is calculated for a large number of input values, and the occurrence and value for the inputs are based on the probability distribution [23].

The objective function is based on equation 1 in order to quantify the expected particle generation due to abrasive or adhesive wear. Here, it is known that abrasive and adhesive wear is dependent on material properties, sliding distance and normal force. In this study, the effect of material properties on particle generation is not considered, since these are independent on the kinematic coupling design. Only the variables sliding distance and normal force are used in the objective function, which is given in equation 5.

$$Obj = \sum_{i=1}^N \sum_{j=1}^K s(i, j) F_n(i, j) \quad (5)$$

Where:

$Obj$  = Objective

$s$  = Sliding distance of contact point  $j$  for positioning error corresponding to iteration  $i$

$F_n$  = Normal force excited by contact point  $j$  for the  $i_{th}$  iteration

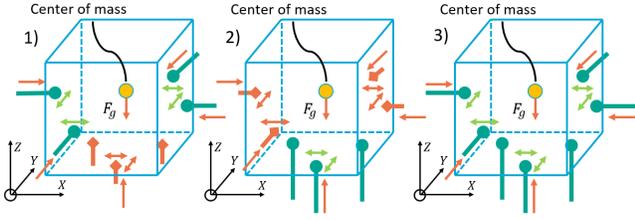
$N$  = Total number of iterations

$K$  = Total number of contact points

#### 3.4.1 Monte-Carlo analysis: Input

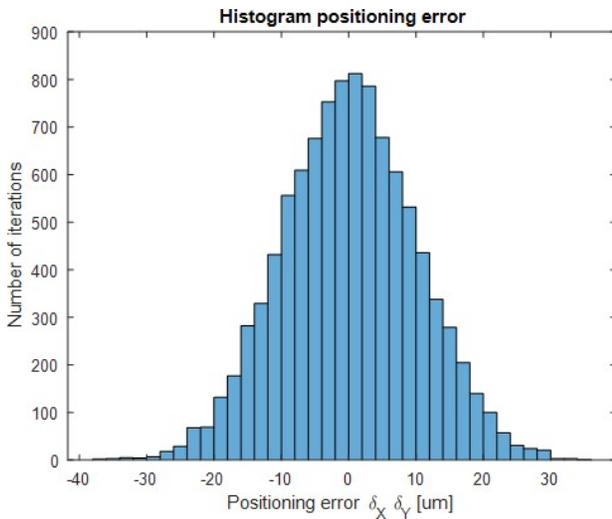
In appendix A, a comprehensive explanation of how the algorithm determines all input variables and calculates the sliding distance, normal force and the number of contact points for each iteration is described. A score for four situations has been calculated. First, a reference score for a well-known kinematic coupling like a V-groove and Kelvin Clamp (figure 6) is calculated. Second, three derivations have been made from the constraint method described in chapter 3.1 and visualized in figure 11.

1. Contact points constraining  $\delta_z$ ,  $r_y$  and  $r_x$  are rigid, other contact points are free to translate parallel to the object's surface.
2. The configuration is vice versa, contact points constraining  $\delta_x$ ,  $\delta_y$  and  $r_z$  are rigid, other contact points are free to translate parallel to the object's surface.
3. Contact point setup according to the guideline: All contact points can translate parallel to the surface, thus eliminating all sliding when positioning the object.

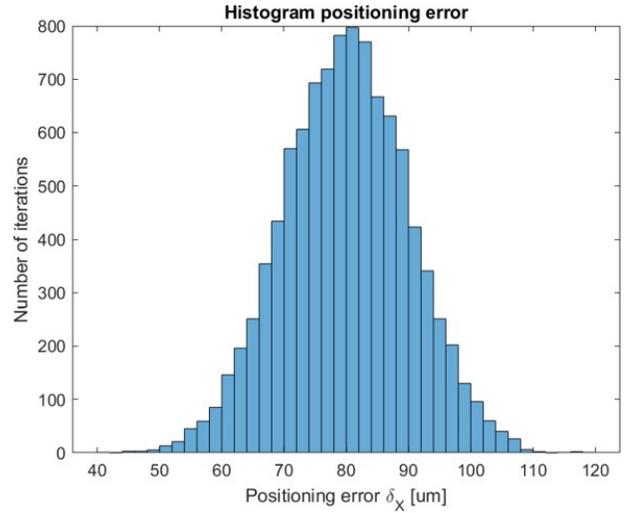


**Figure 11.** 1) Contact points constraining  $\delta_z$ ,  $r_y$  and  $r_x$  are rigid, remaining can translate parallel to the object's surface 2) Contact points constraining  $\delta_x$ ,  $\delta_y$  and  $r_z$  are rigid, remaining can translate parallel to the object's surface 3) Constraint method accordingly to the introduced guideline, all contact points are free to translate parallel to the object's surface.

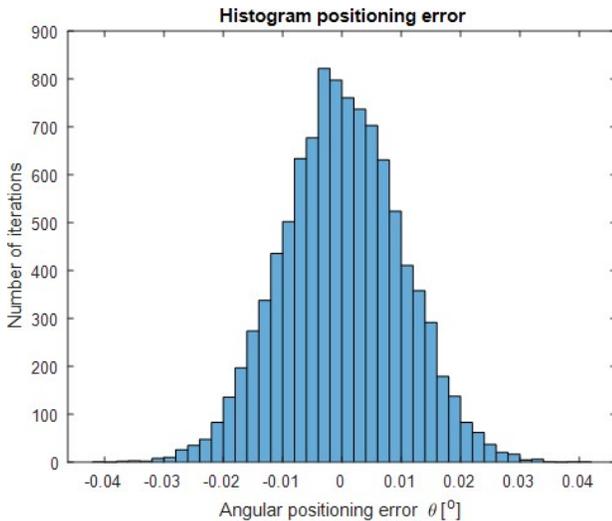
For this simulation, an assumption has been made for the translational and angular positioning error. For reference, the specification of a 3DOF SCARA-robot with a range of 300 mm is used [24]. A standard deviation of  $\sigma_r = 10 \mu\text{m}$  is used as a translational positioning error in the  $XY$ -plane, and an angular positioning error about the  $Z$ -axis of  $\sigma_a = 0.007^\circ$ . Second, for the three constraint method derivations shown in figure 11 it should be prevented that the object make contact with the contact points located at the side of the object. An extra translational and rotational offset is used to obtain a 99,7% ( $3\sigma$ ) successful placements. At last, the probability density function is populated with a total of  $N = 10E4$  iterations (figure 12-15).



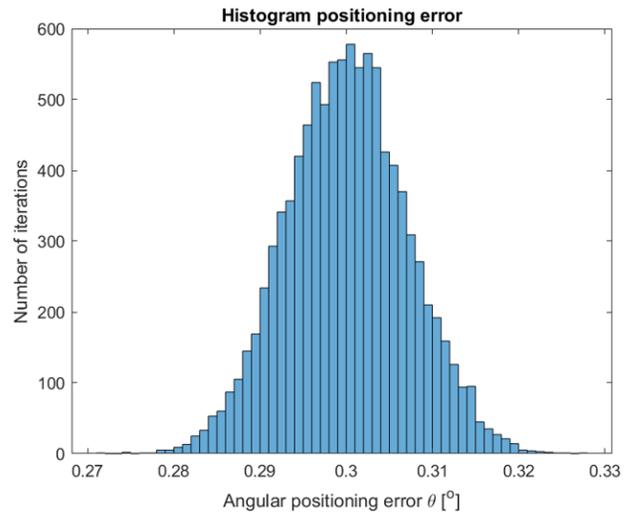
**Figure 12.** Histogram positioning error in  $XY$ -plane for a V-groove or Kelvin clamp.  $\sigma_{xy} = 10 \mu\text{m}$



**Figure 14.** Histogram positioning error in  $X$ -axis for suspended constraint methods.  $\sigma_{xy} = 10 \mu\text{m}$  and offset  $\delta_M = 80 \mu\text{m}$ . Positioning error in  $Y$ -axis is equal to that in figure 39



**Figure 13.** Histogram rotational positioning error for a V-groove or Kelvin clamp.  $\sigma_{rz} = 0.007^\circ$



**Figure 15.** Histogram rotational positioning error for suspended constraints method.  $\sigma_{rz} = 0.007^\circ$  and offset of  $\theta_M = 0.3^\circ$

The algorithm needs to determine what contact points engage with the object, and what normal force and sliding distance is present during positioning at these contact points. The approach to determining these variables is different for each studied constraint method. First, for a 'V-groove or Kelvin clamp' the following approach is used:

**Contact points  $K$ :** For the nested position the V-groove or Kelvin clamp has six contact points. When the object is not in the nested position it has three contact points.

**Sliding distance  $s$ :** Sliding distance is equal to the summation of positioning error in X and Y-axis.

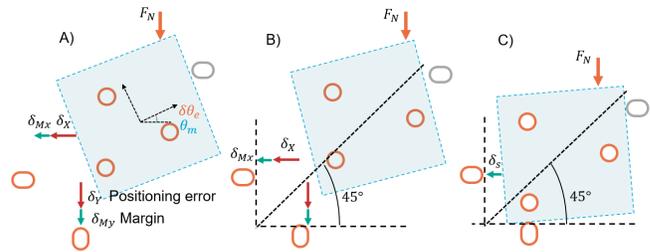
**Normal force  $F_n$ :** The normal force is determined by the force acting in the vertical axis, in this situation it is determined by the mass of the object. When the object is not in its nested position the mass of the object is divided over three contact points. Therefore, the normal force is equal to the mass of the object divided by three.

For the three derivations of constraint methods where different contact points can translate parallel to the object's surface (figure 11), the following approach is used:

**Contact points  $K$  and sliding distance  $s$ :** The object is placed onto three contact points, these make contact during the entire positioning procedure. Furthermore, four contact points are located at the side of the object. One of these contact points delivers the nesting force, it is assumed that for this contact point no sliding is present between the object.

The behavior of the object during positioning is determined using the following procedure:

1. The object is positioned with a translational and rotational positioning error and margin (figure 16A).
2. During positioning, the object can obtain an additional one or two contact points besides the already present nesting force contact. Predicting how the object will rotate about which ICR and how it will translate when it is being pushed by the nesting force is complex to model. Therefore, a simplification is used that states that the object will always translate in a  $45^\circ$  angle (figure 16B).
3. Eventually the object make contact with a extra contact point, now the direction of the object changes to the still not fully constrained direction (figure 16B-C). During this last movement, the constraint located at the side is in contact during the entire last positioning part. If this contact point is set to be rigid, then the remaining distance to the nested position is seen as sliding distance.
4. During the positioning the object will rotate, this creates an extra movement for all contact points.

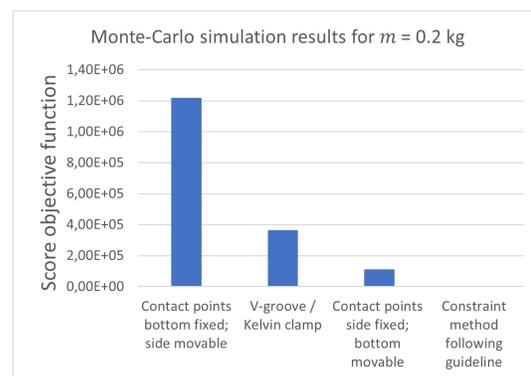


**Figure 16.** Schematic top view of object positioning procedure to determine sliding distance dependent on positioning error and margin

**Normal force  $F_n$ :** The normal force on the three contacts points located at the bottom of the object is equal to the gravitational force divided by three. Second, the normal force on all contact located at the side of the object is based on the minimal required nesting force. The nesting force is dependent on the minimum required actuation force to obtain the minimum required stroke for the contact point's suspension. The stroke is dependent on the stiffness of the suspension, a lower stiffness results in a decrease in the required actuation force. Second, the required stroke is dependent on the positioning error and extra offset margin. This all creates a feedback loop where the goal is to obtain a as low as possible stiffness, this feedback loop is elaborated upon in more detail in appendix D.2.

### 3.4.2 Monte-Carlo simulation: Results

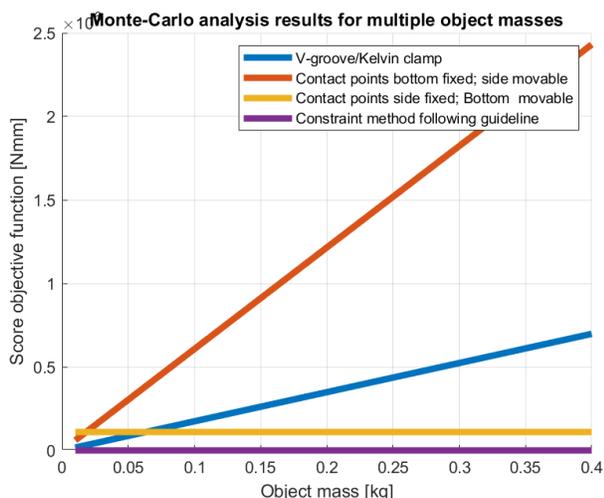
A higher score for the objection function (equation 5) indicates higher abrasive and adhesive wear. Figure 17 shows the result of the Monte-Carlo simulation for an object with a mass equal to 200 grams. This latter addition is important since for some of the constraint methods particle generation is highly dependent on the mass of the object. For the V-groove/Kelvin clamp and constraint option one in figure 11, the contact forces where sliding is present are dependent on the mass. Therefore, these options scored worst and will create the most amount of particles during positioning. The lowest scoring option is the constraint method according to the guideline and shows a significant decrease in particle generation since sliding between the contact points is prevented.



**Figure 17.** Objective score Monte-Carlo analysis for object mass 0.2 kg

The dependency on the mass of the object and abrasive or adhesive wear can be further emphasized using the results shown in figure 18. Here, the Monte-Carlo simulation is performed multiple times for a range of object masses. With an increasing mass of the object, the particle generation due to abrasive and adhesive wear increases for the V-groove/Kelvin Clamp and the constraint derivation where the contact points located at the bottom of the object are fixed. Second, particle generation stays constant for the constraint method where contact points located at the bottom of the object can translate parallel the surface of the object.

It can be concluded that the constraint method that follows the guideline, results in the lowest amount of particles. Especially for object's with high masses the decrease in particle generation is significant. But for low object masses the benefits of applying the guideline could be not significant enough against the added complexity by following the guideline.



**Figure 18.** Objective score Monte-Carlo analysis for a range of object masses

## 4. Guideline two: Indenter shape and dimensions

In this chapter, the introduced particle reduction methods are incorporated for the design of the contact points that will constrain the object, these pins are referred to as indenters. In chapter 2.3 it is described that the local contact stresses and contact surface area has an influence on particle generation. The optimal indenter shape and dimensions are determined to minimize particle generation.

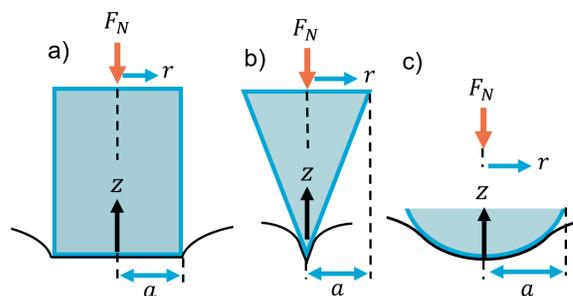
### 4.1 Indenter shape

The indenter shape influences how the stresses are transferred and distributed between the two bodies. The stress distribution and maximum stresses can be calculated using the Method of Dimensionality Reduction (MDR) and Hertz contact theory [25, 26]. In publicly available literature, a direct relationship between particle generation and Hertzian contact stress is not

given. Quantifying how large the decrease in wear rates will be based on design parameters is not possible. VDL ETG gave the following assumptions are made regarding minimizing particle generation and indenter shapes and dimensions:

1. The maximum Hertzian contact stress should not exceed 80% of the lowest yield stress of the contacting material [5].
2. Increasing surface area will increase adhesive forces and so adhesive wear. The surface area that is in contact between the indenter and object should be minimized [5, 14].

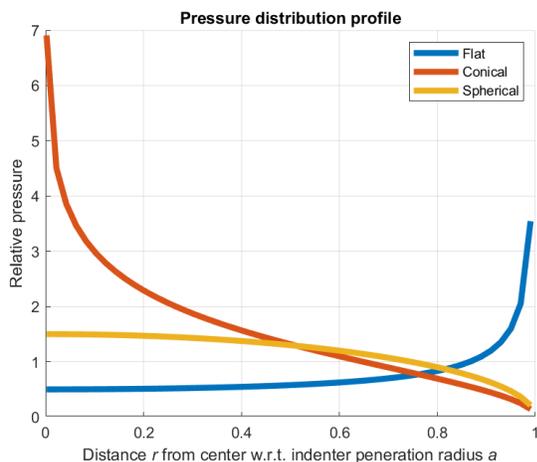
Three indenter shapes, flat, conical and spherical are considered and compared against each other (figure 19). For each shape, the pressure distribution is determined and compared. The pressure distribution is calculated by dividing the normal pressure ( $p_n = -\sigma_{zz}$ ) by the average pressure ( $p_0$ ). The results are plotted in figure 20. The derivation of these formulas is described in appendix B.



**Figure 19.** Indenter profiles. a) Flat b) Conical c) Spherical

The pressure distributions for a conical or flat indenter shape result in high peak stresses at respectively the indenter center or the indenter edge. Due to these high peak stresses, it is hard to comply with the requirement that maximum Hertzian contact stresses can not exceed 80% of the lowest yield stress of the contacting material. Secondly, by increasing the radius of the flat and spherical indenter or decreasing the cone angle, the surface area is increased. This will lead to a decrease in peak stresses. An adverse effect is that increasing the surface area will increase the adhesion forces, and thereby increasing adhesive wear. This is described by assumption two which prescribes contacting surface area should be kept as small as possible [5, 14].

Choosing a spherical indenter shape results in a situation where peak stresses are prevented and the stress distribution is of all solutions most uniformly divided over the contact surface. Therefore, the contacting surface area can be smaller when compared to conical or flat indenters. It can be concluded that a spherical indenter can transfer the stresses from two objects in the most efficient manner in terms of particle generation.



**Figure 20.** Pressure distribution along indenter profile for flat- (blue), spherical- (orange) and Conical (red) indenter

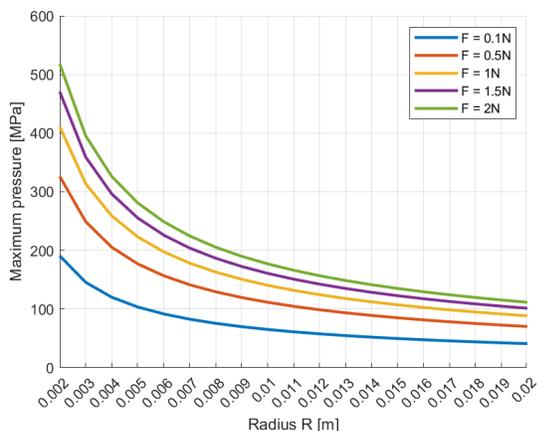
### 4.2 Indenter dimensions

A spherical indenter should be used to uniformly distribute the contact stresses. Hertz [26, 27] found the relation between a circular contact, applied normal force, indenter radius and the elastic material property of the two contacting bodies. Using Hertz contact theory, the maximum stresses can be calculated for a spherical indenter on a flat surface, equation 6 is used. In figure 6 an example is given where the maximum Hertz contact stresses are plotted versus a range of spherical indenter radius and normal force. The Matlab code used is given in appendix B.2.

$$p_{max} = \sqrt[3]{\frac{6FE^*}{\pi^3 R^2}} \tag{6}$$

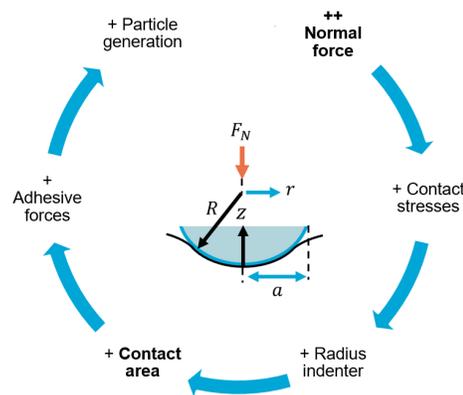
Where:

- $p_{max}$  = maximum pressure spherical indenter [Pa]
- $E^*$  = Equivalent E-modules [Pa]
- $R$  = Radius of spherical indenter [m]
- $F$  = Contact force in [N]



**Figure 21.** Example maximum Hertzian contact stress versus indenter radius and normal force. Material aluminium-aluminium

Adhesion wear increases with increasing normal force, as dictated by the Archard-Holmes equation (eq. 1), and contact surface area [5, 14]. Normal force and contact surface area are correlated with each other (figure 22). Minimizing normal forces will result in a decrease in the local maximum contact pressure. The required radius of the spherical indenter need to be able to keep the maximum Hertzian contact stresses below 80% of the yield stresses for the weakest contacting material. Lowering the radius of the indenter will decrease the contact surface area. To obtain minimum adhesive wear the applied normal force should be kept minimal, this in order to obtain the smallest possible indenter radius.



**Figure 22.** Cause-effect relationship applied normal force, indenter dimensions and particle generation

## 5. Guideline three: Constraints positions

In this chapter, the introduced particle reduction methods in chapter 2.3 and previously made conclusions in chapter 3.1 and 4.2 are incorporated for the determination of the optimal positions of the contact points. The position of the contact points and contact forces influences the positioning repeatability and particle generation. The last guideline introduces an optimization algorithm that can determine the optimal placement of the contact points. In appendix F the Matlab code is provided that is used to calculate the optimal position of the constraints.

### 5.1 Objective goals

A grasp of an object is defined by its constraints position and contact forces. These factors have an influence on positional repeatability and particle generation performance. To determine the optimal grasp the following objective goals are set to obtain the best possible grasp in terms of particle generation and repeatability.

#### Contact forces:

In chapter 4.2 'Indenter dimensions', it is stated that a higher normal force leads to higher adhesive wear. This is first described by the Archard-Holmes equation (eq. 1), where a higher normal force increases adhesive wear. Second, the indenter dimensions need to increase to reduce the maximum

Hertzian contact stresses. An increase in the indenter dimensions increases the contact area, which results in higher adhesive wear (figure 22).

The following design objective is formulated:

*The contact forces should be kept minimal and its magnitude is dependent on the required minimal nesting force.*

**Constraint position:**

The positioning repeatability performance is dependent on the position of the constraint and the magnitude of the contact forces. Yoa et al. [10] has performed experiments where the positioning repeatability is measured for different magnitudes of the restoring moments about the instantaneous center of rotation (ICR). They found that by maximizing the restoring moment about the ICR's, loads to better repeatability. The generated restoring moment is a function of the magnitude of the contact force and moment arm with respect to each ICR.

The following design objective is formulated:

*The restoring moment about all instantaneous center of rotations should be maximized while keeping the required nesting force as low as possible.*

**5.2 Optimising repeatability**

In chapter 3.1, it is stated that a form-closure grasp with seven contact points is required. To fully constrain a object with seven contact points, it is chosen to place three contact points on the bottom of the object and constraints translation in  $z$ -axis, rotation about  $x$ - and  $y$ -axis. The normal vector of these constraints is said to purely act in the  $z$ -axis ( $\mathbf{n} = [0 \ 0 \ 1]$ ), see figure 10. These constraints do not contribute to the set of restoring forces for the positioning of the object in  $x$ ,  $y$  and rotation about  $z$ . The four contact points located at the sides of the object are used to restore the position error in the  $XY$ -plane and rotation about the  $z$ -axis.

The goal for the optimizing algorithm is to find the optimal set of the remaining four contact points located at the sides of the object, in order to constrain and position translation in  $XY$ -plane and rotation about the  $z$ -axis. Together, the set needs to create maximum restoring moments about all ICR's (equation 7). In practice, the leading variable in the objective function is the ICR with the lowest restoring moment.

$$G_{ICR} = \max M_{ICR,i} \forall i \tag{7}$$

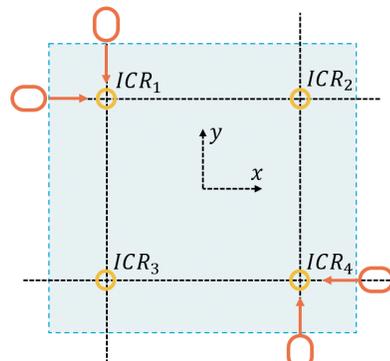
Where:

- $G_{ICR,i}$  = Objective score used for optimizing repeatability
- $M_{icr,i}$  = Restoring moment about  $i_{th}$  ICR

**5.2.1 Calculating objective function**

The instantaneous center of rotation (ICR) can be best explained as the point where no change in position can be measured for a certain translation and rotation of a rigid body [28].

All ICRs positions can be measured by finding the intersection points of the line of actions (LOA) for all constraints [29]. The slope of the LOA's are defined by the direction of the contact forces.



**Figure 23.** The position of the ICR's are determined by the intersecting points of the line of actions (LOA's). The slope of the LOA is equal to the direction of the contact forces.

Each contact point  $k$  generates a torque about the  $i_{th}$  ICR. The generated torque  $\mathbf{M}_{ICR,i}$  is calculated by the contact force  $\mathbf{f}_k$  and the distance  $\mathbf{d}_{k,i}$  between the  $k$  constraint position and the respective  $i$  ICR (equation 8). In appendix F the Matlab code is provided that is used to calculate the objective function.

$$\mathbf{M}_{ICR,i} = \sum_{k=1}^n \mathbf{d}_{k,i} \times \mathbf{f}_k \tag{8}$$

**5.2.2 Execution**

The algorithm uses a brute force approach to calculate the objective function 7, and the algorithm iterates over four input variables. Therefore, the runtime of the algorithm scales with  $O(n^4)$ , resulting in a dramatic increase in runtime for small increases of the data set. The data set needs to be pre-processed to exclude unnecessary data points and to reduce the resolution of the point cloud. In appendix E the used method is described and Matlab code is provided.

**5.3 Checking for stability**

The goal used to optimize for repeatability does not take the feasibility or stability of the grasp into account. The used Matlab code is provided in appendix C.

**5.3.1 Contact model**

In chapter 3.1 'Kinematics' it is stated that a form-closure grasp is required. A frictionless point-on-plane contact is used to model the stability of the grasp (equation 11). In practice, friction is present and will have a positive influence on the stability of the grasp. A point-on-plane contact with friction model can give a more realistic stability estimate. In literature, contact models with and without friction are described [30, 31].

Considering a local reference frame  $F$  (figure 24) with its origin at the contact point, the local  $z$ -axis is pointing towards the surface of the object. The force  $\mathbf{f}$  delivered by the constraint can be written as in equation 9.

$$\mathbf{f} = \mathbf{f}_{normal} + \mathbf{f}_{tangent} \quad (9)$$

The object surface normal vector would be described as  $\mathbf{f}_{normal} = [0, 0, f_z]^T$  and the frictional tangential forces as  $\mathbf{f}_{tangent} = [f_x, f_y, 0]^T$  in the local reference frame 'F'. The force vector always points towards the surface of the object and therefore  $f_z \geq 0$ .

The contact model can be further extended into a point-on-plane model with and without friction. The two contact models describe a set  $F$  of admissible forces the point-on-plane contact can apply:

1. Frictionless point-on-plane contact: The contact only delivers a force acting in the normal direction of the object's surface. No tangential forces are present ( $f_{tangent} = 0$ ) (figure 24):

$$F = \{\mathbf{f}_{normal} \mid f_z \geq 0\} \quad (10)$$

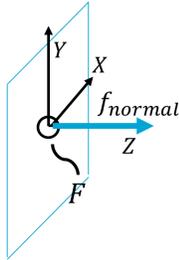


Figure 24. Point contact in its local coordinates system

2. Point-on-Plane contact with friction: The contact can deliver a normal and tangential force to the object. The admissible magnitude of the tangential forces is limited by the friction cone (figure 27), the friction cone is defined by the static friction coefficient  $\mu_s$ :

$$F = \{\mathbf{f} \mid \|\mathbf{f}_{tangent}\| \leq \mu_s \|\mathbf{f}_{normal}\|, f_z \geq 0\} \quad (11)$$

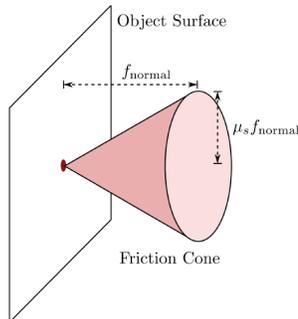


Figure 25. Friction cone [30].

To model the friction cone mathematically it is chosen, due to simplicity, to linearly discretize the friction into a finite set of vectors (figure 26). Second, it is chosen to inner approximate the friction cone, to prevent overestimating the admissible tangential forces.

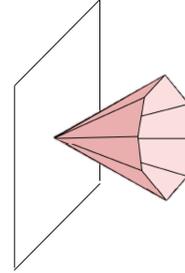


Figure 26. Linear discretized friction cone [30].

### 5.3.2 Grasp model

A grasp can be defined by a set of contact point models, which can together be combined into a single matrix and evaluated. In literature, this matrix that describes the positions of the contact forces is known as the grasp matrix  $\mathbf{G}$  [31, 32]. First, a wrench (equation 12) is defined for each contact point and describes how a single contact point influences the grasp.

$$\mathbf{w} = \begin{bmatrix} \mathbf{f} \\ \boldsymbol{\tau} \end{bmatrix} \in \mathbb{R}^n \quad (12)$$

Where:

$\mathbf{f}$  = Force generated by the contact point

$\boldsymbol{\tau}$  = Torque generated by contact point w.r.t. object CoM

$n$  = For 3D object  $n = 6$ , for 2D object  $n = 3$

Each contact point  $i$  generates a wrench on the object, the torque  $\boldsymbol{\tau}_i$  is calculated by the contact point force  $\mathbf{f}$  and the distance between the Center of Mass. The wrench can be also written as:

$$\mathbf{w}_i = \begin{bmatrix} \mathbf{f}_i \\ \mathbf{d}_i \times \mathbf{f}_i \end{bmatrix} \quad (13)$$

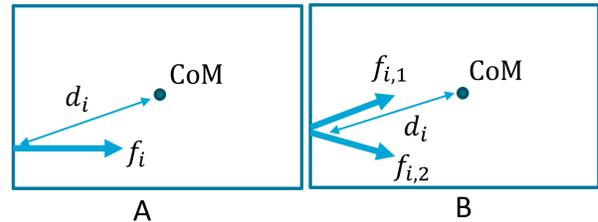


Figure 27. 2D object and point contact A) Point contact and moment arm w.r.t. CoM. B) Linearized friction cone of point contact and moment arm w.r.t. CoM.

A grasp can be defined as a set of wrenches that applies to the object for each contact point force  $\mathbf{f}_i$ . The corresponding

force for the  $i_{th}$  contact point can be linearly mapped into the  $i_{th}$  wrench as  $\mathbf{G}_i \mathbf{f}_i$ . Matrix  $\mathbf{G}_i$  is also referred to as the wrench basis matrix, this also included the transformation from the local contact reference frame  $F$  to the global reference frame  $N$ . The total wrench on the object consisting of  $k$  contact points, can be described with equation 14. In this equation, matrix  $\mathbf{G}$  is referred to as the grasp map.

$$\mathbf{w} = \sum_{i=1}^k \mathbf{G}_i \mathbf{f}_i = \mathbf{G} \begin{bmatrix} f_1 \\ \vdots \\ f_k \end{bmatrix}, \quad \mathbf{G} = [\mathbf{G}_1 \dots \mathbf{G}_k] \quad (14)$$

For a contact model with friction, each contact point generate a linear discretized friction cone consisting of  $m$  wrenches. The edge of the friction cone can be defined by a set of  $m$  forces (equation 15):

$$\{\mathbf{f}_{i,1}, \mathbf{f}_{i,2}, \dots, \mathbf{f}_{i,m}\} \quad (15)$$

Then each wrench basis matrix for a contact point with friction is described as:

$$\mathbf{G}_i = \sum_{j=1}^m \begin{bmatrix} \mathbf{f}_{i,j} \\ \mathbf{d}_i \times \mathbf{f}_{i,j} \end{bmatrix} \quad (16)$$

The total wrench on the object and its grasp map for  $k$  contact points and for a friction cone discretized in  $m$  wrenches is given by:

$$\mathbf{w} = \sum_{i=1}^k \sum_{j=1}^m \mathbf{G}_{i,j} \mathbf{f}_{i,j} = \mathbf{G} \begin{bmatrix} f_{1,1} & \dots & f_{1,m} \\ \vdots & \ddots & \vdots \\ f_{k,1} & \dots & f_{k,m} \end{bmatrix}, \quad \mathbf{G} = [\mathbf{G}_1 \dots \mathbf{G}_k] \quad (17)$$

### 5.3.3 Grasp quality measure

The convex wrench hull of the grasp (equation 18) can be calculated to determine how well the grasp can resist external disturbance forces [32]. For a frictionless point-on-plane contact  $m$  is equal to one.

$$W = \{ \mathbf{w} \mid \mathbf{w} = \sum_{i=1}^k \sum_{j=1}^m \alpha_{i,j} \mathbf{w}_{i,j}, \quad \mathbf{w}_i = \begin{bmatrix} \mathbf{f}_i \\ \mathbf{d}_i \times \mathbf{f}_i \end{bmatrix}, \quad (18) \\ \sum_{i=1}^k \sum_{j=1}^m \alpha_{i,j} = 1, \quad \alpha_{i,j} \geq 0 \}$$

The wrench hull is used to determine a boundary condition called  $Q_{LRW}$ . The radius ( $\epsilon$ ) of the largest ball that can be fitted in the volume of the 3D wrench hull (figure 33) is an indicator of how well the grasp can resist external disturbance forces for a given bound on the constraints reaction forces. The boundary condition is described in equation 19.

$$Q_{LRW} = \min \|\epsilon\| \quad (19)$$

Boundary condition  $Q_{LRW}$  can be used to determine the following [30]:

1. The radius of the largest ball centered at the origin that can be fitted in the wrench hull represents the magnitude of the smallest external disturbance wrench that can be resisted by the grasp. The opposite direction of the vector directing from the origin to the nearest surface of the wrench hull identifies the direction where the grasp is least able to resist external disturbance wrench.
2. The origin of the wrench hull need to be contained in the interior of the wrench hull to make the grasp in force closure. In other words, if the radius of the ball is equal to zero, then the grasp is not stable.

For a 2D object, the wrench hull dimensions are 3D ( $W \in \mathbb{R}^3$ ). For 3D objects, the wrench hull dimensions are 6D ( $W \in \mathbb{R}^6$ ). In the following chapter, a wrench hull is displayed for a 2D object without friction in figure 33 and 34. In appendix C, an example is shown for a 2D shape with friction in figure 48, 49 and 50, this resembles a more realistic prediction of the stability. The mathematical expression of equation 18 can also be explained using these figures, since the wrench hull is the convex hull of the wrenches  $\mathbf{w}_{i,j}$ .

## 6. Case study

In this chapter, a case study is performed using the guidelines and optimization algorithm introduced in this paper. The object that is studied is based on a part of a silicon wafer with the following properties:

- Size: 18x18x0.7 mm; Solid
- Material: Silicium
- Mass: 0.52 grams



Figure 28. Case study: Object

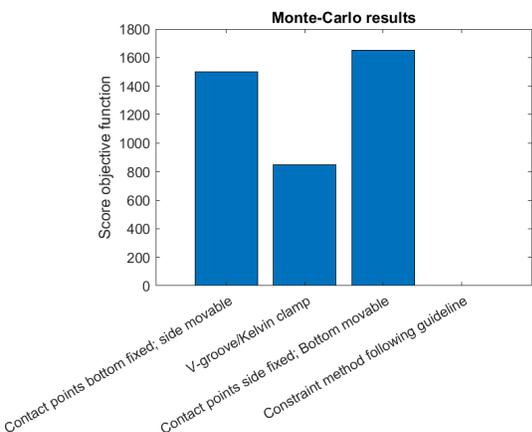
### 6.1 Maximum nesting force and suspension setup

The maximum nesting force is bounded by the allowed tangential contact force at the three contacts located at the bottom of the object. The tangential force can not exceed the static friction force, otherwise, slip will be present. The maximum allowable is dependent on the stiffness of the suspension since the tangential force will deliver the energy to obtain the required stroke of the suspension setup. This feedback loop is elaborated upon in more detail in appendix D.2. For an

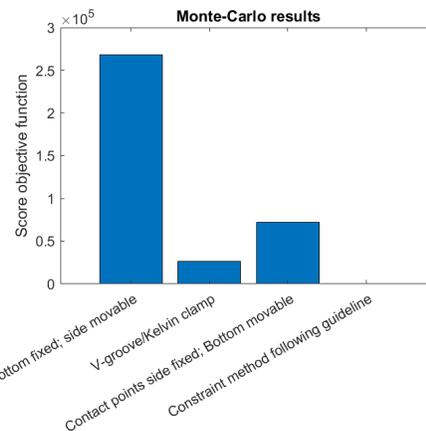
object mass of 0.52 grams, the maximum tangential force is calculated to be  $F_t = 0.0038\text{N}$ . This is assumed to be low, since no feasible suspension can be designed that is able to make the required stroke of  $80\ \mu\text{m}$ .

To fixate an object with extremely low mass, best practice measures for these situations are described below. Using option three particle generation can still be minimized according to the Monte-Carlo simulation (figure 30). This option is chosen to be used.

1. Use a V-groove, Slip will occur at these contact points, but will be limited since the applied contact force is low due to the low mass of the object. The expected particle generation due to abrasive and adhesive wear can be explained by using the Monte-Carlo simulation (figure 29). Since the object mass is very low the expected particle generation could be still low enough for the specific situation.
2. Make the three contact points located at the bottom of the object rigid. Slip will occur at the contact points, this solution scores significantly worse when compared to a V-groove in the Monte-Carlo simulation (figure 29).
3. Add mass to the object, for example by fixating the object in an external fixture. Doing this will result in higher allowable tangential force at the contact points located at the bottom of the object. Therefore, the larger tangential force can achieve a higher stroke of the suspension setup without the risk of slip. By adding an additional 15 grams the maximum tangential force is increased to  $F_t = 0.1\text{N}$ . It is now possible to design a suspension for the contact points located at the bottom of the object. The expected particle generation determined by the Monte-Carlo simulation due to abrasive and adhesive wear is near zero (figure 30).



**Figure 29.** Statistical analysis abrasive and adhesive wear, results for the studied object with mass 0.52 grams. Option 'Constraint method following guideline' is only feasible by adding additional mass.

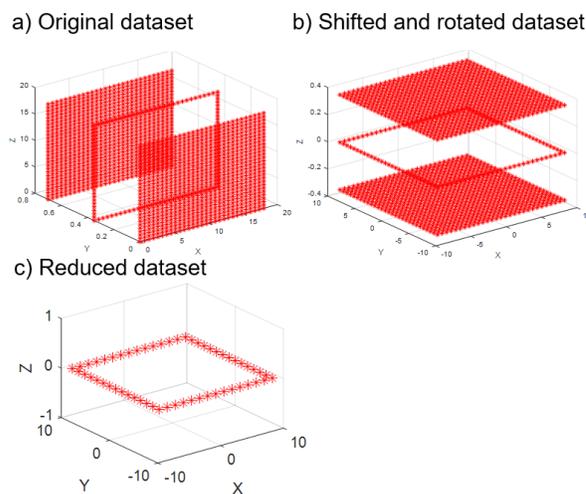


**Figure 30.** Statistical analysis abrasive and adhesive wear, results for the studied object with additional mass 15 grams.

### 6.2 Pre-processing object

A point cloud that resembles the shape of the object is used as input. The imported point cloud should be adapted in order to use it as input in the optimization algorithm. The following actions are taken:

1. Adapt orientation of the point cloud data. It is desired to have the origin (0,0,0) in the middle of the object and to orientate the object so that its height is displayed in the Z-axis (step B, figure 31). Appendix E.2 describes the method and the Matlab code used is given in appendix E.3.
2. Reduce the number of data points in the point cloud from 196 to 56 data points. Runtime scales with  $O(n^4)$ , which result in 1.5-hour runtime for 56 data points (step C, figure 31). The appendix E describes the procedure for how the number of data points is reduced and appendix E.4 provides the Matlab code.

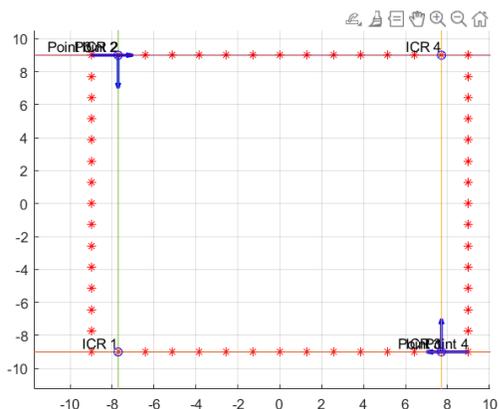


**Figure 31.** Pre-processing point cloud. A) Original dataset. B) Shifted and rotated dataset. C) Reduced dataset.

### 6.3 Optimising grasp repeatability

The optimal position of the contact points is determined by an objective goal that maximizes the restoring moments about the instantaneous centers of rotation. The algorithm determines the location of all ICR's and calculates the restoring moments for each ICR, the goal is to maximize all restoring moments. In appendix F the used code is described.

The optimal grasp places two contact points in the outer corner on opposite sides (figure 32). Note that for each data point the algorithm calculates a single normal vector for this point. In reality, for the square block below two normal vectors with different directions could be placed at the outer edges of the square block. In this case study, a better objective score would be obtained if the two contact points were both placed at the outer edge of the corner.

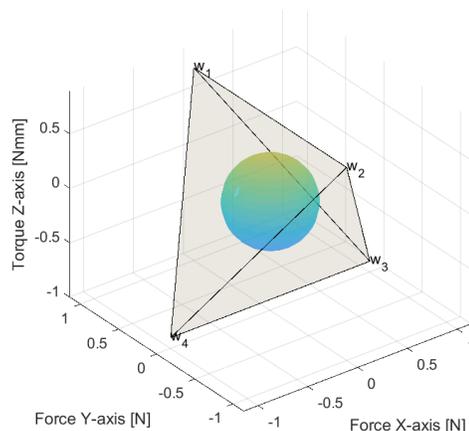


**Figure 32.** Best objective score optimization positions constraints. Circles are the location of the ICR's, Arrows are in the direction of applied contact force

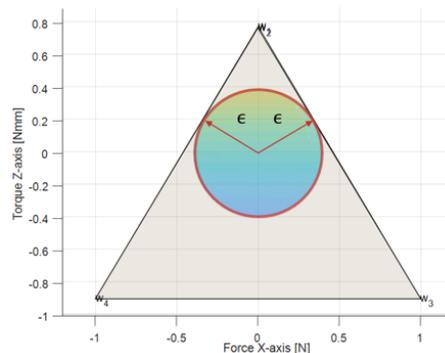
### 6.4 Checking for grasp stability

The constraint position optimization algorithm does not take the stability of the grasp into account. A grasp wrench hull is calculated to determine the ability to withstand external disturbance forces (figure 33). The algorithm method is explained in chapter 5.3, and the used Matlab code is provided in appendix C.

The radius of the largest ball ( $\epsilon$ ) that can fit inside the volume of the grasp wrench hull is equal to the largest allowable magnitude of external disturbance wrenches. The opposite direction where the sphere touches the boundary of the grasp wrench hull indicates the direction where the grasp is least able to resist external wrenches, see figure 34.



**Figure 33.** Wrench hull of optimized grasp. The sphere's radius and direction where the sphere touches the boundary of the wrench hull indicate the maximum allowable external wrench and its direction.



**Figure 34.** Sideview of figure 33: Largest sphere radius  $\epsilon$  equals magnitude- and opposite direction of  $\epsilon$  is the direction of the least able to resist external wrench.

### 6.5 Indenter dimensions

In chapter 4.1 'Indenter shape' the guidelines are stated and can be summarized by:

1. The optimal indenter shape is spherical.
2. The radius of the indenter should be kept minimal in order to minimize adhesive wear by reducing the contact area.
3. The maximum Hertz stresses calculated using 6 should not exceed 80% of the yield strength for the weakest contacting material.

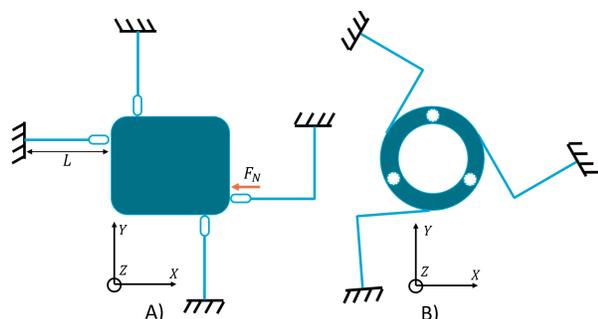
The yield strength of silicon is 170 MPa [33], and the maximum allowable stress is 136 MPa. The nesting force, which is based on the required tangential force to obtain the required stroke is equal to  $F_n = 0.1N$ . The applied contact force due to gravitational forces is equal to  $F_g = 0.05N$ . The highest contact force is used to determine the indenter dimension. Using equation 6 and rounding the result to the nearest millimeter integer result in an indenter radius of 4 mm.

## 6.6 Suspension setup

In chapter 3.1 'Kinematics' it is stated that each constraint needs to have freedom of motion and what direction is required. Multiple solutions can be derived to obtain a suspension setup that can provide the required low stiffness in the free axis and high stiffness in the constraint directions. In this section, an example is given of a possible suspension setup.

### 6.6.1 Building blocks

The four contact points that will constrain translation in the  $X$ - and  $Y$ -axis ( $\delta_x$ ,  $\delta_y$ ) and rotation about  $z$  ( $r_z$ ) can be suspended using leaf springs (figure 35a). The remaining three contact points constrain translation in  $z$ -axis  $\delta_z$  and rotation about  $x$ - and  $y$ -axis ( $r_x$ ,  $r_y$ ). The motion between these three contact points needs to be identical and allow for translation in the  $X$ - and  $Y$ -axis and rotation about the  $z$ -axis. Three double-folded leaf springs connected to a base with the three contact points can be used (figure 35b)

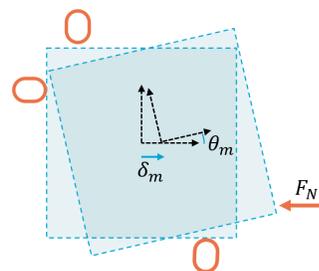


**Figure 35.** Top view of object A) Contact points located at the sides of the object suspended by leaf springs constraining  $\delta_x$ ,  $\delta_y$  and  $r_z$ , one contact point delivers the nesting force and is suspended by a folded leaf spring. B) Three contact points located at the bottom of the object suspended by three folded leaf springs constraining  $\delta_z$ ,  $r_x$  and  $r_y$ .

### 6.6.2 Minimum required stroke

The required stroke of the suspension setup is dependent on the positioning error and an extra rotational offset margin. A standard deviation of  $\sigma_t = 10 \mu\text{m}$  is used as a translational positioning error in the  $XY$ -plane, and an angular positioning error about the  $Z$ -axis of  $\sigma_a = 0.007^\circ$  [24].

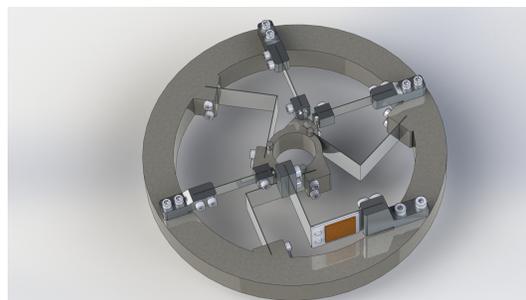
An extra rotational offset  $\theta_m$  is set to be  $0.3^\circ$ , and the translational offset  $\delta_m$  is set to be  $80 \mu\text{m}$  (figure 36). The extra rotational and translational offset enables it to insert the object into the kinematic coupling without touching the contact points located at the sides of the object. The extra margin is set to be equal to obtain 99,7% ( $3\sigma$ ) successful placements. The used Matlab code is provided in appendix G.



**Figure 36.** Schematic overview offset margins to prevent collisions with the contact points.

### 6.6.3 Concept example

In appendix D the derivation of the dimensions of the suspension setup is described. To passively position the object in the  $XY$ -plane it is required to have three constraints to be rigid and one constraint to be the nesting force. The further implementation of an actuation system for the nesting force is outside the scope, in the proposed concept example a piezoelectric transducer is used as actuator for reference [34]. In figure 37 an example is given where all previously described results are implemented.



**Figure 37.** Example of implementation of described guidelines and optimization.

## 7. Discussion

### 7.1 Main findings

The guidelines and optimization methods are a new approach in a tailored solution approach for designing high repeatability kinematic couplings with low particle generation. The introduced guidelines are applicable to any arbitrary object's shape, size and mass. The particle generation mostly originates during the positioning procedure of the object, abrasive and adhesive wear are the predominant factors that have an influence on particle generation. The first guideline is about preventing or minimizing these types of wear. Here, the contact points can be suspended using a spring system. In this way, the contact point obtains a freedom of motion in directions it does not need to constraint, while keeping the contact point rigid in the direction it needs to constrain. This prevents slip and therefore abrasive- and adhesive wear are minimized. It should be noted that this is only valid when the no-slip condition holds. In general, this no-slip condition assumes that a body starts sliding when the tangential forces on a

surface exceed the static friction force. Possibly this assumption does not hold on a micron scale level. An additional benefit of using this constraint method is that it minimizes the correlation between particle generation and gravitational forces. For conventional kinematic couplings, like a V-groove or Kelvin-clamp, particle generation increases with increasing mass due to the higher contact forces that undergo sliding. The Monte-Carlo analysis confirms that for higher masses the benefit of applying flexible contact points increases for higher masses. On the other side, for low object masses, one can make the consideration that the extra effort and corresponding limitations as described in the next subsection are not worth the slight decrease in particle generation.

Although the presented Alchard-Holmes equation (equation 1) states that adhesive wear is eliminated when no sliding between the object and constraint is present, adhesive wear will be still present on a smaller scale. Every instance two bodies make contact, the asperities of the two surfaces will weld together. When the contact points disengage, the welded asperities break, and particles are generated. This type of particle generation can not be prevented, but can be reduced by minimizing the contact area [5, 14]. This leads to the second introduced guideline of this study. Here, it is said that the contact points should be spherical to prevent peak stresses and the radius of the spherical indenter should be kept as small as possible to minimize the contact surface area. In order to minimize the radius the applied contact force should be minimalized.

The last introduced guideline is regarding the correlation between positioning repeatability and particle generation. The positioning repeatability is dependent on the ability of the grasp to nest the object. This ability can be determined by the magnitude of the restoring moment about all the instantaneous center of rotations (ICRs), this restoring moment should be maximized in order to obtain the best repeatability performance [10, 35]. This restoring moment is dependent on the position of all the constraints, which defines where the ICRs are located. The magnitude of the restoring moment is dependent on the moment arm between the contact points and ICRs and second it is dependent on the contact forces. These contact forces should be minimal to reduce particle generation. Any object's shape, size and mass can be used as input in the algorithm to give a quick guide on what the optimal solution should be for constraining the object. A second algorithm constructs the convex wrench hull of the grasp to check the stability of the grasp. Here, a minimum value can be set on what the grasp should be able to resist in terms of disturbance forces.

### 7.1.1 Applications fields

The introduced guidelines and methods provided in this study can be used to make a rapid kinematic coupling design that can be tailored and optimized for any object's shape, size and mass. This study is laid out so that an inexperienced

person can use the provided info to obtain knowledge of how particles are generated by a kinematic coupling. Thereafter, applying the guidelines is simplified for the user due to the use of multiple algorithms. The positioning optimizing method is made in a way that point clouds generated by a 3D-CAD model can be used as input.

## 7.2 Limitations

The proposed guideline that prescribes suspending the contact points, is in practice difficult to apply for objects with very low masses. The suspension should have a low enough stiffness in order to make the required stroke without letting the tangential contact forces exceeds the static friction force. This tangential force is determined by the minimum required force that is required to achieve the required stroke for the suspension. Although the required stroke is very small (10-150  $\mu\text{m}$ ), there is a limit where it is no longer possible to construct the required low-stiffness suspension. The severity of this conclusion can be put into perspective. First, In a vacuum, the friction coefficient dramatically increases up to 4-6x compared to the friction coefficient of air at 1-atmosphere [22], which helps prevent the tangential contact forces from exceeding the static friction force. Second, using the results in the Monte-Carlo analysis it is concluded that there are diminishing returns in terms of particle generation for lower object masses when compared to a V-groove. Possibly, for some applications, it is not worth the extra effort to further restrict particle generation when the object mass is very low. A possible solution for a situation where the object mass is too low could be by adding object mass. This sounds contra-productive, normally particle generation drastically increases for heavier objects, but for a kinematic coupling that uses non-rigid constraints particle generation is independent of the object mass. This relation is supported by the Monte-Carlo analysis (figure 18). A second solution could be statically balancing the suspension mechanism, which can result in a decrease in the required actuation force of 85% [36]. Although designing and implementing a statically balanced mechanism adds complexity. The latter solution is highlighted in more detail in the next section 'design recommendations'.

A second limitation is also dependent on the mass of the object. As described above, low object masses result in a very low suspension stiffness. Due to the low stiffness of the suspension, any vibration is hard to damp. After positioning the vibrations originating from the environment or system the object is placed in are very likely to result in the movement of the constraints. This results in sliding between the constraints and the object after the object is positioned and will lead to particle generation.

The used optimization algorithm for the last guideline is subjected to some limitations. First of all, the algorithm runtime uses a brute force approach to calculate the objective function. Therefore the runtime scales with  $O(n^4)$ , which

resulted in a runtime of 2 hours for 60 data points. To run the algorithm the user has to pre-process the point cloud in order to exclude unnecessary points and lower the resolution of the point cloud. Lowering the resolution can result in multiple limitations, but the effects of these limitations can be mitigated with careful input from the user. First, the low-resolution results in large distance gaps between the data points, the optimal positions of the constraints could be at the intermediate position between two data points. Second, the algorithm calculates its normal vector using its nearest data-point, in some cases, this can result in the wrong calculation of the normal vector of the surface. The current algorithm requires a manual check if all calculated normal vectors are reasonable. In appendix H, an example is shown where sharp angles can result in the wrong calculation of normal vectors.

### 7.3 Design recommendations

Further research into the smarter implementation of the introduced guidelines and methods can improve the maximum achievable performance. Multiple limitations are described that are dependent on the object's mass. One of the limitations is that it is no longer feasible to design a suspension with a low enough stiffness, that is able to keep the minimum required actuation force below the static friction force. Statically balancing the suspension can be used to significantly reduce the required actuation force for a certain stroke. Examples of statically balanced mechanisms can obtain a reduction of up to 85% [36]. Statically balancing a compliant mechanism is generally done by adding members that deliver the opposite (negative) potential energy that is required to deform the suspension. The structural integrity of the structure is not reduced, since the stiffness of the individual member is not decreased. Therefore, the limit where the object mass is too low can be positively shifted by adding this property to the suspension setup. Kuppens [37] describes a mechanism with similar binary stiffness properties, although the introduced mechanism is only able to be in a high stiffness state in one singular position

A second limitation depending on the object's mass states that every vibration in the system will result in the movement of the constraints due to their low stiffness. Therefore, creating particles during the fixation period of the object. A way of solving this limitation could be through the use of compliant mechanisms that have binary stiffness properties. Here, a mechanism can be switched between a stiff state (positioned) and a soft state (during positioning). The mechanism should be in its soft state during the positioning of the object and should have high stiffness when the object is fixated to resist perturbations.

### 7.4 Experimental recommendations

Continuing on the quantification of the performance improvement by implementing the proposed methods with respect to conventional kinematic couplings. In this study, it is not

quantified how many particles will be generated by a conventional kinematic coupling and one that is designed following the proposed guidelines and optimization methods. Theoretically quantifying the number of particles or the wear volume is difficult and a method to do so was not found given the timeframe of this study. In order to give irrefutable proof of the performance increase due to the introduced guidelines an experiment should be executed. Here, the particle generation during positioning should be measured for conventional kinematic couplings and one that is designed following the guidelines introduced in this study.

Additionally, a second experiment can be constructed that can measure the relationship between restoring moments about the ICRs and positioning repeatability. Although it is in multiple papers concluded [10, 35] that the repeatability becomes more accurate when the restoring moment is maximized, in these papers this is only concluded for a single object. A more thorough study that can verify that the relationship is true for every arbitrary object shape and size using the provided optimizing algorithm introduced in this study can provide irrefutable proof of this relationship.

## 8. Conclusion

This study presented new guidelines and optimization methods that can be applied in designing a kinematic coupling with high repeatability that will be used in a particle-critical environment. The proposed guidelines can be applied for a tailored solution approach for any arbitrary object shape, size and mass. The first guideline states a constraint method that can eliminate sliding between the object and contact points, this eliminates and minimizes abrasive and adhesive wear respectively. The expected reduction in particle generation is determined using a statistical Monte-Carlo simulation. The results of this statistical analysis shows that for larger masses the benefit of applying this first guideline increases. The second guideline optimizes the contact point shape and dimensions. Here, the local contact stresses and contacting surface area are considered. The stress distribution is most evenly distributed using a spherical indenter shape, in which the radius should be kept minimal to reduce the contact surface area. The third and last guideline optimizes the positions of the contact points. These positions influence the positioning repeatability, the repeatability dependent on the position of the contact points and the applied contact forces. An optimization algorithm determines the grasp where all restoring moments about the instantaneous center of rotation are maximized, this results in a grasp which obtains the highest repeatability for a certain set of contact forces. A second algorithm has been introduced that checks and determines the ability of the grasp to withstand external disturbance forces. At last, to give an overview and an example how to implement the proposed guidelines and methods a case study is performed.

## References

- [1] S. D. Cheung, “In-Situ Monitoring of Particulate Contamination in Integrated Circuit Process Equipment,” *Particles in Gases and Liquids 1*, pp. 167–173, 1989.
- [2] D. W. Cooper, “Particulate Contamination and Microelectronics Manufacturing: An Introduction,” <http://dx.doi.org.tudelft.idm.oclc.org/10.1080/02786828608959094>, vol. 5, no. 3, pp. 287–299, 2007.
- [3] S. Carey, D. M. Knotter, E. Ooms, J. Boersma, E. Van Der Sar, R. Cop, W. Gerritzen, H. Van Zadelhoff, and H. Bouma, “Yield Impact of Backside Metal-Ion Contamination,” *Solid State Phenomena*, vol. 187, pp. 287–290, 2012.
- [4] D. W. Cooper, “Particulate contamination and microelectronics manufacturing: An introduction,” *Aerosol Science and Technology*, vol. 5, no. 3, pp. 287–299, 1986.
- [5] K. van den Broek, “Interview contamination expert VDL ETG,” 5 2022.
- [6] M. A. van de Kerkhof, E. Galutschek, A. Yakunin, S. Cats, and C. Cloin, “Particulate and molecular contamination control in EUV-induced H<sub>2</sub>-plasma in EUV lithographic scanner,” p. 18, 8 2020.
- [7] L. Lilje, “Controlling Particulates and Dust in Vacuum Systems,” *Glumsløv*, 2017.
- [8] H. Kobayashi, “Reducing particle contamination during pump-down in load lock chamber,” *Journal of the Vacuum Society of Japan*, vol. 53, no. 10, pp. 568–572, 2010.
- [9] A. Slocum, “Kinematic couplings: A review of design principles and applications,” *International Journal of Machine Tools and Manufacture*, vol. 50, pp. 310–327, 4 2010.
- [10] T. F. Yao, A. Duenner, and M. Cullinan, “In-line dimensional metrology in nanomanufacturing systems enabled by a passive semiconductor wafer alignment mechanism,” *Journal of Micro and Nano-Manufacturing*, vol. 5, 3 2017.
- [11] K. v. d. Broek, “[Internal] WOW - DESIGN FOR CONTAMINATION CONTROL,” tech. rep., VDL Enabling Technologies Group, 2021.
- [12] V. L. Popov, *Contact Mechanics and Friction*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017.
- [13] J. Dutta Majumdar and I. Manna, “Laser surface engineering of titanium and its alloys for improved wear, corrosion and high-temperature oxidation resistance,” *Laser Surface Engineering: Processes and Applications*, pp. 483–521, 1 2015.
- [14] D. H. Buckley, “NASA SP-277 FRICTION, WEAR, AND LUBRICATION IN VACUUM NATIONAL AERONAUTICS AND SPACE ADMINISTRATION,”
- [15] K. Jousten, “THERMAL OUTGASSING,” *CERN*, 1999.
- [16] J. M. Lafferty, *Foundations of Vacuum Science and Technology*. Wiley, 1998.
- [17] P. Řepa and M. Rott, “Outgassing of metals stimulated by friction,” *Vacuum*, vol. 48, pp. 775–778, 9 1997.
- [18] R. A. Nevshupa, J. L. De Segovia, and E. A. Deulin, “Outgassing of stainless steel during sliding friction in ultra-high vacuum,” *Vacuum*, vol. 53, pp. 295–298, 5 1999.
- [19] V. D. Nguyen, “Constructing Force-Closure Grasps,” *The International Journal of Robotics Research*, vol. 7, no. 3, pp. 3–16, 1988.
- [20] H. Soemers, *Design Principles for precision mechanisms*, vol. 978-90-365-3103-0. 2010.
- [21] F. Reuleaux, *The kinematics of machinery: outlines of a theory of machines*. New York: Courier Corporation, 2013.
- [22] E. A. Deulin, V. P. Mikhailov, Y. V. Panfi, and R. A. Nevshupa, “Mechanics and Physics of Precise Vacuum Mechanisms,”
- [23] R. L. Harrison, “Introduction to monte carlo simulation,” *AIP Conference Proceedings*, vol. 1204, no. 1, pp. 17–21, 2010.
- [24] Shibaura machines, “Small size SCARA Robot THL series.” [https://www.shibaura-machine.co.jp/en/product/robot/lineup/th/small\\_thl\\_series.html](https://www.shibaura-machine.co.jp/en/product/robot/lineup/th/small_thl_series.html). Accessed: 10-11-2022.
- [25] V. L. Popov, M. Heß, and E. Willert, “Handbook of contact mechanics: Exact solutions of axisymmetric contact problems,” *Handbook of Contact Mechanics: Exact Solutions of Axisymmetric Contact Problems*, pp. 1–347, 4 2019.
- [26] K. L. Johnson, “Normal contact of elastic solids – Hertz theory,” *Contact Mechanics*, pp. 84–106, 3 1985.
- [27] G. F. G. “Hertz’s Miscellaneous Papers,” *Nature 1896 55:1410*, vol. 55, no. 1410, pp. 6–9, 1896.
- [28] B. McCane, J. H. Abbott, and T. King, “On calculating the finite centre of rotation for rigid planar motion,” *Medical Engineering & Physics*, vol. 27, pp. 75–79, 1 2005.
- [29] D. L. Blanding, “Exact Constraint: Machine Design Using Kinematic Principles,” *Exact Constraint*, 8 1999.
- [30] Stanford University, “Course material: AA174B “Principles of Robot Autonomy II”.” [https://web.stanford.edu/class/cs237b/pdfs/lecture/lecture\\_567.pdf](https://web.stanford.edu/class/cs237b/pdfs/lecture/lecture_567.pdf). Accessed: 14-09-2022.
- [31] Y. Zheng and W. H. Qian, “Improving grasp quality evaluation,” *Robotics and Autonomous Systems*, vol. 57, pp. 665–673, 6 2009.
- [32] M. A. Roa and R. Suárez, “Grasp quality measures: re-

view and performance,” *Autonomous Robots*, vol. 38, pp. 65–88, 1 2015.

- [33] M. Properties, “Silicon - Strength - Hardness - Elasticity - Crystal Structure.” <https://material-properties.org/silicon-mechanical-properties-strength-hardness-crystal-structure/>. Accessed: 22-11-2022.
- [34] PI, “P-876 DuraAct Patch Transducer.” <https://www.pi-usa.us/en/products/piezo-actuators-stacks-benders-tubes/p-876-duraact-patch-transducer-101790#downloads>. Accessed: 24-11-2022.
- [35] L. C. Hale, “Friction-based design of kinematic couplings,” *Conference: American Society for Precision Engineering 13th Annual Meeting, St. Louis, MO, October 25-30, 1998*, 7 1998.
- [36] A. J. Lamers, J. A. Gallego Sánchez, and J. L. Herder, “Design of a statically balanced fully compliant grasper,” *Mechanism and Machine Theory*, vol. 92, pp. 230–239, 10 2015.
- [37] P. R. Kuppens, M. A. Bessa, J. L. Herder, and J. B. Hopkins, “Monolithic binary stiffness building blocks for mechanical digital machines,” *Extreme Mechanics Letters*, vol. 42, p. 101120, 1 2021.
- [38] JPE, “Three parallel & tangential folded leaf springs.” <https://www.jpe-innovations.com/precision-point/3-parallel-tangential-folded-leaf-springs/>. Accessed: 22-12-2022.
- [39] O. A. Bauchau and J. I. Craig, *Euler-Bernoulli beam theory*. Dordrecht: Springer Netherlands, 2009.
- [40] JPE, “Precision Point.” <https://www.jpe-innovations.com/precision-point/>. Accessed: 22-12-2022.

## A. Monte-Carlo simulation

The Monte-Carlo simulation is used to approximate particle generation based on positioning error and constraining methods. Modeling particle generation based on these parameters is difficult, the best obtained method to predict particle generation is through the use of the Archard or Archard-Holmes equation 20. Here  $k$  and  $\sigma$  are material properties and therefore independent on the design of a kinematic coupling. Normal force  $F$  and sliding distance  $s$  are dependent on the kinematic coupling design, therefore these parameters can be used to predict particle generation.

$$V = \frac{kFs}{\sigma} \quad (20)$$

Where:

$V$  = Wear volume  
 $k$  = Wear rate, differs for abrasive and adhesive wear  
 $F$  = Normal load  
 $s$  = Sliding distance  
 $\sigma$  = Hardness of softer material

An objective function (equation 21) incorporates the variables sliding distance and normal force. Here, for every placement of the object  $N$  it is determined for each constraint  $k$  what the plausible sliding distance ( $s$ ) and normal force ( $F_n$ ) will be. The final objective score for a kinematic coupling design is the summation over all iterations ( $N$ ).

$$Obj = \sum_{i=1}^N \sum_{j=1}^K s(i, j) F_n(i, j) \quad (21)$$

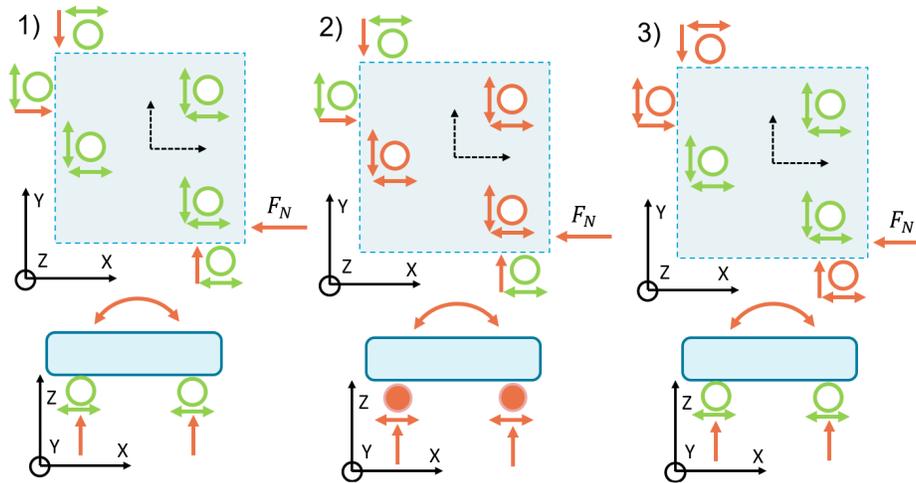
Where:

$Obj$  = Objective  
 $s$  = Sliding distance of constraint  $j$  for positioning error corresponding to iteration  $i$   
 $F_n$  = Normal force excited by constraint  $j$  for the  $i_{th}$  iteration  
 $N$  = Total number of iterations  
 $K$  = Total number of contact points

### A.1 Input

In a Monte-Carlo analysis, an object is positioned  $N$  times. The following objects are studied:

1. V-groove or Kelvin-Clamp
2. All contact points can translate parallel to the surface, thus eliminating all sliding when nesting the object (figure 38A). The parasitic motion of the suspension is taken into account, the suspension is modelled as a single clamped leaf spring.
3. contact points belonging to constraining  $\delta_z$ ,  $r_x$  and  $r_y$  are rigid, other contact points free to translate parallel to the surface, figure 38B.
4. The configuration is vice versa, contact points belonging to constraining  $\delta_x$ ,  $\delta_y$  and  $r_z$  are rigid, other contact points free, figure 38C.

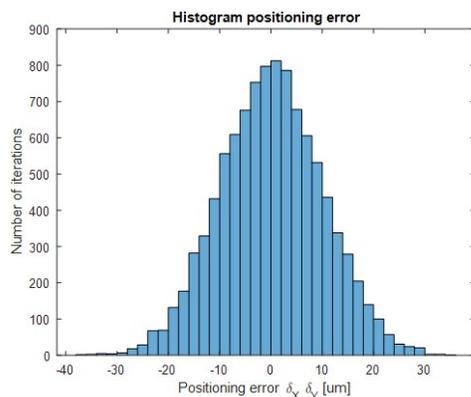


**Figure 38.** Three derivations of fixed and free contact points configurations. Red = fixed; Green = free.

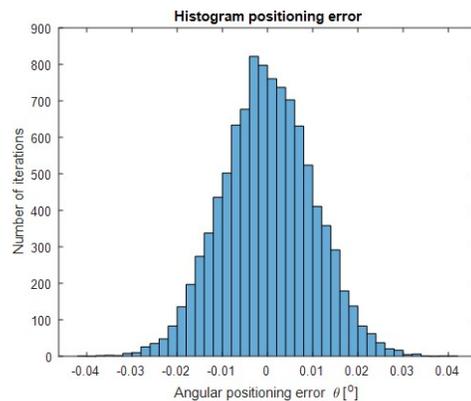
In each iteration, the position error is determined by the standard deviation for translation positioning error in the  $XY$ -plane and rotational error around the  $Z$ -axis. For reference, the specification of a 3DOF SCARA-robot with a range of 300 mm is used [24]. A standard deviation of  $\sigma_r = 10 \mu\text{m}$  is used as translational positioning error in the  $XY$ -plane, and an angular positioning error about the  $Z$ -axis of  $\sigma_a = 0.007^\circ$ . Furthermore, the results are dependent on the chosen object size. It is chosen to use a square object that is 18x18mm in size.

For a V-groove the sliding distance can be directly determined from these positioning errors. For the constraining methods displayed in figure 38, an extra positioning margin needs to be taken into account. When positioning the object can not be placed upon the contact points located at the sides of the object. Therefore, the positioning margin should be chosen that the object are successfully placed with a certainty of  $3\sigma_{xy}$ . In appendix G the Matlab code is provided which is used to determine the minimum required margin offset. For the studied object, this should be an offset of  $\delta_m = 80 \mu\text{m}$  in the  $x$ -axis and a rotational offset of  $\theta_M = 0.3^\circ$ .

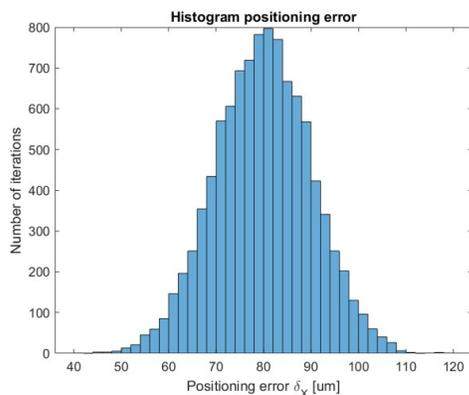
In figure 39 and 40 the positioning error used for a V-groove or Kelvin clamp is shown. In figure 39, 41 and 42 the positioning error in  $x$ ,  $y$  and rotational error respectively is shown for the new introduced constraint method.



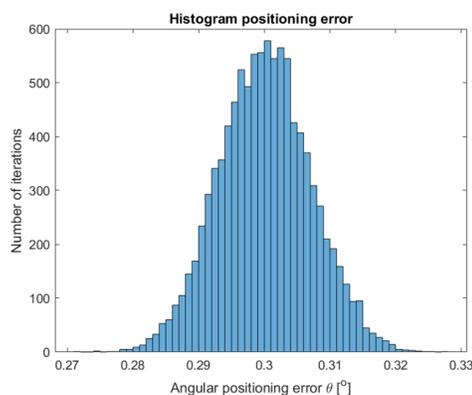
**Figure 39.** Histogram positioning error in  $XY$ -plane for a V-groove or Kelvin clamp.  $\sigma_{xy} = 10 \mu\text{m}$



**Figure 40.** Histogram rotational positioning error for a V-groove or Kelvin clamp.  $\sigma_{rz} = 0.007^\circ$



**Figure 41.** Histogram positioning error in  $X$ -axis for a V-groove or Kelvin clamp.  $\sigma_{xy} = 10 \mu\text{m}$  and offset  $\delta_M = 80 \mu\text{m}$ . Positioning error in  $Y$ -axis is equal to that in figure 39



**Figure 42.** Histogram rotational positioning error for suspended contact points method.  $\sigma_{rz} = 0.007^\circ$  and offset of  $\theta_M = 0.3^\circ$

## A.2 Algorithm procedure V-groove/Kelvin-Clamp

The algorithm needs to determine what contact points engage with the object, and what normal force and sliding distance there is during positioning. The approach to determine these variables is different for each studied constraint method. First, for a 'V-groove or Kelvin clamp' the following approach is used:

1. **Contact points:** For the nested position the V-groove or Kelvin clamp has six contact points when the object is not in the nested position it has three contact points.
2. **Sliding distance  $s$ :** Sliding distance is equal to the summation of positioning error in  $X$  and  $Y$ -axis.
3. **Normal force  $F_n$ :** The normal force is determined by the force acting in the vertical axis, in this situation it is determined by the mass of the object. When the object is not in its nested position the mass of the object is divided over three contact points. Therefore, the normal force is equal to the mass of the object divided by 3.

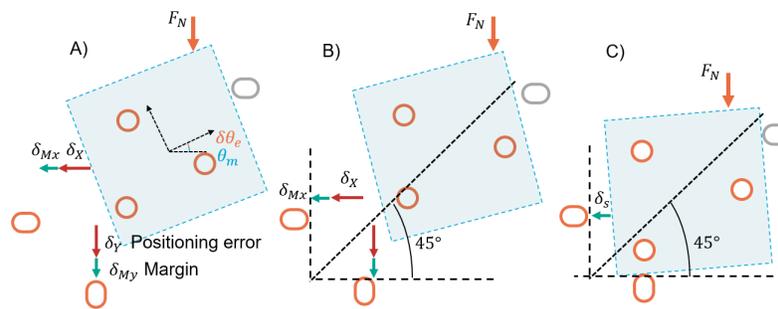
## A.3 Algorithm procedure constraint method

**Contact points:** The object is placed onto three contact points, which make contact during the entire positioning procedure.

Furthermore, four contact points are located at the side of the object. One of these contact points delivers the nesting force. The nesting force is said to be in contact during the entire positioning procedure. The object is fully fixated at the end of the positioning procedure when the object is in contact with all four side contact points

During positioning, the object can obtain an additional one or two contact points besides the already present nesting force contact. Predicting how the object will rotate about which ICR and how it will translate when it is being pushed by the nesting force is complex to model, for example, due to its time-dependent characteristics. Therefore, a simplification is used that states that the object will always translate in a  $45^\circ$  angle (figure 43). The following steps are taken into account when determining the motion behavior of the object.

1. The object is positioned with a translational and rotational positioning error and margin (43A).
2. The object will translate in a  $45^\circ$  angle. Eventually, it reaches one of the contact points located at the side (43B-C).
3. When it reaches one of the contact points, the object moves in the still not fully constrained direction. (43B-C). During this last movement, the constraint located at the side is in contact during the entire last positioning part.
4. During the positioning the object will rotate. This creates an extra movement for all contact points, only the three contact points located at the bottom of the object are incorporated.



**Figure 43.** Schematic procedure to determine sliding distance dependent on positioning error and margin

For the object where all contact points are suspended and can free constrained by suspended (figure 38A) the variables are determined as follow:

1. **Sliding distance  $s$ :** If the constraint is suspended and is free to translate as all contact points are free to move and only generate friction due to the parasitic motion of the suspension. The stroke that is made depends on two factors. First, the positioning error and self-set margin, in figure 39, 41 and 42 these positioning errors are given. Second, the movement of the object during positioning determines when the object will slide over contact points. How this movement is determined when contact is made with the object is explained above in 'contact points'.

For the constraint method derivations shown in 38B and 38C, friction is present when contact is made with the rigid constraint. The sliding distance that will create friction is determined by the motion of the object during positioning. The motion of the object during positioning is described above in 'contact points'.

2. **Normal force  $F_n$ :** The normal force on all contact points is determined based on the minimal required nesting force. The nesting force is dependent on the minimum required actuation force to obtain the minimum required stroke. The stroke is dependent on the stiffness of the suspension, a lower stiffness results in a decrease in the required actuation force. Second, the required stroke is dependent on the positioning error and extra offset margin. This all creates a feedback loop where the goal is to obtain as low as possible stiffness.

For constraints methods using rigid contact points (figure 38B-C), lock-in can occur. This is not incorporated in determining the minimum required nesting force.

#### A.4 Matlab: MonteCarlo analysis

```

1 clear
2
3 syms
4
5 %Input variables
6
7 N = 10000 %Number of iterations
8 STD = 10 %Standard deviation for XY positioning error
9 STD_angle = 0.007; %Standard deviation angular positioning error
10 offset = 0.3; %Offset angle in order to be able to place the object
11 margin = 80 %Set positioning margin
12 L = 18 %V-groove kinematic coupling radius
13 m = 0.2 %Mass of object
14
15 %Calculate standard deviation curve
16 X = STD.*randn(N,1);
17 Y = STD.*randn(N,1);
18 theta_vgroove = STD_angle.*randn(N,1)
19 theta = (STD_angle.*randn(N,1))+offset;
20
21 %Calculate maximum positioning errors including margin
22 Xm = X + margin;
```

```

23 Ym = abs(Y);
24
25 %Contact force
26 F_vgroove = m*9.81/3; %V-Groove or table force
27 F_n = 0.1; %Nesting force
28
29 %Length springs
30 L_spring = 35;
31 L_um = L_spring*1000;
32
33 %Initialize mass of object
34 m_start = 0.01
35 m_end = 0.4;
36 m_step = 0.01;
37 m_range = m_start:m_step:m_end;
38
39 %Initialize counters
40 tablecounterX = 1;
41 tablecounterY = 1;
42 movablecounterX = 1;
43 movablecounterY = 1;
44 negative = 0;
45
46 %Objective scores pre-determined due to large margin
47 obj_table_Y = 0;
48 obj_movable_Y = 0;
49
50 histogram(Xm);
51 title('Histogram positioning error')
52 xlabel('Positioning error \delta_X [um]')
53 ylabel('Number of iterations')
54
55 histogram(Y);
56 title('Histogram positioning error')
57 xlabel('Positioning error \delta_Y [um]')
58 ylabel('Number of iterations')
59
60 histogram(theta);
61 title('Histogram positioning error')
62 xlabel('Angular positioning error \theta [^\circ]')
63 ylabel('Number of iterations')
64
65
66 %V-groove objective function calculations
67 for i = 1:1:N
68     Vgroove_x(i) = 3*abs(X(i))*F_vgroove;
69     Vgroove_y(i) = 3*abs(Y(i))*F_vgroove;
70     Vgroove_theta(i) = 3*sind(abs(theta_vgroove(i)))*L*1000*F_vgroove;
71 end
72
73 %Table moving (3 constrain)
74 for i = 1:1:N
75     if Xm(i) < 0 || Ym(i) < 0
76         negative = negative +1;
77         continue

```

```

78     end
79     if Xm(i) > Ym(i)
80         Xrest = Xm(i)-tand(45)*Ym(i);
81         obj_table_X(tablecounterX) = Xrest*F_n;
82         tablecounterX = tablecounterX + 1;
83     end
84
85     if Xm(i) < Ym(i)
86         Yrest = Ym(i)-tand(45)*Xm(i);
87         obj_table_Y(tablecounterY) = Yrest*F_n;
88         tablecounterY = tablecounterY + 1;
89     end
90
91 end
92
93 %Table fixed mz dx dy movable
94 for i = 1:1:N
95     if Xm(i) < 0 || Ym(i) < 0
96         negative = negative +1;
97         continue
98     end
99
100     obj_side_X(i) = Xm(i)*F_vgroove*3;
101     obj_side_Y(i) = Ym(i)*F_vgroove*3;
102     obj_side_theta(i) = 3*sind(abs(theta(i)))*L*1000*F_vgroove;
103 end
104
105
106 %All movable constraints
107 for i = 1:1:N
108     if Xm(i) < 0 || Ym(i) < 0
109         negative = negative +1;
110         continue
111     end
112
113     Vmovable_x(i) = 3*(3/5)*(abs(X(i))^2/L_um)*F_vgroove;
114     Vmovable_y(i) = 3*(3/5)*(abs(Y(i))^2/L_um)*F_vgroove;
115
116     if Xm(i) > Ym(i)
117         Xrest = Xm(i)-tand(45)*Ym(i);
118         Xrest_par = (3/5)*(Xrest^2/L_um);
119         obj_movable_X(movablecounterX) = Xrest_par*F_n;
120         movablecounterX = movablecounterX + 1;
121     end
122
123     if Xm(i) < Ym(i)
124         Yrest = Ym(i)-tand(45)*Xm(i);
125         Yrest_par = (3/5)*(Yrest^2/L_um);
126         obj_movable_Y(movablecounterY) = Yrest_par*F_n;
127         movablecounterY = movablecounterY + 1;
128     end
129 end
130
131 %Summation of all seperate objective scores
132 Obj_Vgroove = sum(Vgroove_x) + sum(Vgroove_y) + sum(Vgroove_theta)

```

```

133 Obj_table = sum(obj_table_X) + sum(obj_table_Y)
134 Obj_movable = sum(obj_movable_X) + sum(obj_movable_Y) + sum(Vmovable_x) + sum(Vmovable_y)
135 Obj_side = sum(obj_side_X) + sum(obj_side_Y) + sum(obj_side_theta)
136
137
138 barlabel = {'Constraints M_{xy}, \delta_z fixed; M_z, \delta_{xy} movable', 'V-groove', '
           Constraints M_z, \delta_{xy} fixed; M_{xy}, \delta_z movable', 'All constraints movable w/
           parasitic motions'}
139 bar_x = categorical(barlabel);
140 bar_x = reordercats(bar_x,barlabel);
141 bar_y = [Obj_side, Obj_Vgroove, Obj_table, Obj_movable];
142 figure(1)
143 bar(bar_x,bar_y)
144 title('Monte-Carlo results')
145 ylabel('Score objective function')
146
147 %Determine objective score for range of masses
148 for k = 1:1:length(m_range)
149
150     m = m_range(k);
151
152     %V-groove objective function calculations
153     for i = 1:1:N
154         Vgroove_x(i) = 3*abs(X(i))*((m*9.81)/3);
155         Vgroove_y(i) = 3*abs(Y(i))*((m*9.81)/3);
156         Vgroove_theta(i) = 3*sind(abs(theta(i)))*L*1000*((m*9.81)/3);
157         obj_side_X(i) = Xm(i)*((m*9.81)/3)*3;
158         obj_side_Y(i) = Ym(i)*((m*9.81)/3)*3;
159         obj_side_theta(i) = 3*sind(abs(theta(i)))*L*1000*((m*9.81)/3);
160     end
161
162     Obj_Vgroove(k) = sum(Vgroove_x) + sum(Vgroove_y) + sum(Vgroove_theta);
163     obj_side(k) = sum(obj_side_X) + sum(obj_side_Y) + sum(obj_side_theta);
164     Obj_table_m(k) = Obj_table;
165     Obj_movable_m(k) = Obj_movable;
166 end
167
168 %Plotting mass versus objective
169 lijnbreedte = 4;
170 figure(2)
171
172 hold on
173 grid on
174 plot(m_range, Obj_Vgroove, 'LineWidth',lijnbreedte)
175 plot(m_range, obj_side, 'LineWidth',lijnbreedte)
176 plot(m_range, Obj_table_m, 'LineWidth',lijnbreedte)
177 plot(m_range, Obj_movable_m, 'LineWidth',lijnbreedte)
178
179 title('Monte-Carlo analysis results for multiple object masses')
180 xlabel('Object mass [kg]')
181 ylabel('Score objective function [Nmm]')
182
183 legend('V-groove/Kelvin clamp', 'Constraints M_{xy}, \delta_z fixed; M_z, \delta_{xy} movable', '
           Constraints M_z, \delta_{xy} fixed; M_{xy}, \delta_z movable', 'All constraints movable w/
           parasitic motions')

```

## B. Indenter shape

In this chapter, the pressure distribution for a flat, conical and spherical indenter is determined. The pressure distribution for a variety of indenter shapes can be calculated using Hertz contact theory and Method of Dimensionality Reduction (MDR) [25, 26]. A frictionless contact is considered where two elastic bodies come in contact. First, to estimate the stresses and deflections located at the contact between the indenter and object the equivalent E-modules is determined.

$$\frac{1}{E^*} = \frac{1 - \nu_1^2}{E_1} + \frac{1 - \nu_2^2}{E_2} \quad (22)$$

Where:

$E^*$  = Equivalent E-modules

$\nu_i$  = Poisson ratio for material i

$E_i$  = E-modules for material i

First, in figure 44 a flat indenter is displayed, the stress distribution in the z-axis for the contact area is given by formula 23.

$$\sigma_{zz}(r, d) = -\frac{E^* d}{\pi \sqrt{a^2 - r^2}}, \quad r \leq a \quad (23)$$

Where:

$\sigma_{zz}$  = Stress in perpendicular to surface

$E^*$  = Equivalent E-modules

$a$  = Penetration radius

$r$  = Radius with origin in center of indenter

$d$  = Indenter depth

The average pressure distribution of a flat indenter is given by equation 24:

$$p_0 = \frac{2E^* d}{\pi a} \quad (24)$$

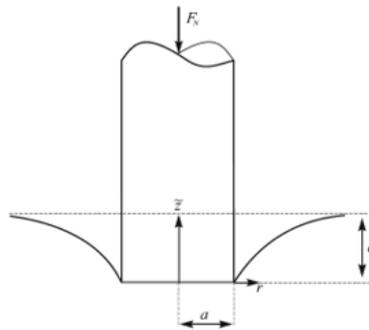
Where:

$p_0$  = Average pressure flat indenter

$E^*$  = Equivalent E-modules

$a$  = Penetration radius

$d$  = Indenter depth



**Figure 44.** Flat indenter

Second, in figure 45 a conical indenter is displayed. The normal stress distribution for this indenter shape can be calculated with equation 25.

$$\sigma_{zz}(r, d) = -p_0 \operatorname{arcCosh}\left(\frac{a}{r}\right), \quad r \leq a \quad (25)$$

Where:

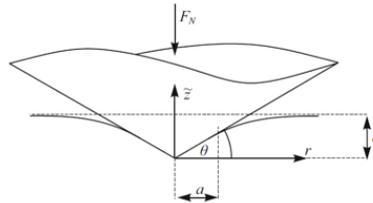
- $\sigma_{zz}$  = Stress in perpendicular to surface
- $a$  = Penetration radius
- $r$  = Radius with origin in center of indenter

The average pressure distribution of a conical indenter is given by equation 26:

$$p_0 = \frac{1}{2} E^* \tan \theta \quad (26)$$

Where:

- $p_0$  = Average pressure conical indenter
- $E^*$  = Equivalent E-modules
- $\theta$  = Penetration angle of conical cone



**Figure 45.** Conical indenter

At last, in figure 46 a spherical indenter is displayed. The normal stress distribution for this indenter shape can be calculated with equation 27.

$$\sigma_{zz}(r, d) = -\frac{2E^*}{\pi R \sqrt{a^2 - r^2}}, \quad r \leq a \quad (27)$$

Where:

- $\sigma_{zz}$  = Stress in perpendicular to surface
- $E^*$  = Equivalent E-modules
- $R$  = Radius of spherical indenter
- $a$  = Penetration radius
- $r$  = Radius with origin in center of indenter

The average pressure distribution of a conical indenter is given by equation 28:

$$p_0 = \frac{4E^* a}{3\pi R} \quad (28)$$

Where:

- $p_0$  = Average pressure flat indenter
- $E^*$  = Equivalent E-modules
- $R$  = Radius of spherical indenter
- $a$  = Penetration radius

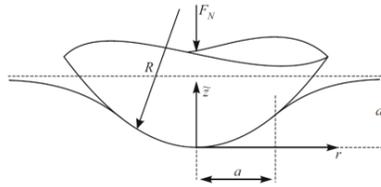


Figure 46. Spherical indenter

### B.1 Matlab Code: Stress distribution

```

1 clear
2
3
4 syms R d a F v1 v2 E1 E2 Edot k(d) r
5 %Input variables
6 %Material properties body 1
7 E1 = 69E9;
8 v1 = 0.3;
9 %Material properties body 2
10 v2 = 0.3;
11 E2 = 69E9;
12
13 %Equivalent E-modules
14 Edot = 1/(((1-v1^2)/E1)+((1-v2^2)/E2))
15
16 %Contact radius
17 a = ((3/4)*((F*R)/Edot))^(1/3)
18
19 %Indentation depth
20 d = -(1/Edot)*(3/2)*(F/(4*a))*(2-(r^2/a^2))
21
22
23 %Contact stiffness
24 k =2*Edot*sqrt(R*d)
25
26 % %Pressure profile
27 % %Max contact pressure for spherical point contact
28 fprintf('Maximale spanning in Mpa')
29 p_max = ((6*F*Edot^2)/(pi^3*R^2))^(1/3)/10^6
30
31 %Max stress vs. F and R
32 fprintf(Maximum stress relation between F and R)
33
34 Flat indenter
35 a = 5 %Radius indenter
36 r = linspace(0.01,4.95,50) %Distance from center of indenter
37
38 ratio = r./a
39
40 for i = 1:length(r)
41     psigma_flat(i) = a/(2*sqrt(a^2-r(i)^2));
42 end
43
44 Conical indenter
45 %Average pressure

```

```

46 for i = 1:1:length(r)
47     psigma_con(i) = acosh(a/r(i));
48 end
49
50 Spherical indenter
51 for i = 1:1:length(r)
52     psigma_sphere(i) = (3*sqrt(a^2-r(i)^2))/(2*a);
53 end
54
55 Plot
56 hold on
57 title('Pressure distribution profile')
58 xlabel('Distance\{it r\} from center w.r.t. indenter peneration radius\{it a\}')
59 ylabel('Relative pressure')
60 grid on
61
62 plot(ratio, psigma_flat, 'LineWidth', 4)
63 plot(ratio, psigma_con, 'LineWidth', 4)
64 plot(ratio, psigma_sphere, 'LineWidth', 4)
65 legend('Flat','Conical', 'Spherical')

```

## B.2 Matlab Code: Maximum stress

```

1 clear
2 %input
3 %Material properties body 1
4 E1 = 69E9; %E-modules body 1
5 v1 = 0.3; %Poisson ratio body 1
6 v2 = 0.3; %Poisson ratio body 2
7 E2 = 69E9; %E-modules body 2
8 F = 0.1; %Normal force
9 R = 0.005; %Radius indenter
10 r = 0.005; %Distance from center
11
12 %Equivalent E-modules
13 Edot = 1/(((1-v1^2)/E1)+((1-v2^2)/E2))
14
15 %Contact radius
16 a = ((3/4)*((F*R)/Edot))^(1/3)
17
18 %Indentation depth
19 d = -(1/Edot)*(3/2)*(F/(4*a))*(2-(r^2/a^2))
20
21 %Contact stiffness
22 k =2*Edot*sqrt(R*d)
23
24 % %Pressure profile
25 % %Max contact pressure for spherical point contact
26 fprintf('Maximale spanning in Mpa')
27 p_max = ((6*F*Edot^2)/(pi^3*R^2))^(1/3)/10^6
28
29 %Max stress vs. F and R
30 fprintf(Maximum stress relation between F and R)
31
32 Rstep = 0.001

```

```

33 Rmin = 0.002
34 Rmax = 0.020
35
36 Fstep = 0.01
37 Fmin = 0.05
38 Fmax = 2
39
40 Rrange = Rmin:Rstep:Rmax;
41 Frange = Fmin:Fstep:Fmax;
42
43 %maximum yield stress for material
44 sf = 1.2
45 p_yield = 200
46
47 for i = 1:1:length(Frange)
48     for j = 1:1:length(Rrange)
49         %Max pressure following Summary of hertz contact area
50         a_summary(i,j) = ((3*Frange(i)*Rrange(j))/(4*Edot))^(1/3);
51         p_max_summary(i,j) = (3*Frange(i))/(2*pi*(a_summary(i,j))^2)/10^6;
52     end
53 end
54
55 fprintf('Results: Max pressure vs. F and R')
56
57 %2D plot of max hertzian stresses. Y—as maximum stresses x—as radius
58 %indenter
59 hold on
60 line1 = 6
61 line2 = 46
62 line3 = 96
63 line4 = 146
64 line5 = 196
65
66 plot(Rrange, p_max_summary(line1,:), 'LineWidth', 2)
67 plot(Rrange, p_max_summary(line2,:), 'LineWidth', 2)
68 plot(Rrange, p_max_summary(line3,:), 'LineWidth', 2)
69 plot(Rrange, p_max_summary(line4,:), 'LineWidth', 2)
70 plot(Rrange, p_max_summary(line5,:), 'LineWidth', 2)
71 xticks(Rmin:0.001:Rmax);
72 xlabel('Radius R [m]')
73 ylabel('Maximum pressure [MPa]')
74 grid on
75 legend
76 fprintf('Normal forces plotted')
77 Fplotted = [Frange(line1), Frange(line2), Frange(line3), Frange(line4), Frange(line5)]
78 legend('F = 0.1N', 'F = 0.5N', 'F = 1N', 'F = 1.5N', 'F = 2N')

```

### C. Wrenchhull

#### C.1 Example of stability check for a 2D object with friction

For a 2D object the wrench hull dimensions is 3D ( $W \in \mathbb{R}^3$ ) and for 3D objects the wrench hull dimensions are 6D ( $W \in \mathbb{R}^6$ ). In figure 48, 49 and 50 the wrench hull is visualized for a 2D shape with friction visualized in figure 47. Here, a friction coefficient of 0.2 is chosen. in vacuum, these friction coefficients will be significantly higher. But, a better representation of the method can be given using a realistic friction coefficient for atmospheric air environments.

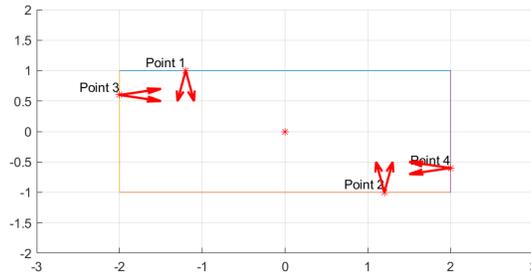


Figure 47. 2D test object and friction cone wrenches

The mathematical expression of equation 18 can also be explained using the figures, since the wrench hull is the convex hull of the wrenches  $w_{i,j}$  (figure 48).

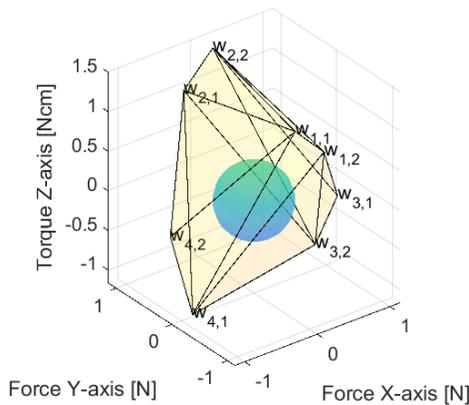


Figure 48. Convex wrench hull for 2D test object with friction

Figure 49 and 50 shows a section view of the total wrench hull of figure 48, here the determination of the largest radius within the wrench hull is clearly visualized. Also, it is clearly visualized that the wrench hull is equal to the convex hull of the wrenches. Since in figure 49 the wrenches  $w_{2,1}$  and  $w_{1,2}$  are inside the convex shape and are not used to construct the wrench hull.

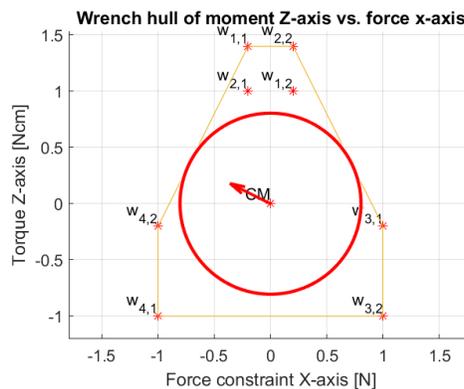
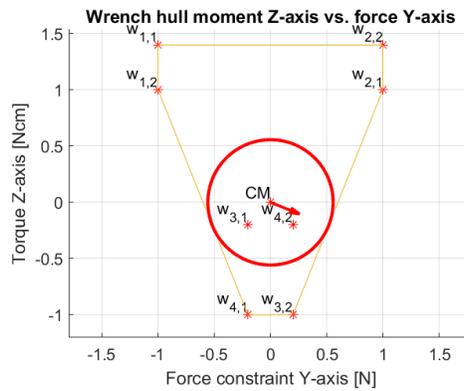


Figure 49. Section view of convex wrench hull: Torque  $z$  vs. constraint forces in X-direction



**Figure 50.** Section view of convex wrench hull: Torque  $z$ -axis vs. constraint forces in  $Y$ -direction

## C.2 Matlab Code: Wrench hull and objective measure - No friction

Function 'Wrenchhull\_2D\_Test' is used in the main 'ICR\_Optimisation' function.

```

1  function [QM3] = Wrenchhull_2D_test(constraint_indices, constraintsx, constraintsy, constraintsz,
2     surfacenormal, CM)
3  %Determine normal vectors of constraints
4  for i = 1:length(constraint_indices)
5     C_normal(i,:) = surfacenormal(constraint_indices(i,:),:);
6  end
7
8  for i = 1:length(constraint_indices)
9     f_n(i,:) = surfacenormal(constraint_indices(i,:),:);
10 end
11
12 d1 = [(constraintsx(1)-CM(1,1)), (constraintsy(1)-CM(1,2)), (constraintsz(1)-CM(1,3))];
13 d2 = [(constraintsx(2)-CM(1,1)), (constraintsy(2)-CM(1,2)), (constraintsz(2)-CM(1,3))];
14 d3 = [(constraintsx(3)-CM(1,1)), (constraintsy(3)-CM(1,2)), (constraintsz(3)-CM(1,3))];
15 d4 = [(constraintsx(4)-CM(1,1)), (constraintsy(4)-CM(1,2)), (constraintsz(4)-CM(1,3))];
16
17 %Row is intersection point Column is XYZ
18 torque1 = cross(d1, C_normal(1,:))';
19 torque2 = cross(d2, C_normal(2,:))';
20 torque3 = cross(d3, C_normal(3,:))';
21 torque4 = cross(d4, C_normal(4,:))';
22
23
24 f_c1 = f_n(1,:)' ;
25 f_c2 = f_n(2,:)' ;
26 f_c3 = f_n(3,:)' ;
27 f_c4 = f_n(4,:)' ;
28
29 %Delete Mx, Mz and Fz. 3D -> 2D
30 f_c1(3) = [];
31 f_c2(3) = [];
32 f_c3(3) = [];
33 f_c4(3) = [];
34
35 for i = 1:1:2
36     torque1(i) = [];
37     torque2(i) = [];
38     torque3(i) = [];

```

```

39     torque4(1) = [];
40 end
41
42 W_1 = [f_c1;torque1];
43 W_2 = [f_c2;torque2];
44 W_3 = [f_c3;torque3];
45 W_4 = [f_c4;torque4];
46 G_w = [W_1, W_2, W_3, W_4];
47
48 for i = 1:1:length(G_w)
49     wrenchhull_Mz_x(i,:) = [G_w(1,i) G_w(3,i)];
50     wrenchhull_Mz_y(i,:) = [G_w(2,i) G_w(3,i)];
51 end
52
53
54 %Constructing Wrench Hull
55 % Only valid in 2D applications!
56 wrenchhull_Mz_xy = [wrenchhull_Mz_x(:,1) wrenchhull_Mz_y(:,1) wrenchhull_Mz_x(:,2)];
57
58 %Determine convex hull of wrench hull
59 [k_Mz_x av_x] = convhull(wrenchhull_Mz_x(:,1), wrenchhull_Mz_x(:,2));
60 [k_Mz_y av_y] = convhull(wrenchhull_Mz_y(:,1), wrenchhull_Mz_y(:,2));
61 [k_Mz_xy av_xy] = convhull(wrenchhull_Mz_xy(:,1), wrenchhull_Mz_xy(:,2), wrenchhull_Mz_xy(:,3));
62
63 % Calculating shortest distance for each edge
64 for i = 1:1:(length(k_Mz_x) - 1)
65     v1 = [wrenchhull_Mz_x(k_Mz_x(i), 1), wrenchhull_Mz_x(k_Mz_x(i), 2), 0];
66     v2 = [wrenchhull_Mz_x(k_Mz_x(i+1), 1), wrenchhull_Mz_x(k_Mz_x(i+1), 2), 0];
67     CM = [0,0,0];
68
69     a = v1 - v2;
70     b = CM - v2;
71     shortest_edge_Mz_x(i) = norm(cross(a,b)) / norm(a);
72 end
73
74 for i = 1:1:(length(k_Mz_y) - 1)
75     v1 = [wrenchhull_Mz_y(k_Mz_y(i), 1), wrenchhull_Mz_y(k_Mz_y(i), 2), 0];
76     v2 = [wrenchhull_Mz_y(k_Mz_y(i+1), 1), wrenchhull_Mz_y(k_Mz_y(i+1), 2), 0];
77     CM = [0,0,0];
78
79     a = v1 - v2;
80     b = CM - v2;
81     shortest_edge_Mz_y(i) = norm(cross(a,b)) / norm(a);
82 end
83
84 %Determine largest ball in 3D volume for Mz_xy
85 for j = 1:1:length(k_Mz_xy)
86     %Make plane from 3 points spanning two vectors
87     startingpoint = [wrenchhull_Mz_xy(k_Mz_xy(j,1),1), wrenchhull_Mz_xy(k_Mz_xy(j,1),2),
88         wrenchhull_Mz_xy(k_Mz_xy(j,1),3)];
89     point2 = [wrenchhull_Mz_xy(k_Mz_xy(j,2),1), wrenchhull_Mz_xy(k_Mz_xy(j,2),2), wrenchhull_Mz_xy
90         (k_Mz_xy(j,2),3)];
91     point3 = [wrenchhull_Mz_xy(k_Mz_xy(j,3),1), wrenchhull_Mz_xy(k_Mz_xy(j,3),2), wrenchhull_Mz_xy
92         (k_Mz_xy(j,3),3)];
93     %Spans two vectors

```

```

91     K1 = point2 - startingpoint;
92     K2 = point3 - startingpoint;
93     %Normal vector of plane
94     K1K2 = cross(K1,K2);
95     K_normal = K1K2 / norm(K1K2);
96     CM = [0,0,0];
97
98     %Vector from CM to a point on the plane
99     CM_startingpoint = CM-startingpoint;
100
101     %Shortest distance of plane
102     dist_Mz_xy(j) = dot(CM_startingpoint, K_normal);
103 end
104
105 %Calculating edge with minimum distance
106 [D_min_Mz_x, D_index_Mz_x] = min(shortest_edge_Mz_x);
107 [D_min_Mz_y, D_index_Mz_y] = min(shortest_edge_Mz_y);
108 [D_min_Mz_xy, D_index_Mz_xy] = min(abs(dist_Mz_xy));
109
110 QM3 = abs(D_min_Mz_xy);
111
112 %Determine orthogonal projection of CM to closest edge
113 V1_shortest_x = [wrenchhull_Mz_x(k_Mz_x(D_index_Mz_x), 1), wrenchhull_Mz_x(k_Mz_x(D_index_Mz_x),
114     2)];
115 V2_shortest_x = [wrenchhull_Mz_x(k_Mz_x(D_index_Mz_x+1), 1), wrenchhull_Mz_x(k_Mz_x(D_index_Mz_x +
116     1), 2)];
117 V_shortest_x = [V1_shortest_x; V2_shortest_x];
118 [ProjPoint_Mz_x] = ProjectPoint(V_shortest_x, [0,0]); %Function projectpoint source: https://nl.
119     mathworks.com/matlabcentral/answers/26464-projecting-a-point-onto-a-line
120
121 V1_shortest_y = [wrenchhull_Mz_y(k_Mz_y(D_index_Mz_y), 1), wrenchhull_Mz_y(k_Mz_y(D_index_Mz_y),
122     2)];
123 V2_shortest_y = [wrenchhull_Mz_y(k_Mz_y(D_index_Mz_y+1), 1), wrenchhull_Mz_y(k_Mz_y(D_index_Mz_y +
124     1), 2)];
125 V_shortest_y = [V1_shortest_y; V2_shortest_y];
126 [ProjPoint_Mz_y] = ProjectPoint(V_shortest_y, [0,0]); %%Function projectpoint source: https://nl.
127     mathworks.com/matlabcentral/answers/26464-projecting-a-point-onto-a-line
128
129 %Disable this part if use is in for loop
130
131 % %Plot wrenchhull Z-axis vs x-axis
132 % figure(2)
133 subplot(2,2,1)
134 hold on
135 plot(wrenchhull_Mz_x(:,1), wrenchhull_Mz_x(:,2), 'r*')
136 %plot all labels
137 labels_wrench = {'w_{1}','w_{2}', 'w_{3}','w_{4}'};
138 text(wrenchhull_Mz_x(:,1), wrenchhull_Mz_x(:,2),labels_wrench,'VerticalAlignment','bottom','
139     HorizontalAlignment','right');
140 plot(0,0, 'r*')
141 label_cm = {'CM'}
142 text(0,0,label_cm,'VerticalAlignment','bottom','HorizontalAlignment','right');
143 %Lay-out graph

```

```

139 grid on
140 axis equal padded
141 title('Wrench hull of moment Z-axis vs. force x-axis')
142 ylabel('Torque Z-axis [Ncm]')
143 xlabel('Force constraint X-axis [N]')
144 %Plot convex hull of points
145 plot(wrenchhull_Mz_x(k_Mz_x, 1), wrenchhull_Mz_x(k_Mz_x, 2))
146
147 %Plot shortest distance arrow
148 quiver(0, 0, ProjPoint_Mz_x(1), ProjPoint_Mz_x(2), 0.5, 'r', 'LineWidth', 1.75, 'MaxHeadSize',
149     1.5)
150 %Plot circle containing
151 viscircles([0,0], D_min_Mz_x);
152 hold off
153
154 %Plot wrenchhull Z-axis vs y-axis
155 % figure(2)
156 subplot(2,2,2)
157 hold on
158 plot(wrenchhull_Mz_y(:,1), wrenchhull_Mz_y(:,2), 'r*')
159 %ot all labels
160 labels_wrench = {'w_{1}', 'w_{2}', 'w_{3}', 'w_{4}'};
161 text(wrenchhull_Mz_y(:,1), wrenchhull_Mz_y(:,2), labels_wrench, 'VerticalAlignment', 'bottom', '
162     HorizontalAlignment', 'right');
163 plot(0,0, 'r*')
164 label_cm = {'CM'}
165 text(0,0, label_cm, 'VerticalAlignment', 'bottom', 'HorizontalAlignment', 'right');
166 %Lay-out graph
167 grid on
168 axis equal padded
169 title('Wrench hull moment Z-axis vs. force Y-axis')
170 ylabel('Torque Z-axis [Ncm]')
171 xlabel('Force constraint Y-axis [N]')
172
173 %Plot convex hull of points
174 plot(wrenchhull_Mz_y(k_Mz_y, 1), wrenchhull_Mz_y(k_Mz_y, 2))
175
176 %Plot shortest distance arrow
177 quiver(0, 0, ProjPoint_Mz_y(1), ProjPoint_Mz_y(2), 0.5, 'r', 'LineWidth', 1.75, 'MaxHeadSize',
178     1.5)
179 %Plot circle containing
180 viscircles([0,0], D_min_Mz_y);
181 hold off
182
183 %Plot Mz XY-axis in one plot
184 %Generate generic sphere coordinates
185 [sphere_x, sphere_y, sphere_z] = sphere;
186 hold on
187 subplot(2,2,3)
188 trisurf(k_Mz_xy, wrenchhull_Mz_xy(:,1), wrenchhull_Mz_xy(:,2), wrenchhull_Mz_xy(:,3), '
189     FaceAlpha', 0.1)
190
191 %plot all labels
192 labels_wrench = {'w_{1}', 'w_{2}', 'w_{3}', 'w_{4}'};

```

```

190     text(wrenchhull_Mz_xy(:,1),wrenchhull_Mz_xy(:,2), wrenchhull_Mz_xy(:,3), labels_wrench);
191
192     %Lay-out graph
193     grid on
194     axis equal padded
195     xlabel('Force X-axis [N]')
196     ylabel('Force Y-axis [N]')
197     zlabel('Torque Z-axis [Ncm]')
198
199     %Plot smallest ball
200     sphere_x_Mz_xy = sphere_x*D_min_Mz_xy;
201     sphere_y_Mz_xy = sphere_y*D_min_Mz_xy;
202     sphere_z_Mz_xy = sphere_z*D_min_Mz_xy;
203     hold on
204     s = surf(sphere_x_Mz_xy, sphere_y_Mz_xy, sphere_z_Mz_xy, 'FaceAlpha',0.5);
205     s.EdgeColor = 'none';
206
207 hold off
208
209
210 end

```

### C.3 Matlab Code: Wrench hull and objective measure - With friction

```

1 clear
2 set(gcf,'Visible','on')
3 syms f_c1 f_c2 f_c3 f_c4 lambda
4
5 lambda = 1
6 mu = 0.2
7
8 %Rectangle contact points of 2,5x2 cm
9 X = linspace(-2,2,6);
10 X2 = -2.* ones(6,1);
11 Y = [1, 1 , 1,1,1,1];
12 Y2 = linspace(-1,1,6);
13
14 %Center of mass
15 CM = [0, 0];
16
17 % figure(1)
18 subplot(2,2,1)
19 hold on
20 grid on
21 plot(X, Y)
22 plot(X, -Y)
23 plot(X2, Y2)
24 plot(-X2, Y2)
25
26 point1 = [X(2),Y(2)]
27 point2 = [X(5),-Y(2)];
28 point3 = [X(1),Y2(5)];
29 point4 = [X(6),Y2(2)];
30
31 pointsx = [point1(1), point2(1), point3(1), point4(1)]

```

```

32 pointsy = [point1(2), point2(2), point3(2), point4(2)]
33
34 plot(point1(1),point1(2),'r*') %point 1
35 plot(point2(1),point2(2),'r*') %point 2
36 plot(point3(1),point3(2),'r*') %point 3
37 plot(point4(1),point4(2),'r*') %point 4
38
39 %Plot unitvectors
40 labels = {'Point 1','Point 2','Point 3', 'Point 4'};
41 text(pointsx, pointsy,labels,'VerticalAlignment','bottom','HorizontalAlignment','right')
42
43 plot(CM(1),CM(2),'r*')
44 xlim([-3 3])
45 ylim([-2 2])
46
47 %Wrench basis No-friction
48 %Point 1
49 F_c1 = [-mu, mu;
50         -1, -1]
51 d_1 = [(point1(1)-CM(1)); point1(2)-CM(2)]
52
53 torque11 = d_1(1)*F_c1(2,1) - d_1(2)*F_c1(1,1)
54 torque12 = d_1(1)*F_c1(2,2) - d_1(2)*F_c1(1,2)
55
56 cross1 = [torque11, torque12]
57 Wc_1 = [F_c1;
58         lambda*cross1]
59
60 %Wrench Point 2
61 F_c2 = [-mu, mu;
62         1, 1]
63
64 d_2 = [(point2(1)-CM(1)); point2(2)-CM(2)]
65 torque21 = d_2(1)*F_c2(2,1) - d_2(2)*F_c2(1,1)
66 torque22 = d_2(1)*F_c2(2,2) - d_2(2)*F_c2(1,2)
67 cross2 = [torque21, torque22]
68 Wc_2 = [F_c2;
69         lambda*cross2]
70
71 %Wrench Point 3
72 F_c3 = [1 1;
73         -mu mu]
74 d_3 = [(point3(1)-CM(1)); point3(2)-CM(2)]
75 torque31 = d_3(1)*F_c3(2,1) - d_3(2)*F_c3(1,1)
76 torque32 = d_3(1)*F_c3(2,2) - d_3(2)*F_c3(1,2)
77 cross3 = [torque31, torque32]
78
79 Wc_3 = [F_c3;
80         lambda*cross3]
81
82 %Wrench point 4
83 F_c4 = [-1 -1;
84         -mu mu]
85 d_4 = [(point4(1)-CM(1)); point4(2)-CM(2)]
86 torque41 = d_4(1)*F_c4(2,1) - d_4(2)*F_c4(1,1)

```

```

87 torque42 = d_4(1)*F_c4(2,2) - d_4(2)*F_c4(1,2)
88 cross4 = [torque41, torque42]
89
90 Wc_4 = [F_c4;
91         lambda*cross4]
92
93 for i = 1:1:length(F_c1)
94     X1(i) = point1(1);
95     Y1(i) = point1(2);
96     Xl2(i) = point2(1);
97     Yl2(i) = point2(2);
98     Xl3(i) = point3(1);
99     Yl3(i) = point3(2);
100    Xl4(i) = point4(1);
101    Yl4(i) = point4(2);
102 end
103 %Plot unitvectors
104 quiver(X1, Y1, F_c1(1,:), F_c1(2,:), 0.5, 'r', 'LineWidth', 1.75, 'MaxHeadSize', 1.5)
105 quiver(Xl2, Yl2, F_c2(1,:), F_c2(2,:), 0.5, 'r', 'LineWidth', 1.75, 'MaxHeadSize', 1.5)
106 quiver(Xl3, Yl3, F_c3(1,:), F_c3(2,:), 0.5, 'r', 'LineWidth', 1.75, 'MaxHeadSize', 1.5)
107 quiver(Xl4, Yl4, F_c4(1,:), F_c4(2,:), 0.5, 'r', 'LineWidth', 1.75, 'MaxHeadSize', 1.5)
108 hold off
109
110 % Set up Grasp map
111 G_w = [[Wc_1], [Wc_2], [Wc_3], [Wc_4]]
112 x = [f_c1 f_c2 f_c3 f_c4]
113
114 %Constructing Wrench Hull
115 % Only valid in 2D applications!
116
117 %Pick out generated torques and x-y data for each point
118 % x-axis
119 % y-axis is torque in Z-axis
120
121 %First torque z-axis vs y-axis
122 d = [d_1 d_2 d_3 d_4] %all position vectors
123
124 for i = 1:1:length(G_w)
125     wrenchhull_Mz_x(i,:) = [G_w(1,i) G_w(3,i)]
126     wrenchhull_Mz_y(i,:) = [G_w(2,i) G_w(3,i)]
127 end
128
129 wrenchhull_Mz_xy = [wrenchhull_Mz_x(:,1) wrenchhull_Mz_y(:,1) wrenchhull_Mz_x(:,2)]
130
131 %Determine convex hull of wrench hull
132 [k_Mz_x av_x] = convhull(wrenchhull_Mz_x(:,1), wrenchhull_Mz_x(:,2));
133 [k_Mz_y av_y] = convhull(wrenchhull_Mz_y(:,1), wrenchhull_Mz_y(:,2));
134 [k_Mz_xy av_xy] = convhull(wrenchhull_Mz_xy(:,1), wrenchhull_Mz_xy(:,2), wrenchhull_Mz_xy(:,3));
135
136 %Calculating shortest distance for each edge
137 for i = 1:(length(k_Mz_x) - 1)
138     v1 = [wrenchhull_Mz_x(k_Mz_x(i), 1), wrenchhull_Mz_x(k_Mz_x(i), 2), 0];
139     v2 = [wrenchhull_Mz_x(k_Mz_x(i+1), 1), wrenchhull_Mz_x(k_Mz_x(i+1), 2), 0];
140     CM = [0,0,0];
141

```

```

142     a = v1 - v2;
143     b = CM - v2;
144     shortest_edge_Mz_x(i) = norm(cross(a,b)) / norm(a);
145 end
146
147 for i = 1:1:(length(k_Mz_y) - 1)
148     v1 = [wrenchhull_Mz_y(k_Mz_y(i), 1), wrenchhull_Mz_y(k_Mz_y(i), 2), 0];
149     v2 = [wrenchhull_Mz_y(k_Mz_y(i+1), 1), wrenchhull_Mz_y(k_Mz_y(i+1), 2), 0];
150     CM = [0,0,0];
151
152     a = v1 - v2;
153     b = CM - v2;
154     shortest_edge_Mz_y(i) = norm(cross(a,b)) / norm(a);
155 end
156
157 %Determine largest ball in 3D volume for Mz_xy
158 for j = 1:1:length(k_Mz_xy)
159     %Make plane from 3 points spanning two vectors
160     startingpoint = [wrenchhull_Mz_xy(k_Mz_xy(j,1),1), wrenchhull_Mz_xy(k_Mz_xy(j,1),2),
161         wrenchhull_Mz_xy(k_Mz_xy(j,1),3)];
162     point2 = [wrenchhull_Mz_xy(k_Mz_xy(j,2),1), wrenchhull_Mz_xy(k_Mz_xy(j,2),2), wrenchhull_Mz_xy
163         (k_Mz_xy(j,2),3)];
164     point3 = [wrenchhull_Mz_xy(k_Mz_xy(j,3),1), wrenchhull_Mz_xy(k_Mz_xy(j,3),2), wrenchhull_Mz_xy
165         (k_Mz_xy(j,3),3)];
166     %Spans two vectors
167     K1 = point2 - startingpoint;
168     K2 = point3 - startingpoint;
169     %Normal vector of plane
170     K1K2 = cross(K1,K2);
171     K_normal = K1K2 / norm(K1K2);
172     CM = [0,0,0];
173
174     %Vector from CM to a point on the plane
175     CM_startingpoint = CM-startingpoint;
176
177     %Shortest distance of plane
178     dist_Mz_xy(j) = dot(CM_startingpoint, K_normal);
179 end
180
181 %Calculating edge with minimum distance
182 [D_min_Mz_x, D_index_Mz_x] = min(shortest_edge_Mz_x)
183 [D_min_Mz_y, D_index_Mz_y] = min(shortest_edge_Mz_y)
184 [D_min_Mz_xy, D_index_Mz_xy] = min(abs(dist_Mz_xy));
185
186 %Determine orthogonal projection of CM to closest edge
187 V1_shortest_x = [wrenchhull_Mz_x(k_Mz_x(D_index_Mz_x), 1), wrenchhull_Mz_x(k_Mz_x(D_index_Mz_x),
188     2)]
189 V2_shortest_x = [wrenchhull_Mz_x(k_Mz_x(D_index_Mz_x+1), 1), wrenchhull_Mz_x(k_Mz_x(D_index_Mz_x +
190     1), 2)]
191 V_shortest_x = [V1_shortest_x; V2_shortest_x]
192 [ProjPoint_Mz_x] = ProjectPoint(V_shortest_x, [0,0]) %Function projectpoint source: https://nl.mathworks.com/matlabcentral/answers/26464-projecting-a-point-onto-a-line
193
194 V1_shortest_y = [wrenchhull_Mz_y(k_Mz_y(D_index_Mz_y), 1), wrenchhull_Mz_y(k_Mz_y(D_index_Mz_y),
195     2)]

```

```

190 V2_shortest_y = [wrenchhull_Mz_y(k_Mz_y(D_index_Mz_y+1), 1), wrenchhull_Mz_y(k_Mz_y(D_index_Mz_y +
191     1), 2)]
191 V_shortest_y = [V1_shortest_y; V2_shortest_y]
192 [ProjPoint_Mz_y] = ProjectPoint(V_shortest_y, [0,0]) %%Function projectpoint source: https://nl.
193     mathworks.com/matlabcentral/answers/26464-projecting-a-point-onto-a-line
194 %%Plot wrenchhull Z-axis vs x-axis
195 % figure(2)
196 subplot(2,2,2)
197 hold on
198 plot(wrenchhull_Mz_x(:,1), wrenchhull_Mz_x(:,2), 'r*')
199 %plot all labels
200 labels_wrench = {'w_{1,1}', 'w_{1,2}', 'w_{2,1}', 'w_{2,2}', 'w_{3,1}', 'w_{3,2}', 'w_{4,1}', 'w_
201     {4,2}',};
202 text(wrenchhull_Mz_x(:,1), wrenchhull_Mz_x(:,2), labels_wrench, 'VerticalAlignment', 'bottom', '
203     HorizontalAlignment', 'right');
204 plot(0,0, 'r*')
205 label_cm = {'CM'}
206 text(0,0, label_cm, 'VerticalAlignment', 'bottom', 'HorizontalAlignment', 'right');
207 %Lay-out graph
208 grid on
209 axis equal padded
210 title('Wrench hull of moment Z-axis vs. force x-axis')
211 ylabel('Torque Z-axis [Ncm]')
212 xlabel('Force constraint X-axis [N]')
213 %Plot convex hull of points
214 plot(wrenchhull_Mz_x(k_Mz_x, 1), wrenchhull_Mz_x(k_Mz_x, 2))
215 %Plot shortest distance arrow
216 quiver(0, 0, ProjPoint_Mz_x(1), ProjPoint_Mz_x(2), 0.5, 'r', 'LineWidth', 1.75, 'MaxHeadSize',
217     1.5)
218 %Plot circle containing
219 viscircles([0,0], D_min_Mz_x);
220 hold off
221 %%Plot wrenchhull Z-axis vs y-axis
222 % figure(2)
223 subplot(2,2,3)
224 hold on
225 plot(wrenchhull_Mz_y(:,1), wrenchhull_Mz_y(:,2), 'r*')
226 %ot all labels
227 labels_wrench = {'w_{1,1}', 'w_{1,2}', 'w_{2,1}', 'w_{2,2}', 'w_{3,1}', 'w_{3,2}', 'w_{4,1}', 'w_
228     {4,2}',};
229 text(wrenchhull_Mz_y(:,1), wrenchhull_Mz_y(:,2), labels_wrench, 'VerticalAlignment', 'bottom', '
230     HorizontalAlignment', 'right');
231 plot(0,0, 'r*')
232 label_cm = {'CM'}
233 text(0,0, label_cm, 'VerticalAlignment', 'bottom', 'HorizontalAlignment', 'right');
234 %Lay-out graph
235 grid on
236 axis equal padded
237 title('Wrench hull moment Z-axis vs. force Y-axis')
238 ylabel('Torque Z-axis [Ncm]')
239 xlabel('Force constraint Y-axis [N]')

```

```

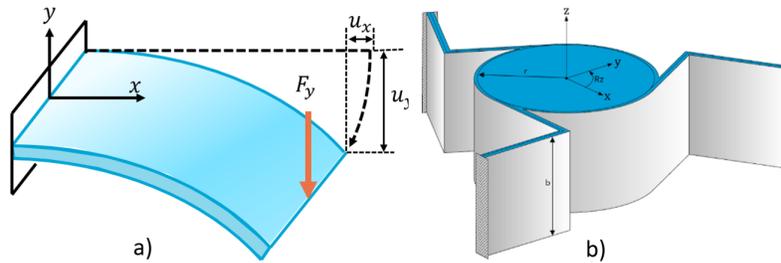
238
239 %Plot convex hull of points
240 plot(wrenchhull_Mz_y(k_Mz_y, 1), wrenchhull_Mz_y(k_Mz_y, 2))
241
242 %Plot shortest distance arrow
243 quiver(0, 0, ProjPoint_Mz_y(1), ProjPoint_Mz_y(2), 0.5, 'r', 'LineWidth', 1.75, 'MaxHeadSize',
244       1.5)
245 %Plot circle containing
246 viscircles([0,0], D_min_Mz_y);
247 hold off
248
249 %Plot Mz XY-axis in one plot
250 %Generate generic sphere coordinates
251 [sphere_x, sphere_y, sphere_z] = sphere;
252 hold on
253 subplot(2,2,4)
254 trisurf(k_Mz_xy, wrenchhull_Mz_xy(:,1), wrenchhull_Mz_xy(:,2), wrenchhull_Mz_xy(:,3), '
255       FaceAlpha', 0.1)
256
257 %plot all labels
258 labels_wrench = {'w_{1,1}', 'w_{1,2}', 'w_{2,1}', 'w_{2,2}', 'w_{3,1}', 'w_{3,2}', 'w_{4,1}', 'w_
259               {4,2}'};
260 text(wrenchhull_Mz_xy(:,1), wrenchhull_Mz_xy(:,2), wrenchhull_Mz_xy(:,3), labels_wrench);
261 % plot3(0,0,0, 'r*')
262 % % label_cm = {'CM'}
263 % text(0,0,label_cm, 'VerticalAlignment', 'bottom', 'HorizontalAlignment', 'right');
264
265 %Lay-out graph
266 grid on
267 axis equal padded
268 xlabel('Force X-axis [N]')
269 ylabel('Force Y-axis [N]')
270 zlabel('Torque Z-axis [Ncm]')
271
272 %Plot smallest ball
273 sphere_x_Mz_xy = sphere_x*D_min_Mz_xy;
274 sphere_y_Mz_xy = sphere_y*D_min_Mz_xy;
275 sphere_z_Mz_xy = sphere_z*D_min_Mz_xy;
276 hold on
277 s = surf(sphere_x_Mz_xy, sphere_y_Mz_xy, sphere_z_Mz_xy, 'FaceAlpha', 0.5);
278 s.EdgeColor = 'none';
279 hold off

```

## D. Casestudy suspension

### D.1 Suspension setup

For the suspension setup two building blocks are considered (figure 51). The required dimensions is for both the leaf spring and folded leaf spring determined.

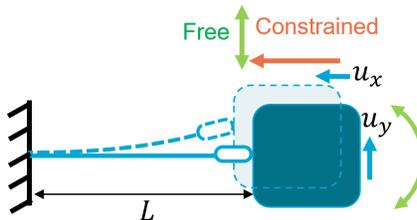


**Figure 51.** Example of suspension building blocks to constraint a) Leaf spring to constraint single translation or in a combination of multiple leaf springs to constraint  $\delta_x$ ,  $\delta_y$  and  $M_z$ . b) Combination of three folded leaf springs to constrain  $\delta_z$ ,  $M_x$  and  $M_y$  [38]

For a single-clamped leaf spring, figure 52, the stiffness can be calculated using Euler-Bernoulli Beam theory [39] using equation 29).

$$k_y = \frac{3EI_y}{L^3} \tag{29}$$

- $k_y$  = Stiffness of spring in y-axis
- $I_x$  = Moment of inertia around x-axis
- $L$  = Length of leaf spring
- $E$  = E-modules



**Figure 52.** Single clamped spring; Constrains beam and corresponding deflection curve.

The moment of inertia around the x for a rectangular beam can be determined using equation 30.

$$I_x = \frac{bh^3}{12} \tag{30}$$

Where:

- $I_x$  = Moment of inertia around x-axis
- $h$  = Height of rectangular profile
- $b$  = width of rectangular profile

The second building block consists of three folded-leaf springs connected in parallel. The radial stiffness can be determined using equation 31 [38].

$$k_{xy} = \frac{45EI_x^3}{L^3} \tag{31}$$

Where:

- $k_{xy}$  = Radial stiffness in  $XY$ -plane
- $I_x$  = Moment of inertia around  $x$ -axis
- $L$  = Length of leaf spring
- $E$  = E-modules

The deflection of the suspension is determined with equation 32.

$$u_y = \frac{F_N}{k} \tag{32}$$

Where:

- $u$  = Deflection of suspension
- $F_N$  = Applied nesting force
- $k_y$  = Stiffness suspension

Since the suspension consists of a single leaf spring, during its stroke there is a parasitic motion present ( $u_x$  in figure 52). This parasitic motion is calculated using equation 33 [40].

$$u_x = \frac{3u_y}{5L} \tag{33}$$

### D.2 Minimum required nesting force

The normal force on all constraints is determined based on the minimal required nesting force. The nesting force is dependent on the minimum required actuation force to obtain the minimum required stroke. The stroke is dependent on the stiffness of the suspension, a lower stiffness results in a decrease in the required actuation force. Second, the required stroke is dependent on the positioning error and extra offset margin. This all creates a feedback loop where the goal is to obtain as low as possible stiffness.

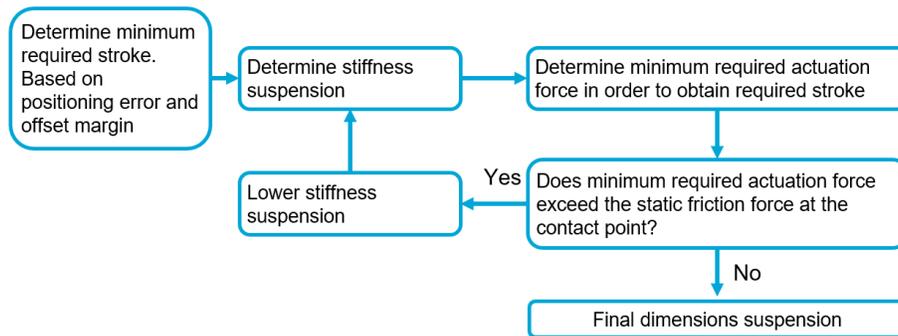
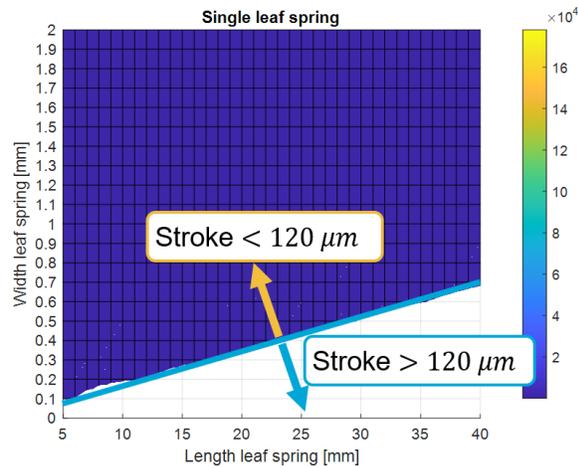


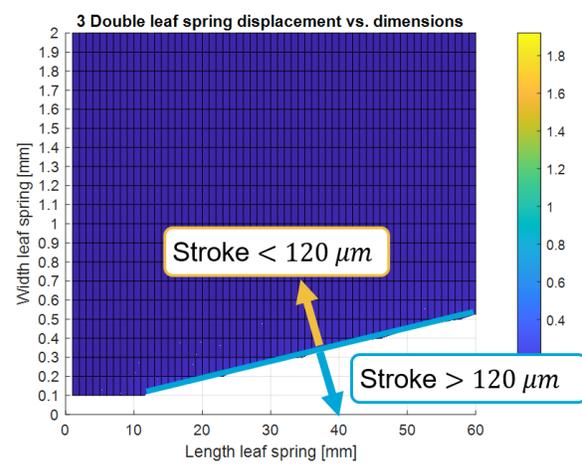
Figure 53. Flowchart determining suspension dimensions

### D.3 Suspension results

In figure 54 and 55 the results are plotted that is used to determine the dimensions of the suspension. The diagonal lines indicate the exact solution for a suspension setup that is able to obtain the required stroke with the set actuation force. Here, an actuation force of  $F_N = 0.1N$  is used and the required stroke is equal to  $120\mu m$ . The required stroke is based upon the positioning error and offset margin, displayed in figures 41 and 42.



**Figure 54.** Obtained deflection by a actuation force of 0.1N versus leaf spring dimensions.



**Figure 55.** Obtained deflection by a actuation force of 0.1N versus folded leaf spring dimensions.

#### D.4 Matlab code: Leaf spring dimensions

```

1 clear
2
3 %Inputs
4 E = 69E3
5 F = 0.1
6 b = 10
7
8 tstep = 0.1;
9 tmin = 0.1;
10 tmax = 2;
11
12 Lstep = 1;
13 Lmin = 5;
14 Lmax = 40;
15
16 dstep = 0.2;
17 dmin = 1;
18 dmax = 3;
19
20 trange = tmin:tstep:tmax;
21 Lrange = Lmin:Lstep:Lmax;
22 drange = dmin:dstep:dmax;
23
24 for i = 1:length(trange)
25     for j = 1:length(Lrange)
26         Ileaf = (1/12)*b*trange(i)^3; %Moment of inertia rod
27         kleaf = (3*E*Ileaf)/(Lrange(j))^3; %Stiffness for leafspring
28         u_leaf(i,j) = F/kleaf;
29         u_leaf_um(i,j) = u_leaf(i,j)*1000;
30
31         sigma_leaf(i,j) = (F*Lrange(j)*u_leaf(i,j))/Ileaf;
32     end
33 end
34
35 fprintf('Results: Folded leaf spring setup')
36 [plotx,ploty] = meshgrid(Lmin:Lstep:Lmax, tmin:tstep:tmax);

```

```

37 surf(plotx, ploty, u_leaf_um, sigma_leaf)
38 grid on
39 yticks(0:tstep:tmax);
40 title('Single leaf spring')
41 ylabel('Width leaf spring [mm]')
42 xlabel('Length leaf spring [mm]')
43 zlabel('Displacement [\mum]')
44 colorbar
45 % xlim([0 40])
46 % ylim([0 2])
47 zlim([0 120])
48 % caxis([0 55]);

```

### D.5 Matlab code: folded leaf spring dimensions

```

1 clear
2
3 F = 0.1
4 h = 10
5 E = 69E3
6
7 tstep = 0.1;
8 tmin = 0.1;
9 tmax = 2;
10
11 Lstep = 1;
12 Lmin = 1;
13 Lmax = 60;
14
15 trange = tmin:tstep:tmax;
16 Lrange = Lmin:Lstep:Lmax;
17
18 for i = 1:1:length(trange)
19     for j = 1:1:length(Lrange)
20
21         I = (1/12)*h*trange(i)^3;
22
23         C = (15/2)*(E*I)/Lrange(j)^3;
24         C_radial = 3*C;
25
26         d_r(i,j) = F/C_radial;
27         d_r_um(i,j) = d_r(i,j)*1000;
28
29         sigma_max(i,j) = ((d_r_um(i,j)*E*trange(i))/Lrange(j))/1E6;
30     end
31 end
32
33 fprintf('Results: Folded leaf spring setup')
34 [plotx,ploty] = meshgrid(Lmin:Lstep:Lmax, tmin:tstep:tmax);
35 surf(plotx, ploty, d_r_um, sigma_max)
36 grid on
37 yticks(0:tstep:tmax);
38 title('3 Double leaf spring displacement vs. dimensions')
39 ylabel('Width leaf spring [mm]')
40 xlabel('Length leaf spring [mm]')

```

```
41 xlabel('Displacement [\mum]')
42 colorbar
43 % xlim([0 80])
44 % ylim([0 2])
45 zlim([0 120])
46 % caxis([0 55]);
```

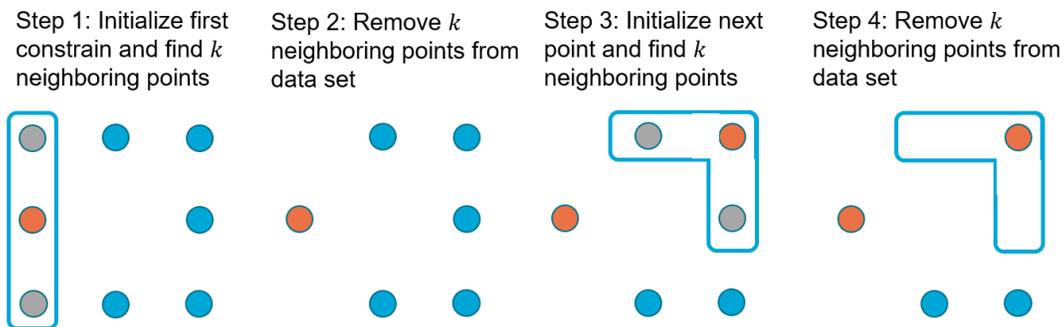
## E. Algorithm execution time optimisation

### E.1 Run-time optimization

The optimization algorithm uses a brute force method to calculate the best grasp and is able to calculate 67000 iterations per minute. The runtime of the algorithm scales with  $O(n^k)$ ,  $k$  is equal to the number of included contact points in the optimization algorithm. For a test object with 300 datapoints where 7 contact points are optimized the runtime scales with  $O(n^7)$ . Resulting in a computational extreme long runtime of  $t = 300^7/67000 = 2.27E9$  days.

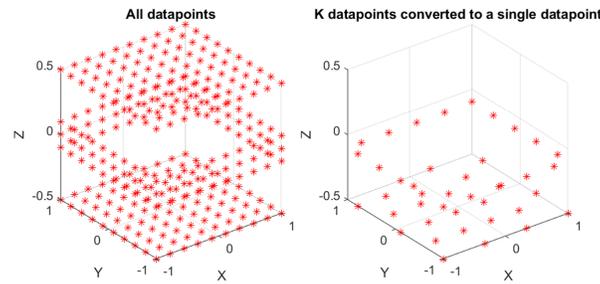
Numerous assumptions and methods are used to shorten the runtime, while not limiting the performance too much.

1. There is a minimum of three constraints imposed on the bottom of the object, these will constrain  $\delta_Z, M_X$  and  $M_Y$ . If the normal forces originating from the three constraints are not in the vertical direction it will create a slope where the object can slide on, the sliding will generate particles and need to be prevented. The following limitation is added to prevent sliding between platform and object:  
*Three constraints imposed by the movable platform should always have the direction equal to  $\mathbf{n} = [0\ 0\ 1]$ , resulting in a normal force acting in the vertical direction.*
2. The object is placed with a translational positioning error in the  $XY$ -plane and rotational error around the  $Z$ -axis. Four constraints located at the sides of the object are used to restore the positioning error. A surface normal force acting in the vertical axis ( $\mathbf{n} = [0\ 0\ \pm 1]$ ) will not deliver a usable force to restore the positioning error. Therefore, a limitation is set in what surfaces are used as input.  
*Object surfaces with normal vectors acting between  $[\sim \sim \pm 0.8] \leq \mathbf{n} \leq [\sim \sim \pm 1]$  are excluded from the input data set.*
3. Continuing on the previous assumptions. Following assumption 1, the three constraints acting on the bottom of the object constraining  $\delta_Z, M_X$  and  $M_Y$ , need to act in the positive  $z$ -axis ( $\mathbf{n} = [0\ 0\ 1]$ ). Therefore, these contact points can be excluded following assumption 2.  
*The three contact points imposed at the bottom of the object constraining  $\delta_Z, M_X$  and  $M_Y$ , are excluded from the optimization algorithm*
4. The stability check algorithm calculates what the least resisted torque can be with respect to the object's center of mass. The check favors contact point positions that can generate high torques with respect to the center of mass. The three constraints imposed on the bottom of the object are excluded from the optimization algorithm, as stated by assumption 3. The position of these constraints can be determined by a fixed guideline:  
*The three constraints imposed at the bottom of the object, constraining  $\delta_Z, M_X$  and  $M_Y$ , should be positioned in a star configuration and placed as far away from the Center of Mass.*
5. Reducing the resolution of the data set will decrease the number of points to iterate over. To decrease the number of points in the data set  $k$  neighboring points are converted to one single point, the converting algorithm is shown in figure 56. The overall best grasp will probably lay close to the best solution grasp found with the reduced data set.



**Figure 56.** Algorithm remove neighboring points

To visualize the assumptions given above the test object's pointcloud of the object is displayed in figure 57 using the original data set and reduced data set. Using all assumptions above the number of data points for the test object is decreased from 300 to 41 points where  $k = 9$ . The algorithm has to iterate over 41 data points and 4 constraints, this results in a computational time of  $41^4/67000 = 43$  minutes.



**Figure 57.** Original data set and reduced data set for the test object for  $k = 9$

## E.2 Pointcloud orientation

Objects are converted from a 3D-CAD model to a pointcloud containing points with XYZ-coordinates data. The pointcloud orientation and origin should be adapted for easier implementation in all other Matlab scripts used in this study. First it should be noticed that the middle of the object is not in the origin, this can be adapted by a shift in XYZ-datapoints:

$$\mathbf{u}_s = \mathbf{u} + \mathbf{u}_o = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} \quad (34)$$

Where:

$\mathbf{u}_s$  = Shifted datapoints

$\mathbf{u}$  = Original dataset imported from 3D-CAD model

$\mathbf{u}_o$  = Offset

Second, it is preferred to have pointcloud oriented so that the z-axis represents the object's height. The data points need to be rotated, this action is performed by the rotation matrixes:

$${}^N C_f(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (35)$$

Where:

${}^N C_f(\phi)$  = Rotation matrix with x-axis as rotation axis

$\phi$  = Angle of rotation

$${}^N C_f(\psi) = \begin{bmatrix} \cos \psi & 0 & \sin \psi \\ 0 & 1 & 0 \\ -\sin \psi & 0 & \cos \psi \end{bmatrix} \quad (36)$$

Where:

${}^N C_f(\psi)$  = Rotation matrix with y-axis as rotation axis

$\psi$  = Angle of rotation

$${}^N C_f(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (37)$$

Where:

${}^N C_f(\theta)$  = Rotation matrix with Z-axis as rotation axis

$\theta$  = Angle of rotation

The shifted data points ( $\mathbf{u}_s$ ) are multiplied by all rotation matrixes in order to rotate the entire dataset from its original frame  $F$  to the rotated frame  $N$ .

$${}^N\mathbf{u}_s = {}^N\mathbf{C}_B(\theta) {}^B\mathbf{C}_G(\psi) {}^G\mathbf{C}_F(\phi) {}^F\mathbf{u}_s \quad (38)$$

Where:

- ${}^F\mathbf{u}_s$  = Shifted data points in the original frame  $F$
- ${}^N\mathbf{u}_s$  = Shifted and rotated data points in the rotated frame  $N$
- ${}^N\mathbf{C}_B(\theta)$  = Rotation matrix with Z-axis as rotation axis
- ${}^B\mathbf{C}_G(\psi)$  = Rotation matrix with Y-axis as rotation axis
- ${}^G\mathbf{C}_F(\phi)$  = Rotation matrix with X-axis as rotation axis

### E.3 Shift and rotate pointcloud

```

1 clear
2
3 %LOAD Pointcloud XYZ datapoints nxm matrix. n is each point. m corresponds
4 %to XYZ data
5 XYZdatapoints = readmatrix('Rectangularblock.xlsx');
6 transform_k = 9; %Select how many points will be transformed to a single point
7 k_normal = 5; %Select how many points are used to calculate normal vector
8
9 %Variables
10 CM = [0, 0, 0];
11
12 ptCloud = pointCloud(XYZdatapoints);
13 surfacenormal = pcnormals(ptCloud, k_normal);
14 ptCloudpoints = [ptCloud.Location(:,1), ptCloud.Location(:,2), ptCloud.Location(:,3)];
15
16 % Input rotational correction
17 theta_x = 90;
18 theta_y = 0;
19 theta_z = 0;
20
21 % Input translational correction
22 X_shift = -9;
23 Y_shift = 9;
24 Z_shift = -0.35;
25
26 %Setting up rotation matrices
27 C_x = [1 0 0; ...
28        0 cosd(theta_x) -sind(theta_x);...
29        0 sind(theta_x) cosd(theta_x)];
30
31 C_y = [cosd(theta_y) 0 sind(theta_y); ...
32        0 1 0;...
33        -sind(theta_y) 0 cosd(theta_y)];
34
35 C_z = [cosd(theta_z) -sind(theta_z) 0; ...
36        sind(theta_z) cosd(theta_z) 0;...
37        0 0 1];
38
39 %Applying shift and rotation matrices
40 XYZPointsRotation = C_x*C_y*C_z*ptCloudpoints'
```

```

41 surfaceNormal_rotation = C_x*C_y*C_z*surfacenormal'
42 XYZPointsRotation = XYZPointsRotation'
43 surfaceNormal_rotation = surfaceNormal_rotation'
44 XYZPointsRotationShifted(:,1) = XYZPointsRotation(:,1) + X_shift
45 XYZPointsRotationShifted(:,2) = XYZPointsRotation(:,2) + Y_shift
46 XYZPointsRotationShifted(:,3) = XYZPointsRotation(:,3) + Z_shift
47
48 %Saving results to .mat file
49 save('XYZdatapoints_checked.mat','XYZPointsRotationShifted');
50
51 %Plotting results
52 view(3)
53 hold on;
54 grid on;
55 % title('Original dataset')
56 xlabel('X');
57 ylabel('Y');
58 zlabel('Z');
59 plot3(XYZPointsRotationShifted(:,1), XYZPointsRotationShifted(:,2), XYZPointsRotationShifted(:,3),
      'r*');

```

#### E.4 Main function to exclude datapoints and lower resolution

```

1 clear
2
3 %LOAD Pointcloud XYZ datapoints nxm matrix. n is each point. m corresponds
4 %to XYZ data
5 XYZdatapoints_checked = matfile('XYZdatapoints_checked.mat');
6 XYZdatapoints = XYZdatapoints_checked.XYZPointsRotationShifted;
7
8 %Select how many points will be used to determine normal vector
9 k_normal = 6;
10 %Determine maximum and minimum range for Z-axis from which points can be
11 %included
12 zmax = 0.1;
13 zmin = -0.1;
14 %Determine resolution scale factor. K neighbouring points will be reduced to a single point
15 transform_k = 20;
16
17 %Variables
18 CM = [0, 0, 0]; %Center of mass
19
20 ptCloud = pointCloud(XYZdatapoints);
21 surfacenormal = pcnormals(ptCloud, k_normal);
22 ptCloudpoints = [ptCloud.Location(:,1), ptCloud.Location(:,2), ptCloud.Location(:,3)];
23
24
25 Normal vectors and pointcloud reduction
26 %Test if normal vectors are inward or outward. Source: From mathworks.com pcnormal
27 %function page
28 for k = 1:numel(ptCloudpoints(:,1))
29     p1 = CM - [ptCloudpoints(k,1),ptCloudpoints(k,2),ptCloudpoints(k,3)];
30     p2 = [surfacenormal(k,1),surfacenormal(k,2),surfacenormal(k,3)];
31     % Flip the normal vector if it is not pointing towards the sensor.
32     angle = atan2(norm(cross(p1,p2)),p1*p2');

```

```

33     if angle > pi/2 || angle < -pi/2
34         surfacenormal(k,1) = -surfacenormal(k,1);
35         surfacenormal(k,2) = -surfacenormal(k,2);
36         surfacenormal(k,3) = -surfacenormal(k,3);
37     end
38 end
39
40 Plot normal vectors
41 subplot(2,2,1)
42 view(3)
43 hold on;
44 grid on;
45 title('Original dataset with normal vectors')
46 xlabel('X');
47 ylabel('Y');
48 zlabel('Z');
49 plot3(ptCloudpoints(:,1),ptCloudpoints(:,2), ptCloudpoints(:,3), 'r*');
50
51 einde = size(ptCloudpoints,1)
52 x = ptCloudpoints(1:1:einde,1);
53 y = ptCloudpoints(1:1:einde,2);
54 z = ptCloudpoints(1:1:einde,3);
55 u = surfacenormal(1:1:einde,1);
56 v = surfacenormal(1:1:einde,2);
57 w = surfacenormal(1:1:einde,3);
58
59 quiver3(x,y,z,u,v,w);
60 hold off
61
62
63
64 PointCloud reduction
65 tic
66 %Function iterationpointcloudV2 reduces the number of datapoints. Output:
67 %iterationcloud are XYZ datapoints; Check_indices are the indices of these
68 %datapoints in the original dataset ptCloudpoints
69 [iterationcloud, check_indices] = iterationpointcloudV2(ptCloud, ptCloudpoints, zmax, zmin,
70     transform_k, surfacenormal);
71
72 check_indices = check_indices';
73 toc
74
75 save('XYZdatapoints_reduced.mat', 'iterationcloud', 'check_indices');
76 subplot(2,2,2)
77 view(3)
78 hold on;
79 grid on;
80 title('Shifted dataset')
81 xlabel('X');
82 ylabel('Y');
83 zlabel('Z');
84 plot3(ptCloudpoints(:,1), ptCloudpoints(:,2), ptCloudpoints(:,3), 'r*');
85
86 subplot(2,2,3)
87 view(3)

```

```

87 hold on;
88 grid on;
89 % title('Reduced dataset')
90 xlabel('X');
91 ylabel('Y');
92 zlabel('Z');
93 plot3(iterationcloud(:,1), iterationcloud(:,2), iterationcloud(:,3), 'r*');

```

### E.5 Function IterationPointCloudV2: Removes datapoints and lowers resolution

```

1 function [iterationcloud, check_indices] = iterationpointcloudV2(ptCloud, ptCloudpoints, zmax,
2   zmin, k, surfacenormal)
3 % Exclude points with normal vector n = [~ ~ ..] and n = [~ ~ ..]
4 iterationrownumber = 1
5 for i = 1:1:size(surfacenormal,1)
6     if surfacenormal(i,3) > -0.8 && surfacenormal(i,3) < 0.8 && ptCloudpoints(i,3) < zmax &&
7       ptCloudpoints(i,3) > zmin
8         %Save points that need to be included
9         iterationCloud_indices(iterationrownumber) = [i];
10        iterationrownumber = iterationrownumber + 1;
11    end
12 end
13 %Lower resolution of pointcloud
14 iterationCloud_indices = iterationCloud_indices';
15 iterationcloud(1,:) = ptCloudpoints(iterationCloud_indices(1),:);
16 check_indices(1,:) = iterationCloud_indices(1);
17 iterationrownumberV2 = 2;
18
19 for i = 1:1:length(iterationCloud_indices)
20     %Get indice for ptCloudpoints
21     indice = iterationCloud_indices(i);
22     %Get corresponding XYZ point
23     value = ptCloudpoints(indice,:);
24
25     [nearest_indices, nearest_dists] = findNearestNeighbors(ptCloud,ptCloudpoints(
26       iterationCloud_indices(i),:),k);
27     %Check if nearest point is not already included
28     if ismember(nearest_indices, check_indices) == 0
29         %Add value to the iterationcloud
30         iterationcloud(iterationrownumberV2,:) = value;
31         check_indices(iterationrownumberV2) = indice;
32         iterationrownumberV2 = iterationrownumberV2 + 1;
33     end
34 end
35 end

```

## F. Matlab: ICR optimisation

### F.1 Main script: ICR optimization

```

1
2 clear
3
4 syms t
5 nullmatrix = 0
6
7 %Determine input variables
8 %Initialize reference point thats INSIDE the object
9 IP = [0, 0, 0]
10
11 %Initialize Center of Mass
12 CM = [0, 0, 0]
13
14 %Initialize stabilitymargin epsilon for use in wrench hull
15 stabilitymargin = 0.01
16
17 %Initialize dummy variables
18 error = 0;
19 NotInInterior = 0;
20 GraspstabilityCatch = 0;
21 QM3Error = 0;
22 QM3Low = 0;
23
24 %Create pointcloud from XYZ data
25 %Load pointcloud
26 % XYZdatapoints = readmatrix('XYZdatapoints.xlsx');
27 % XYZdatapoints_reduced = matfile('XYZdatapoints_reduced.mat');
28 XYZdatapoints_reduced_checked = matfile('XYZdatapoints_reduced_checked.mat');
29 XYZdatapoints = XYZdatapoints_reduced_checked.iterationcloud;
30 check_indices = XYZdatapoints_reduced_checked.check_indices;
31 surfacenormal = XYZdatapoints_reduced_checked.surfacenormal;
32
33 ptCloud = pointCloud(XYZdatapoints)
34 ptCloudpoints = [ptCloud.Location(:,1), ptCloud.Location(:,2), ptCloud.Location(:,3)];
35
36 %Calculating objective functions for all possible input combinations
37 tic
38 for q = 1:1:length(ptCloudpoints)
39 for w = 1:1:length(ptCloudpoints)
40 for r = 1:1:length(ptCloudpoints)
41 for t = 1:1:length(ptCloudpoints)
42
43     punt1 = q;
44     punt2 = w;
45     punt3 = r;
46     punt4 = t;
47     constraint_indices = [punt1 punt2 punt3 punt4];
48     constraints = [ptCloudpoints(punt1,:); ...
49                   ptCloudpoints(punt2,:); ...
50                   ptCloudpoints(punt3,:); ...
51                   ptCloudpoints(punt4,:)];
52     constraintsx = constraints(:,1);

```

```

53 constraintsy = constraints(:,2);
54 constraintsz = constraints(:,3);
55
56 try
57 stablegrasp = graspstability(constraint_indices, constraintx, constraintsy, constraintsz,
    surfacenormal, CM);
58 catch
59 %     First stability check, if convex hull can not be constructed
60 %     then not stable
61     GraspstabilityCatch = GraspstabilityCatch + 1;
62     continue
63 end
64
65 %     Second stability check
66 %     Check if grasp is constrained. (Calculates the wrenchhull and checks
67 %     if origin is in the interior of the convex hull
68 %     Function Inhull from source: John D'Errico (2022). Inhull (https://www.mathworks.com/matlabcentral/fileexchange/10226-inhull), MATLAB Central File Exchange. Retrieved October 26, 2022.
69 %     Graspstability = 0 when not stable; graspstability = 1 when stable.
70 if stablegrasp == 0
71     NotInInterior = NotInInterior + 1;
72     QM3(q,w,r,t) = 0;
73     continue
74 end
75
76 try
77 [QM3(q,w,r,t)] = Wrenchhull_2D_test(constraint_indices, constraintx, constraintsy,
    constraintsz, surfacenormal, CM);
78 catch
79     %Grasp is not stable and so return
80     QM3Error = QM3Error + 1;
81     continue
82 end
83
84 try
85 if QM3(q,w,r,t) > stabilitymargin
86     [QM2(q,w,r,t)] = ICR_optimisation(constraint_indices, constraintx, constraintsy,
        constraintsz, surfacenormal, IP);
87 else
88     QM3Low = QM3Low + 1;
89     QM2(q,w,r,t) = 0;
90 end
91 catch
92 %     Error: Not well constraint, continue with next iteration.
93 %     (when the loop is finished error should be 0)
94     error = error + 1;
95 end
96 end
97 end
98 end
99 end
100 toc
101
102 % Postprocessing

```

```

103 QM2_max = max(QM2(:));
104 QM2_Idx = find(QM2(:) == QM2_max);
105 [QM2_max_dim1, QM2_max_dim2, QM2_max_dim3, QM2_max_dim4] = ind2sub(size(QM2), QM2_Idx);
106
107 QM2_max_idx = [QM2_max_dim1 QM2_max_dim2 QM2_max_dim3 QM2_max_dim4]
108
109 QM2_idx_s = QM2_max_idx;
110 leftcounter = 2;
111 counter = 1;
112 for j = 1:1:length(QM2_idx_s)
113     for k = leftcounter:1:length(QM2_idx_s)
114         if ismember(QM2_idx_s(j,:), QM2_idx_s(k,:)) == 1
115             eliminaterow(counter,:) = k;
116             counter = counter + 1;
117         end
118     end
119     leftcounter = leftcounter + 1;
120 end
121
122 %Make new matrix consisting of rows to be deleted in QM2
123 counter = 1;
124 for i = 1:1:length(QM2_idx_s)
125     if ismember(i, eliminaterow) == 0
126         QM2_final(counter,:) = QM2_idx_s(i,:);
127         counter = counter + 1;
128     end
129 end
130
131 %Calculate QM3 for solutions with largest QM2 (all solutions for max(QM2
132 %included)
133 for i = 1:1:length(QM2_max_dim1)
134     QM3_final(i) = QM3(QM2_max_dim1(i), QM2_max_dim2(i), QM2_max_dim3(i), QM2_max_dim4(i))
135 end
136
137 %Calculate QM3 for all unique max(QM2) solutions
138 for i = 1:1:size(QM2_final,1)
139     QM3_final_red(i) = QM3(QM2_final(i,1), QM2_final(i,2), QM2_final(i,3), QM2_final(i,4))
140 end
141
142 %Overview max QM2 including corresponding QM3
143 QM = [QM2_max_dim1, QM2_max_dim2, QM2_max_dim3, QM2_max_dim4, QM3_final']
144 %Over max QM2 including corresponding QM3 Unique solutions
145 QM_red = [QM2_final(:,1), QM2_final(:,2), QM2_final(:,3), QM2_final(:,4), QM3_final_red']

```

## F.2 Function: 'ICR\_Optimisation'

In main script [F.1](#) this function is called as 'ICR\_Optimisation'.

```

1 function [QM2] = quality_measuresfunction(constraint_indices, constraintxs, constraintys,
2     constraintzs, surfacenormal, IP)
3
4 %Determine normal vectors of constraints
5 for i = 1:1:length(constraint_indices)
6     C_normal(i,:) = surfacenormal(constraint_indices(i,:),:);
7 end

```

```

 8 %Create sweeping lines from normal vectors
 9 t = -10:0.1:10;
10 for i = 1:1:length(constraint_indices)
11     SlineX(i,:) = [C_normal(i, 1)*t + constraintsx(i)];
12     SlineY(i,:) = [C_normal(i, 2)*t + constraintsy(i)];
13 end
14
15 for i = 1:1:length(constraint_indices)
16     SPointLine(i,:) = [SlineX(i,1), SlineY(i,1), 0];
17     EPointLine(i,:) = [SlineX(i,length(t)), SlineY(i,length(t)), 0 ];
18 end
19
20 %Source for code calculate intersection point for 2 lines in 3D: Reference https://nl.mathworks.
    com/matlabcentral/fileexchange/37192-intersection-point-of-lines-in-3d-space
21 counter = 1;
22 for i = 1:1:length(constraint_indices)
23     for j = 1:1:length(constraint_indices)
24         if i ~= j
25             Spointmatrix = [SPointLine(i,:); SPointLine(j,:)];
26             EPointmatrix = [EPointLine(i,:); EPointLine(j,:)];
27             [P_intersect(counter,:), distances(:,counter)] = lineIntersect3D(Spointmatrix,
                EPointmatrix);
28             counter = counter + 1;
29         end
30     end
31 end
32 %Delete rows containing NaN elements
33 P_intersect(any(isnan(P_intersect), 2), :) = [];
34
35 %Delete identical rows in P_intersect
36 P_intersect = unique(P_intersect, 'rows');
37
38 if size(P_intersect) < 1
39     QM2 = 0;
40     return
41 end
42
43 %Calculate intersection points for each IRC
44 %Iterate over IRC locations and determine moments
45 for i = 1:1:size(P_intersect,1)
46     %Calculate distance vectors for IRC
47     d1(i,:) = [(constraintsx(1)-P_intersect(i,1)), (constraintsy(1)-P_intersect(i,2)), (
        constraintsz(1)-P_intersect(i,3))];
48     d2(i,:) = [(constraintsx(2)-P_intersect(i,1)), (constraintsy(2)-P_intersect(i,2)), (
        constraintsz(2)-P_intersect(i,3))];
49     d3(i,:) = [(constraintsx(3)-P_intersect(i,1)), (constraintsy(3)-P_intersect(i,2)), (
        constraintsz(3)-P_intersect(i,3))];
50     d4(i,:) = [(constraintsx(4)-P_intersect(i,1)), (constraintsy(4)-P_intersect(i,2)), (
        constraintsz(4)-P_intersect(i,3))];
51
52     %Row is intersection point Column is XYZ
53     torque1(i,:) = cross(d1(i,:), C_normal(1,:));
54     torque2(i,:) = cross(d2(i,:), C_normal(2,:));
55     torque3(i,:) = cross(d3(i,:), C_normal(3,:));
56     torque4(i,:) = cross(d4(i,:), C_normal(4,:));

```

```

57 end
58
59 %Total moment about each ICR and determine objective function
60 if size(torque1,1) == 6
61     [M_ICR1p, M_ICR1n, M_ICR2p, M_ICR2n, M_ICR3p, M_ICR3n, M_ICR4p, M_ICR4n, M_ICR5p, M_ICR5n,
62         M_ICR6p, M_ICR6n] = TorqueV2S6(torque1, torque2 ,torque3, torque4);
63
64     absIRC1 = abs(M_ICR1p) + abs(M_ICR1n);
65     absIRC2 = abs(M_ICR2p) + abs(M_ICR2n);
66     absIRC3 = abs(M_ICR3p) + abs(M_ICR3n);
67     absIRC4 = abs(M_ICR4p) + abs(M_ICR4n);
68     absIRC5 = abs(M_ICR5p) + abs(M_ICR5n);
69     absIRC6 = abs(M_ICR6p) + abs(M_ICR6n);
70
71     QM2_abs = [absIRC1 absIRC2 absIRC3 absIRC4 absIRC5 absIRC6];
72     QM2 = min(QM2_abs);
73 end
74
75 if size(torque1,1) == 5
76     [M_ICR1p, M_ICR1n, M_ICR2p, M_ICR2n, M_ICR3p, M_ICR3n, M_ICR4p, M_ICR4n, M_ICR5p, M_ICR5n] =
77         TorqueV2S5(torque1, torque2 ,torque3, torque4);
78
79     absIRC1 = abs(M_ICR1p) + abs(M_ICR1n);
80     absIRC2 = abs(M_ICR2p) + abs(M_ICR2n);
81     absIRC3 = abs(M_ICR3p) + abs(M_ICR3n);
82     absIRC4 = abs(M_ICR4p) + abs(M_ICR4n);
83     absIRC5 = abs(M_ICR5p) + abs(M_ICR5n);
84
85     QM2_abs = [absIRC1 absIRC2 absIRC3 absIRC4 absIRC5];
86     QM2 = min(QM2_abs);
87 end
88
89 if size(torque1,1) == 4
90     [M_ICR1p, M_ICR1n, M_ICR2p, M_ICR2n, M_ICR3p, M_ICR3n, M_ICR4p, M_ICR4n] = TorqueV2S4(torque1,
91         torque2 ,torque3, torque4);
92
93     absIRC1 = abs(M_ICR1p) + abs(M_ICR1n);
94     absIRC2 = abs(M_ICR2p) + abs(M_ICR2n);
95     absIRC3 = abs(M_ICR3p) + abs(M_ICR3n);
96     absIRC4 = abs(M_ICR4p) + abs(M_ICR4n);
97
98     QM2_abs = [absIRC1 absIRC2 absIRC3 absIRC4];
99     QM2 = min(QM2_abs);
100 end
101
102 if size(torque1,1) == 3
103     [M_ICR1p, M_ICR1n, M_ICR2p, M_ICR2n, M_ICR3p, M_ICR3n] = TorqueV2S3(torque1, torque2 ,torque3,
104         torque4);
105
106     absIRC1 = abs(M_ICR1p) + abs(M_ICR1n);
107     absIRC2 = abs(M_ICR2p) + abs(M_ICR2n);
108     absIRC3 = abs(M_ICR3p) + abs(M_ICR3n);
109
110     QM2_abs = [absIRC1 absIRC2 absIRC3];
111     QM2 = min(QM2_abs);
112 end

```

```

108 if size(torque1,1) == 2
109     [M_ICR1p, M_ICR1n, M_ICR2p, M_ICR2n] = TorqueV2S2(torque1, torque2 ,torque3, torque4);
110
111     absIRC1 = abs(M_ICR1p) + abs(M_ICR1n);
112     absIRC2 = abs(M_ICR2p) + abs(M_ICR2n);
113
114     QM2_abs = [absIRC1 absIRC2];
115     QM2 = min(QM2_abs);
116 end
117 if size(torque1,1) == 1
118     [M_ICR1p, M_ICR1n] = TorqueV2S1(torque1, torque2 ,torque3, torque4);
119     absIRC1 = abs(M_ICR1p) + abs(M_ICR1n);
120
121     QM2_abs = [absIRC1];
122     QM2 = min(QM2_abs);
123 end
124
125 end

```

### F.3 Function: Graspstability

```

1
2 function graspstability = graspstability(constraint_indices, constraintxs, constraintsy,
3     constraintsz, surfacenormal, CM)
4 %Determine normal vectors of constraints
5 for i = 1:length(constraint_indices)
6     C_normal(i,:) = surfacenormal(constraint_indices(i,:));
7 end
8
9 for i = 1:length(constraint_indices)
10    f_n(i,:) = surfacenormal(constraint_indices(i,:));
11 end
12
13 d1 = [(constraintxs(1)-CM(1,1)), (constraintsy(1)-CM(1,2)), (constraintsz(1)-CM(1,3))];
14 d2 = [(constraintxs(2)-CM(1,1)), (constraintsy(2)-CM(1,2)), (constraintsz(2)-CM(1,3))];
15 d3 = [(constraintxs(3)-CM(1,1)), (constraintsy(3)-CM(1,2)), (constraintsz(3)-CM(1,3))];
16 d4 = [(constraintxs(4)-CM(1,1)), (constraintsy(4)-CM(1,2)), (constraintsz(4)-CM(1,3))];
17
18 %Row is intersection point Column is XYZ
19 torque1 = cross(d1, C_normal(1,:))';
20 torque2 = cross(d2, C_normal(2,:))';
21 torque3 = cross(d3, C_normal(3,:))';
22 torque4 = cross(d4, C_normal(4,:))';
23
24
25 f_c1 = f_n(1,:)' ;
26 f_c2 = f_n(2,:)' ;
27 f_c3 = f_n(3,:)' ;
28 f_c4 = f_n(4,:)' ;
29
30 %Delete Mx, Mz and Fz. 3D → 2D
31 f_c1(3) = [];
32 f_c2(3) = [];
33 f_c3(3) = [];

```

```

34 f_c4(3) = [];
35
36 for i = 1:1:2
37     torque1(1) = [];
38     torque2(1) = [];
39     torque3(1) = [];
40     torque4(1) = [];
41 end
42
43 W_1 = [f_c1;torque1];
44 W_2 = [f_c2;torque2];
45 W_3 = [f_c3;torque3];
46 W_4 = [f_c4;torque4];
47 G_w = [W_1, W_2, W_3, W_4];
48
49 for i = 1:1:length(G_w)
50     wrenchhull_Mz_x(i,:) = [G_w(1,i) G_w(3,i)];
51     wrenchhull_Mz_y(i,:) = [G_w(2,i) G_w(3,i)];
52 end
53
54 %Constructing Wrench Hull
55 % Only valid in 2D applications!
56 wrenchhull_Mz_xy = [wrenchhull_Mz_x(:,1) wrenchhull_Mz_y(:,1) wrenchhull_Mz_x(:,2)];
57
58 %Function Inhull calculates if a point is inside the wrench hull
59 %Function Inhull source: https://nl.mathworks.com/matlabcentral/fileexchange/10226-inhull
60 graspstability = inhull([0, 0, 0], wrenchhull_Mz_xy);
61
62 end

```

#### F.4 Function: TorqueV2S1 to TorqueV2S6

```

1 function [M_ICR1p M_ICR1n] = TorqueV2(torque1, torque2 ,torque3, torque4)
2
3 M_ICR1p = 0;
4 M_ICR1n = 0;
5
6 %Torque1 positive
7 if torque1(1,3) > 0
8     M_ICR1p = M_ICR1p + torque1(1,3);
9 end
10
11 if torque2(1,3) > 0
12     M_ICR1p = M_ICR1p + torque2(1,3);
13 end
14
15 if torque3(1,3) > 0
16     M_ICR1p = M_ICR1p + torque3(1,3);
17 end
18
19 if torque4(1,3) > 0
20     M_ICR1p = M_ICR1p + torque4(1,3);
21 end
22
23 %Torque1 negative

```

```
24 if torque1(1,3) < 0
25     M_ICR1n = M_ICR1n + torque1(1,3);
26 end
27
28 if torque2(1,3) < 0
29     M_ICR1n = M_ICR1n + torque2(1,3);
30 end
31
32 if torque3(1,3) < 0
33     M_ICR1n = M_ICR1n + torque3(1,3);
34 end
35
36 if torque4(1,3) < 0
37     M_ICR1n = M_ICR1n + torque4(1,3);
38 end
39
40 end
41
42 function [M_ICR1p M_ICR1n M_ICR2p M_ICR2n] = TorqueV2(torque1, torque2 ,torque3, torque4)
43
44 M_ICR1p = 0;
45 M_ICR1n = 0;
46 M_ICR2p = 0;
47 M_ICR2n = 0;
48
49 %Torque1 positive
50 if torque1(1,3) > 0
51     M_ICR1p = M_ICR1p + torque1(1,3);
52 end
53
54 if torque2(1,3) > 0
55     M_ICR1p = M_ICR1p + torque2(1,3);
56 end
57
58 if torque3(1,3) > 0
59     M_ICR1p = M_ICR1p + torque3(1,3);
60 end
61
62 if torque4(1,3) > 0
63     M_ICR1p = M_ICR1p + torque4(1,3);
64 end
65
66 %Torque1 negative
67 if torque1(1,3) < 0
68     M_ICR1n = M_ICR1n + torque1(1,3);
69 end
70
71 if torque2(1,3) < 0
72     M_ICR1n = M_ICR1n + torque2(1,3);
73 end
74
75 if torque3(1,3) < 0
76     M_ICR1n = M_ICR1n + torque3(1,3);
77 end
78
```

```

79 if torque4(1,3) < 0
80     M_ICR1n = M_ICR1n + torque4(1,3);
81 end
82
83
84
85 %Torque2 positive
86 if torque1(2,3) > 0
87     M_ICR2p = M_ICR2p + torque1(2,3);
88 end
89
90 if torque2(2,3) > 0
91     M_ICR2p = M_ICR2p + torque2(2,3);
92 end
93
94 if torque3(2,3) > 0
95     M_ICR2p = M_ICR2p + torque3(2,3);
96 end
97
98 if torque4(2,3) > 0
99     M_ICR2p = M_ICR2p + torque4(2,3);
100 end
101
102 %Torque2 negative
103 if torque1(2,3) < 0
104     M_ICR2n = M_ICR2n + torque1(2,3);
105 end
106
107 if torque2(2,3) < 0
108     M_ICR2n = M_ICR2n + torque2(2,3);
109 end
110
111 if torque3(2,3) < 0
112     M_ICR2n = M_ICR2n + torque3(2,3);
113 end
114
115 if torque4(2,3) < 0
116     M_ICR2n = M_ICR2n + torque4(2,3);
117 end
118
119
120 end
121
122 function [M_ICR1p M_ICR1n M_ICR2p M_ICR2n M_ICR3p M_ICR3n] = TorqueV2(torque1, torque2 ,torque3,
    torque4)
123
124 M_ICR1p = 0;
125 M_ICR1n = 0;
126 M_ICR2p = 0;
127 M_ICR2n = 0;
128 M_ICR3p = 0;
129 M_ICR3n = 0;
130
131 %Torque1 positive
132 if torque1(1,3) > 0

```

```
133     M_ICR1p = M_ICR1p + torque1(1,3);
134 end
135
136 if torque2(1,3) > 0
137     M_ICR1p = M_ICR1p + torque2(1,3);
138 end
139
140 if torque3(1,3) > 0
141     M_ICR1p = M_ICR1p + torque3(1,3);
142 end
143
144 if torque4(1,3) > 0
145     M_ICR1p = M_ICR1p + torque4(1,3);
146 end
147
148 %Torque1 negative
149 if torque1(1,3) < 0
150     M_ICR1n = M_ICR1n + torque1(1,3);
151 end
152
153 if torque2(1,3) < 0
154     M_ICR1n = M_ICR1n + torque2(1,3);
155 end
156
157 if torque3(1,3) < 0
158     M_ICR1n = M_ICR1n + torque3(1,3);
159 end
160
161 if torque4(1,3) < 0
162     M_ICR1n = M_ICR1n + torque4(1,3);
163 end
164
165
166
167 %Torque2 positive
168 if torque1(2,3) > 0
169     M_ICR2p = M_ICR2p + torque1(2,3);
170 end
171
172 if torque2(2,3) > 0
173     M_ICR2p = M_ICR2p + torque2(2,3);
174 end
175
176 if torque3(2,3) > 0
177     M_ICR2p = M_ICR2p + torque3(2,3);
178 end
179
180 if torque4(2,3) > 0
181     M_ICR2p = M_ICR2p + torque4(2,3);
182 end
183
184 %Torque2 negative
185 if torque1(2,3) < 0
186     M_ICR2n = M_ICR2n + torque1(2,3);
187 end
```

```

188
189     if torque2(2,3) < 0
190         M_ICR2n = M_ICR2n + torque2(2,3);
191     end
192
193     if torque3(2,3) < 0
194         M_ICR2n = M_ICR2n + torque3(2,3);
195     end
196
197     if torque4(2,3) < 0
198         M_ICR2n = M_ICR2n + torque4(2,3);
199     end
200
201
202     %ICR3 positive
203     if torque1(3,3) > 0
204         M_ICR3p = M_ICR3p + torque1(3,3);
205     end
206
207     if torque2(3,3) > 0
208         M_ICR3p = M_ICR3p + torque2(3,3);
209     end
210
211     if torque3(3,3) > 0
212         M_ICR3p = M_ICR3p + torque3(3,3);
213     end
214
215     if torque4(3,3) > 0
216         M_ICR3p = M_ICR3p + torque4(3,3);
217     end
218
219     %Torque3 negative
220     if torque1(3,3) < 0
221         M_ICR3n = M_ICR3n + torque1(3,3);
222     end
223
224     if torque2(3,3) < 0
225         M_ICR3n = M_ICR3n + torque2(3,3);
226     end
227
228     if torque3(3,3) < 0
229         M_ICR3n = M_ICR3n + torque3(3,3);
230     end
231
232     if torque4(3,3) < 0
233         M_ICR3n = M_ICR3n + torque4(3,3);
234     end
235
236
237 end
238
239
240 function [M_ICR1p M_ICR1n M_ICR2p M_ICR2n M_ICR3p M_ICR3n M_ICR4p M_ICR4n] = TorqueV2(torque1,
241     torque2 ,torque3, torque4)

```

```
242 M_ICR1p = 0;
243 M_ICR1n = 0;
244 M_ICR2p = 0;
245 M_ICR2n = 0;
246 M_ICR3p = 0;
247 M_ICR3n = 0;
248 M_ICR4p = 0;
249 M_ICR4n = 0;
250
251 %Torque1 positive
252 if torque1(1,3) > 0
253     M_ICR1p = M_ICR1p + torque1(1,3);
254 end
255
256 if torque2(1,3) > 0
257     M_ICR1p = M_ICR1p + torque2(1,3);
258 end
259
260 if torque3(1,3) > 0
261     M_ICR1p = M_ICR1p + torque3(1,3);
262 end
263
264 if torque4(1,3) > 0
265     M_ICR1p = M_ICR1p + torque4(1,3);
266 end
267
268 %Torque1 negative
269 if torque1(1,3) < 0
270     M_ICR1n = M_ICR1n + torque1(1,3);
271 end
272
273 if torque2(1,3) < 0
274     M_ICR1n = M_ICR1n + torque2(1,3);
275 end
276
277 if torque3(1,3) < 0
278     M_ICR1n = M_ICR1n + torque3(1,3);
279 end
280
281 if torque4(1,3) < 0
282     M_ICR1n = M_ICR1n + torque4(1,3);
283 end
284
285
286 if size(torque1,1) > 1
287     %Torque2 positive
288     if torque1(2,3) > 0
289         M_ICR2p = M_ICR2p + torque1(2,3);
290     end
291
292     if torque2(2,3) > 0
293         M_ICR2p = M_ICR2p + torque2(2,3);
294     end
295
296     if torque3(2,3) > 0
```

```
297     M_ICR2p = M_ICR2p + torque3(2,3);
298 end
299
300 if torque4(2,3) > 0
301     M_ICR2p = M_ICR2p + torque4(2,3);
302 end
303
304 %Torque2 negative
305 if torque1(2,3) < 0
306     M_ICR2n = M_ICR2n + torque1(2,3);
307 end
308
309 if torque2(2,3) < 0
310     M_ICR2n = M_ICR2n + torque2(2,3);
311 end
312
313 if torque3(2,3) < 0
314     M_ICR2n = M_ICR2n + torque3(2,3);
315 end
316
317 if torque4(2,3) < 0
318     M_ICR2n = M_ICR2n + torque4(2,3);
319 end
320 end
321
322 if size(torque1,1) > 2
323     %ICR3 positive
324     if torque1(3,3) > 0
325         M_ICR3p = M_ICR3p + torque1(3,3);
326     end
327
328     if torque2(3,3) > 0
329         M_ICR3p = M_ICR3p + torque2(3,3);
330     end
331
332     if torque3(3,3) > 0
333         M_ICR3p = M_ICR3p + torque3(3,3);
334     end
335
336     if torque4(3,3) > 0
337         M_ICR3p = M_ICR3p + torque4(3,3);
338     end
339
340     %Torque3 negative
341     if torque1(3,3) < 0
342         M_ICR3n = M_ICR3n + torque1(3,3);
343     end
344
345     if torque2(3,3) < 0
346         M_ICR3n = M_ICR3n + torque2(3,3);
347     end
348
349     if torque3(3,3) < 0
350         M_ICR3n = M_ICR3n + torque3(3,3);
351     end
```

```

352
353     if torque4(3,3) < 0
354         M_ICR3n = M_ICR3n + torque4(3,3);
355     end
356 end
357
358 if size(torque1,1) > 3
359     %ICR4 positive
360     if torque1(4,3) > 0
361         M_ICR4p = M_ICR4p + torque1(4,3);
362     end
363
364     if torque2(4,3) > 0
365         M_ICR4p = M_ICR4p + torque2(4,3);
366     end
367
368     if torque3(4,3) > 0
369         M_ICR4p = M_ICR4p + torque3(4,3);
370     end
371
372     if torque4(4,3) > 0
373         M_ICR4p = M_ICR4p + torque4(4,3);
374     end
375
376     %Torque3 negative
377     if torque1(4,3) < 0
378         M_ICR4n = M_ICR4n + torque1(4,3);
379     end
380
381     if torque2(4,3) < 0
382         M_ICR4n = M_ICR4n + torque2(4,3);
383     end
384
385     if torque3(4,3) < 0
386         M_ICR4n = M_ICR4n + torque3(4,3);
387     end
388
389     if torque4(4,3) < 0
390         M_ICR4n = M_ICR4n + torque4(4,3);
391     end
392 end
393
394 end
395
396
397 function [M_ICR1p M_ICR1n M_ICR2p M_ICR2n M_ICR3p M_ICR3n M_ICR4p M_ICR4n M_ICR5p M_ICR5n] =
    TorqueV2(torque1, torque2 ,torque3, torque4)
398
399 M_ICR1p = 0;
400 M_ICR1n = 0;
401 M_ICR2p = 0;
402 M_ICR2n = 0;
403 M_ICR3p = 0;
404 M_ICR3n = 0;
405 M_ICR4p = 0;

```

```
406 M_ICR4n = 0;
407 M_ICR5p = 0;
408 M_ICR5n = 0;
409
410 %Torque1 positive
411 if torque1(1,3) > 0
412     M_ICR1p = M_ICR1p + torque1(1,3);
413 end
414
415 if torque2(1,3) > 0
416     M_ICR1p = M_ICR1p + torque2(1,3);
417 end
418
419 if torque3(1,3) > 0
420     M_ICR1p = M_ICR1p + torque3(1,3);
421 end
422
423 if torque4(1,3) > 0
424     M_ICR1p = M_ICR1p + torque4(1,3);
425 end
426
427 %Torque1 negative
428 if torque1(1,3) < 0
429     M_ICR1n = M_ICR1n + torque1(1,3);
430 end
431
432 if torque2(1,3) < 0
433     M_ICR1n = M_ICR1n + torque2(1,3);
434 end
435
436 if torque3(1,3) < 0
437     M_ICR1n = M_ICR1n + torque3(1,3);
438 end
439
440 if torque4(1,3) < 0
441     M_ICR1n = M_ICR1n + torque4(1,3);
442 end
443
444
445
446 %Torque2 positive
447 if torque1(2,3) > 0
448     M_ICR2p = M_ICR2p + torque1(2,3);
449 end
450
451 if torque2(2,3) > 0
452     M_ICR2p = M_ICR2p + torque2(2,3);
453 end
454
455 if torque3(2,3) > 0
456     M_ICR2p = M_ICR2p + torque3(2,3);
457 end
458
459 if torque4(2,3) > 0
460     M_ICR2p = M_ICR2p + torque4(2,3);
```

```
461 end
462
463 %Torque2 negative
464 if torque1(2,3) < 0
465     M_ICR2n = M_ICR2n + torque1(2,3);
466 end
467
468 if torque2(2,3) < 0
469     M_ICR2n = M_ICR2n + torque2(2,3);
470 end
471
472 if torque3(2,3) < 0
473     M_ICR2n = M_ICR2n + torque3(2,3);
474 end
475
476 if torque4(2,3) < 0
477     M_ICR2n = M_ICR2n + torque4(2,3);
478 end
479
480
481
482 %ICR3 positive
483 if torque1(3,3) > 0
484     M_ICR3p = M_ICR3p + torque1(3,3);
485 end
486
487 if torque2(3,3) > 0
488     M_ICR3p = M_ICR3p + torque2(3,3);
489 end
490
491 if torque3(3,3) > 0
492     M_ICR3p = M_ICR3p + torque3(3,3);
493 end
494
495 if torque4(3,3) > 0
496     M_ICR3p = M_ICR3p + torque4(3,3);
497 end
498
499 %ICR3 negative
500 if torque1(3,3) < 0
501     M_ICR3n = M_ICR3n + torque1(3,3);
502 end
503
504 if torque2(3,3) < 0
505     M_ICR3n = M_ICR3n + torque2(3,3);
506 end
507
508 if torque3(3,3) < 0
509     M_ICR3n = M_ICR3n + torque3(3,3);
510 end
511
512 if torque4(3,3) < 0
513     M_ICR3n = M_ICR3n + torque4(3,3);
514 end
515
```

```
516
517 %ICR4 positive
518 if torque1(4,3) > 0
519     M_ICR4p = M_ICR4p + torque1(4,3);
520 end
521
522 if torque2(4,3) > 0
523     M_ICR4p = M_ICR4p + torque2(4,3);
524 end
525
526 if torque3(4,3) > 0
527     M_ICR4p = M_ICR4p + torque3(4,3);
528 end
529
530 if torque4(4,3) > 0
531     M_ICR4p = M_ICR4p + torque4(4,3);
532 end
533
534 %ICR4 negative
535 if torque1(4,3) < 0
536     M_ICR4n = M_ICR4n + torque1(4,3);
537 end
538
539 if torque2(4,3) < 0
540     M_ICR4n = M_ICR4n + torque2(4,3);
541 end
542
543 if torque3(4,3) < 0
544     M_ICR4n = M_ICR4n + torque3(4,3);
545 end
546
547 if torque4(4,3) < 0
548     M_ICR4n = M_ICR4n + torque4(4,3);
549 end
550
551
552 %ICR5 positive
553 if torque1(5,3) > 0
554     M_ICR5p = M_ICR5p + torque1(5,3);
555 end
556
557 if torque2(5,3) > 0
558     M_ICR5p = M_ICR5p + torque2(5,3);
559 end
560
561 if torque3(5,3) > 0
562     M_ICR5p = M_ICR5p + torque3(5,3);
563 end
564
565 if torque4(5,3) > 0
566     M_ICR5p = M_ICR5p + torque4(5,3);
567 end
568
569 %ICR5 negative
570 if torque1(5,3) < 0
```

```
571     M_ICR5n = M_ICR5n + torque1(5,3);
572 end
573
574 if torque2(5,3) < 0
575     M_ICR5n = M_ICR5n + torque2(5,3);
576 end
577
578 if torque3(5,3) < 0
579     M_ICR5n = M_ICR5n + torque3(5,3);
580 end
581
582 if torque4(5,3) < 0
583     M_ICR5n = M_ICR5n + torque4(5,3);
584 end
585
586
587 end
588
589
590
591 function [M_ICR1p M_ICR1n M_ICR2p M_ICR2n M_ICR3p M_ICR3n M_ICR4p M_ICR4n M_ICR5p M_ICR5n M_ICR6p
592     M_ICR6n] = TorqueV2(torque1, torque2 ,torque3, torque4)
593
594 M_ICR1p = 0;
595 M_ICR1n = 0;
596 M_ICR2p = 0;
597 M_ICR2n = 0;
598 M_ICR3p = 0;
599 M_ICR3n = 0;
600 M_ICR4p = 0;
601 M_ICR4n = 0;
602 M_ICR5p = 0;
603 M_ICR5n = 0;
604 M_ICR6p = 0;
605 M_ICR6n = 0;
606
607 %Torque1 positive
608 if torque1(1,3) > 0
609     M_ICR1p = M_ICR1p + torque1(1,3);
610 end
611
612 if torque2(1,3) > 0
613     M_ICR1p = M_ICR1p + torque2(1,3);
614 end
615
616 if torque3(1,3) > 0
617     M_ICR1p = M_ICR1p + torque3(1,3);
618 end
619
620 if torque4(1,3) > 0
621     M_ICR1p = M_ICR1p + torque4(1,3);
622 end
623
624 %Torque1 negative
625 if torque1(1,3) < 0
```

```
625     M_ICR1n = M_ICR1n + torque1(1,3);
626 end
627
628 if torque2(1,3) < 0
629     M_ICR1n = M_ICR1n + torque2(1,3);
630 end
631
632 if torque3(1,3) < 0
633     M_ICR1n = M_ICR1n + torque3(1,3);
634 end
635
636 if torque4(1,3) < 0
637     M_ICR1n = M_ICR1n + torque4(1,3);
638 end
639
640
641
642 %Torque2 positive
643 if torque1(2,3) > 0
644     M_ICR2p = M_ICR2p + torque1(2,3);
645 end
646
647 if torque2(2,3) > 0
648     M_ICR2p = M_ICR2p + torque2(2,3);
649 end
650
651 if torque3(2,3) > 0
652     M_ICR2p = M_ICR2p + torque3(2,3);
653 end
654
655 if torque4(2,3) > 0
656     M_ICR2p = M_ICR2p + torque4(2,3);
657 end
658
659 %Torque2 negative
660 if torque1(2,3) < 0
661     M_ICR2n = M_ICR2n + torque1(2,3);
662 end
663
664 if torque2(2,3) < 0
665     M_ICR2n = M_ICR2n + torque2(2,3);
666 end
667
668 if torque3(2,3) < 0
669     M_ICR2n = M_ICR2n + torque3(2,3);
670 end
671
672 if torque4(2,3) < 0
673     M_ICR2n = M_ICR2n + torque4(2,3);
674 end
675
676
677
678 %ICR3 positive
679 if torque1(3,3) > 0
```

```
680     M_ICR3p = M_ICR3p + torque1(3,3);
681 end
682
683 if torque2(3,3) > 0
684     M_ICR3p = M_ICR3p + torque2(3,3);
685 end
686
687 if torque3(3,3) > 0
688     M_ICR3p = M_ICR3p + torque3(3,3);
689 end
690
691 if torque4(3,3) > 0
692     M_ICR3p = M_ICR3p + torque4(3,3);
693 end
694
695 %ICR3 negative
696 if torque1(3,3) < 0
697     M_ICR3n = M_ICR3n + torque1(3,3);
698 end
699
700 if torque2(3,3) < 0
701     M_ICR3n = M_ICR3n + torque2(3,3);
702 end
703
704 if torque3(3,3) < 0
705     M_ICR3n = M_ICR3n + torque3(3,3);
706 end
707
708 if torque4(3,3) < 0
709     M_ICR3n = M_ICR3n + torque4(3,3);
710 end
711
712
713 %ICR4 positive
714 if torque1(4,3) > 0
715     M_ICR4p = M_ICR4p + torque1(4,3);
716 end
717
718 if torque2(4,3) > 0
719     M_ICR4p = M_ICR4p + torque2(4,3);
720 end
721
722 if torque3(4,3) > 0
723     M_ICR4p = M_ICR4p + torque3(4,3);
724 end
725
726 if torque4(4,3) > 0
727     M_ICR4p = M_ICR4p + torque4(4,3);
728 end
729
730 %ICR4 negative
731 if torque1(4,3) < 0
732     M_ICR4n = M_ICR4n + torque1(4,3);
733 end
734
```

```
735 if torque2(4,3) < 0
736     M_ICR4n = M_ICR4n + torque2(4,3);
737 end
738
739 if torque3(4,3) < 0
740     M_ICR4n = M_ICR4n + torque3(4,3);
741 end
742
743 if torque4(4,3) < 0
744     M_ICR4n = M_ICR4n + torque4(4,3);
745 end
746
747
748 %ICR5 positive
749 if torque1(5,3) > 0
750     M_ICR5p = M_ICR5p + torque1(5,3);
751 end
752
753 if torque2(5,3) > 0
754     M_ICR5p = M_ICR5p + torque2(5,3);
755 end
756
757 if torque3(5,3) > 0
758     M_ICR5p = M_ICR5p + torque3(5,3);
759 end
760
761 if torque4(5,3) > 0
762     M_ICR5p = M_ICR5p + torque4(5,3);
763 end
764
765 %ICR5 negative
766 if torque1(5,3) < 0
767     M_ICR5n = M_ICR5n + torque1(5,3);
768 end
769
770 if torque2(5,3) < 0
771     M_ICR5n = M_ICR5n + torque2(5,3);
772 end
773
774 if torque3(5,3) < 0
775     M_ICR5n = M_ICR5n + torque3(5,3);
776 end
777
778 if torque4(5,3) < 0
779     M_ICR5n = M_ICR5n + torque4(5,3);
780 end
781
782 %ICR6 positive
783 if torque1(6,3) > 0
784     M_ICR6p = M_ICR6p + torque1(6,3);
785 end
786
787 if torque2(6,3) > 0
788     M_ICR6p = M_ICR6p + torque2(6,3);
789 end
```

```
790
791 if torque3(6,3) > 0
792     M_ICR6p = M_ICR6p + torque3(6,3);
793 end
794
795 if torque4(6,3) > 0
796     M_ICR6p = M_ICR6p + torque4(6,3);
797 end
798
799 %IC6 negative
800 if torque1(6,3) < 0
801     M_ICR6n = M_ICR6n + torque1(6,3);
802 end
803
804 if torque2(6,3) < 0
805     M_ICR6n = M_ICR6n + torque2(6,3);
806 end
807
808 if torque3(6,3) < 0
809     M_ICR6n = M_ICR6n + torque3(6,3);
810 end
811
812 if torque4(6,3) < 0
813     M_ICR6n = M_ICR6n + torque4(6,3);
814 end
815
816 end
```

## G. Matlab: Rotational Offset

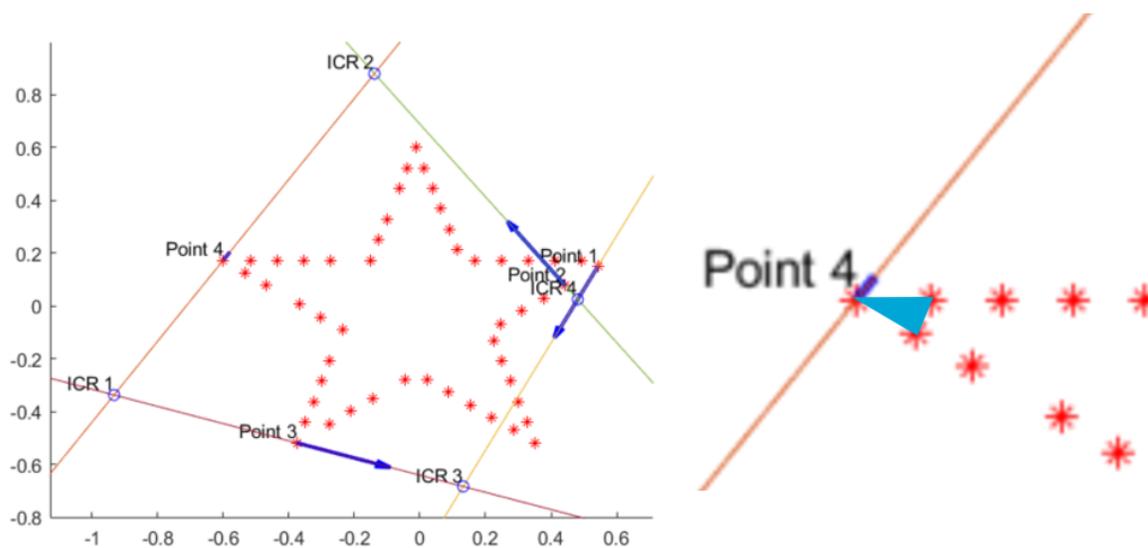
```

1 clear
2 %All input in mm
3 %Generate corner points
4 XY = [-9,-9,0; 9,-9,0;-9,9,0;9,9,0];
5 constraints = [-9,8,0;-8,9,0;8,-9,0];
6
7 %Rotational offset
8 % theta_z = 0;
9 theta_z = 0.30;
10
11 %shift object XYZ
12 % X_shift = 0;
13 X_shift = 80/1000;
14 Y_shift = 0;
15 Z_shift = 0;
16
17 %Transformation matrix
18 C_z = [cosd(theta_z) -sind(theta_z) 0; ...
19        sind(theta_z) cosd(theta_z) 0;...
20        0 0 1];
21
22 %Apply shift and rotation to dataset
23 XYZPointsRotation = C_z*XY';
24 XYZPointsRotation = XYZPointsRotation';
25
26 XYZPointsRotationShifted(:,1) = XYZPointsRotation(:,1) + X_shift;
27 XYZPointsRotationShifted(:,2) = XYZPointsRotation(:,2) + Y_shift;
28 XYZPointsRotationShifted(:,3) = XYZPointsRotation(:,3) + Z_shift;
29
30 %Calculate shortest distance constraint to object
31 %Distance constraint 1 to line
32 v1 = [XYZPointsRotationShifted(1,1), XYZPointsRotationShifted(1,2), 0];
33 v2 = [XYZPointsRotationShifted(3,1), XYZPointsRotationShifted(3,2), 0];
34 CM = [constraints(1,1),constraints(1,2),constraints(1,3)];
35
36 a = v1 - v2;
37 b = CM - v2;
38 shortest_dist_1 = norm(cross(a,b)) / norm(a);
39
40 %Distance constraint 2 to line
41 v1 = [XYZPointsRotationShifted(3,1), XYZPointsRotationShifted(3,2), 0];
42 v2 = [XYZPointsRotationShifted(4,1), XYZPointsRotationShifted(4,2), 0];
43 CM = [constraints(2,1),constraints(2,2),constraints(2,3)];
44
45 a = v1 - v2;
46 b = CM - v2;
47 shortest_dist_2 = norm(cross(a,b)) / norm(a);
48
49 %Distance constraint 3 to line
50 v1 = [XYZPointsRotationShifted(1,1), XYZPointsRotationShifted(1,2), 0];
51 v2 = [XYZPointsRotationShifted(2,1), XYZPointsRotationShifted(2,2), 0];
52 CM = [constraints(3,1),constraints(3,2),constraints(3,3)];
53

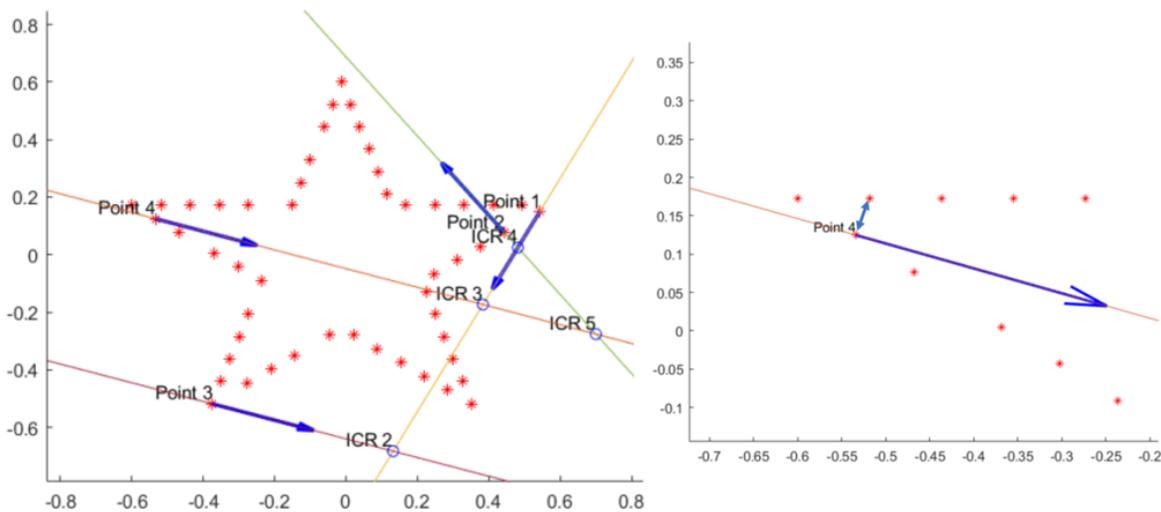
```

```
54 a = v1 - v2;
55 b = CM - v2;
56 shortest_dist_3 = norm(cross(a,b)) / norm(a);
57
58 %Results
59 shortest_dist = [shortest_dist_1; shortest_dist_2; shortest_dist_3]
60 shortest_dist_um = shortest_dist*1000
```

**H. Discussion: Example limitation wrong calculation of normal vector**



**Figure 58.** Point 4 wrong normal vector. Algorithm takes nearest point to set up a line or plane. In this case, two points have identical distance from point 4, and make a sharp angle and construct a plane, with a normal vector in the Z-axis.

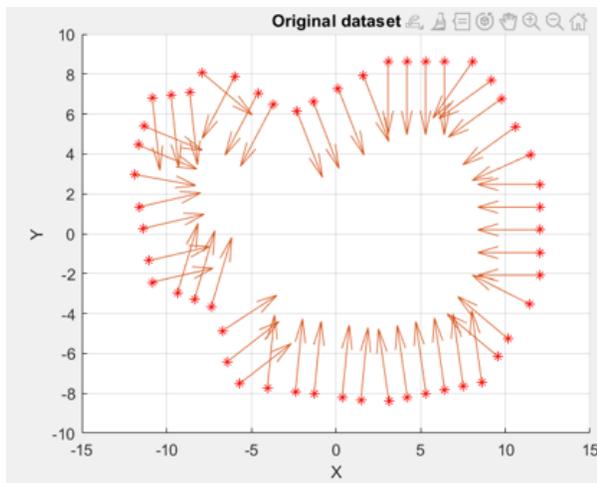


**Figure 59.** Point 4 wrong normal vector. The algorithm takes the nearest point to set up a line or plane. In this case, the point corresponding to a different surface is the nearest point. Therefore, calculating a wrong normal vector.

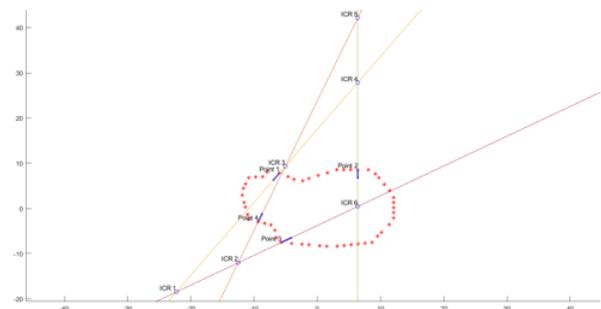
## I. Example constraint position optimization with different boundary conditions

In figure 60 an example is given of a more complex shape and size. For this object two optimization results are discussed in this chapter. First, the boundary condition that checks the stability of the grasp is set on a fairly low value. The obtained optimized grasp is displayed in figure 61 and 62. What can be noticed is that only contact point 1 has a relatively small magnitude of its force in the negative  $x$ -axis direction. It is easy to see that this grasp is not able to withstand large disturbance forces, which is further emphasized by the wrench hull (figure 63. Here, the radius of the ball that can be fitted in the wrench hull is fairly small.

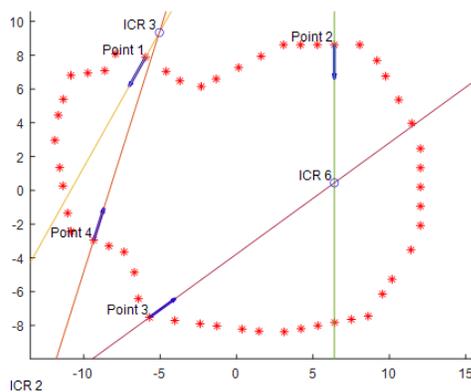
A second optimization result is calculated for a grasp with a higher boundary condition. Here, the boundary condition is set to be 30 times larger. The now-obtained best grasp is significantly different. The obtained best grasp for a larger boundary condition is shown in figure 64. The set of contact points of this grasp are better able to resist external forces on the object.



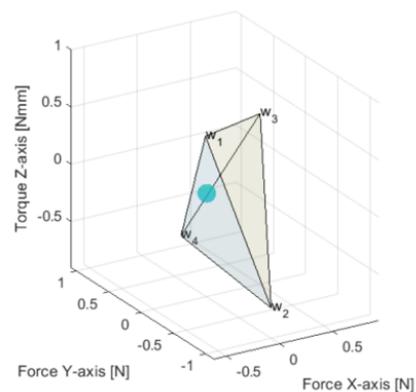
**Figure 60.** Point cloud of example object and the normal vector for each data point



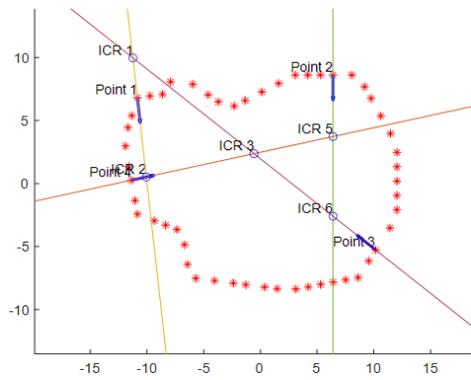
**Figure 61.** Result using the contact point position optimization; boundary condition set to be equal to 0.01.



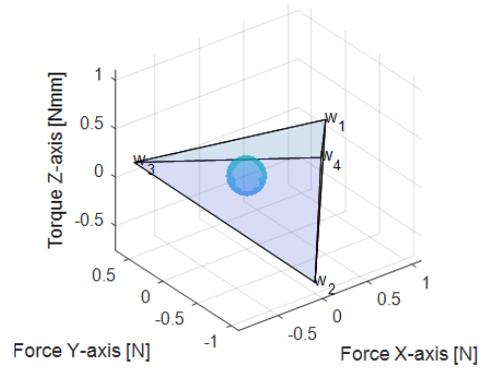
**Figure 62.** Zoomed-in plot of best result position optimization; boundary condition is set to be equal to 0.01.



**Figure 63.** Wrench hull of grasp; boundary condition set to be equal to 0.01.



**Figure 64.** Result using the contact point position optimization; boundary condition set to be equal to 0.3.



**Figure 65.** Wrench hull of grasp; boundary condition set to be equal to 0.3.

## J. Literature Review: Micro-positioning and fixating of an object in a vacuum environment with restrictive contamination constraints

— 1/18

### Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
	<b>Introduction</b>	<b>2</b>
1.1	Objective	3
1.2	Definition	3
1.3	Content	3
<b>2</b>	<b>Methods</b>	<b>3</b>
2.1	Search strategy	3
2.2	Eligibility criteria	3
<b>3</b>	<b>Classification</b>	<b>3</b>
<b>4</b>	<b>Results</b>	<b>4</b>
4.1	Particle contamination	4
4.2	Mechanical particle generation	4
	Pump down • Gate valve • Shocks and vibrations • Friction	
4.3	Physics particle generation	5
	Outgassing • Transportation • Van der Waals forces • Electrostatic attraction	
4.4	Compliant mechanisms	7
	Statically balanced compliant mechanisms	
4.5	Positioning stages	8
4.6	Micro-gripper	9
	Alignment using externally applied force • Passive alignment • Statically balanced gripper • Form-closure gripper • Bi-stable locking mechanism • Switchable binary stiffness gripper • Micro-gripper based on two independent positioning stages	
<b>5</b>	<b>Overview relationship strategies and particle generation</b>	<b>13</b>
<b>6</b>	<b>Discussion</b>	<b>15</b>
6.1	Main findings	15
6.2	Limitations	15
6.3	Recommendations	16
<b>7</b>	<b>Conclusion</b>	<b>16</b>
	<b>References</b>	<b>16</b>

# Micro-positioning and fixating of an object in an ultra-high vacuum environment with restrictive contamination constraints - Literature Review

Jelle Kortman

## Abstract

In the high-tech industry there are multiple applications where high-precision positioning, pick-up and placing of a millimeter or centimeter scale object is performed. Also, particle contamination becomes increasingly more important in the ever-increasing efforts to increase performance in applications with high cleanliness restrictions in ultra-high vacuum systems. This literature review had the goal to give an overview of positioning and gripper strategies, and how these strategies have an effect on particle generation in an ultra-high vacuum environment. In this study, only compliant mechanisms are described, because of the large benefits with respect to small-scale, low backlash and limited particle generation due to no friction. First, comprehensive research is performed on what the different sources of particle generation can be present in an ultra-high-vacuum environment. It was found that adhesive and abrasive wear is the leading factor for particle contamination source that has a strong relationship with the obtained positioning and gripper strategies. A second important factor with respect to the obtained strategies is outgassing, which is dependent on the used material properties.

## Keywords

High precision positioning – Fixating objects – grasping objects – Ultra-high vacuum – compliant mechanisms – Particle contamination

## 1. Introduction

Micro-positioning, fixating and grasping an object is commonly referred to as object handling and is often performed in the high-tech industry. Performing these tasks in an ultra-high vacuum and where particle contamination results in high performance limiting risks is most commonly known in the semiconductor industry. Here, a wafer is picked up, transferred to a secondary chamber and a new wafer is positioned [1]. Although the wafer is handled very carefully, at the location where the pins of the gripper make contact with the wafer particles are still generated. In vacuum, these particles can be transported to every open volume, therefore the generated particles on the non-critical backside of the wafer will eventually be transported to the critical topside of the wafer surface [2]. Besides the semiconductor industry and in many vacuum systems particle contamination needs to be kept at a minimum [2, 3, 4].

Handling strategies largely depend on the object dimensions, for a wafer this is a large circular disk. Each object shape, size and environment a different strategy to perform object handling is required. For mechanisms where small design areas, high precision, vacuum environment and particle contamination risks are important factors, a compliant mechanism design can deliver highly beneficial benefits. The monolithic design approach results in multiple advantages when a force is applied, such as zero friction, no hysteresis,

no backlash and no lubrication required [5]. Therefore, this literature review focuses solely on compliant mechanisms and reports the various different strategies that were used in literature and their differentiating characteristics for positioning and gripper mechanisms

Positioning stages are widely used in different industries, increasingly more interest is going to micro- and nanopositioning stages. Applications for high precision positioning stages can be found in fields like scanning probe microscopy [6], cell micro-injection [7] or precision alignment of various objects, for example precision alignment of mirrors for space or lithography applications [5]. For the latter example, particle contamination is also a large concern. In ultra-high vacuum applications, grippers can make use of different strategies in order to temporarily fixate an object and comes in a wide variety of designs. Multiple gripper mechanisms only require actuation for opening and closing of the gripper [8, 9, 10, 11, 12, 13], while others need active actuation to remain the required force on the grasped object [11, 14]. Gripper mechanisms can be statically balanced which can lower required actuation forces and can provide a near 1:1 force input-output relation [15, 16].

Knowledge on how different strategies for object handling can be used and how they affect particle generation is useful to generate new designs for kinematic couplings with high cleanliness restrictions.

### 1.1 Objective

In current available literature, there is limited information available on how different kinds of positioning and gripper strategies have an effect on particle generation. The goal of this literature review is to provide an overview on the unique characteristics of the positioning and gripper strategies and describe their respective effect on particle generation.

### 1.2 Definition

This literature review describes multiple gripper mechanisms, a definition is set in order to determine the boundaries of gripper mechanisms that are included:

*Gripper: A mechanism that is able to perform multiple tasks including grasping, temporary fixating and ultimately releasing of a millimeter or centimeter scale object.*

### 1.3 Content

This literature review entails the following chapters. In chapter 2, the methods are provided and include the search strategy, eligibility criteria and the collection process. In chapter 3, the classification of the categories is shown. In chapter 4, the results of the literature review are elaborated, followed by a summarized overview of the obtained relationships between particle generation and the described strategies in chapter 5. In chapter 6 the conclusions are elaborated on in the discussion and followed by the conclusion in chapter 7.

## 2. Methods

### 2.1 Search strategy

In order to obtain relevant literature concerning the subjects multiple search terms are used in the search engine Scopus. Second, information is obtained in conversations with experts in the respective fields. In figure 1 the used search terms are described to obtain relevant literature relating to positioning stages and gripper mechanisms. In figure 2 the used search terms are described to obtain relevant literature relating to particle contamination sources.

OR	AND			AND NOT
	Subject	Function	Feature	Mems
	Compliant mechanism*	Positioning stage 2-DOF	Ultra-high vacuum	Moems
	Compliant	Micro-stage 2-DOF	Statically balanced	Control
	Wafer	Gripper handling	Passive alignment	
			Form closure	
			Bi-stable	
			Binary stiffness	
			Particle contamination	
			High precision	
			Mass balancing	

**Figure 1.** Overview of search terms and respective categories for gripper and positioning mechanisms.

OR	AND			AND NOT
	Subject	Function	Feature	
	Particle contamination	Ultra-high vacuum	Outgassing	
	Particulate contamination	Vacuum	Pump down	
	Cleanliness	Wafer	Gate valve	
	Particulate contamination		Shock*	
	Tribology		Vibration	
	Contact mechanics		Transportation	
			Electrostatic attraction	
			Van der Waals forces	
			Friction	
			Friction induced outgassing	

**Figure 2.** Overview of search terms and respective categories for particle contamination sources.

### 2.2 Eligibility criteria

Eligibility criteria are specified for the obtained results to further exclude irrelevant papers.

- The study mainly focuses on mechanisms that are aimed to be used in high-precision systems.
- The study mainly focuses on papers describing mechanisms designed for small-scale centimeter or millimeter-scale objects.
- The study only focuses on compliant gripper and positioning mechanisms.
- Selected papers should be accessible and available online.
- Selected papers should be written in English or Dutch language.

## 3. Classification

This literature review focuses on strategies for how a millimeter or centimeter-scale object can be grasped and micro-positioned. A given boundary condition is that this process is performed in an ultra-high vacuum where particle generation is a high-performance risk. Therefore, this literature review will describe solely compliant positioning stages and grippers mechanisms. Compliant mechanisms are monolithic structures, this characteristic results in many advantages. When an input force is applied to the structure it experiences zero friction, no hysteresis, no backlash and no lubrication is needed [5]. These characteristics are highly beneficial for applications where highly restrictive particle contamination risks are involved.

The included literature in this paper revealed that no papers were found that describe an object handling mechanism that had a special focus on low particle generation. Therefore, this goal is decomposed into multiple categories which together give an overview of the most important areas.

First, two categories are included that describe literature about particle contamination sources in ultra-high vacuum. Here, the included literature focuses on the various sources of how particles are generated related to a compliant mechanism in an ultra-high vacuum system. The sources are divided into

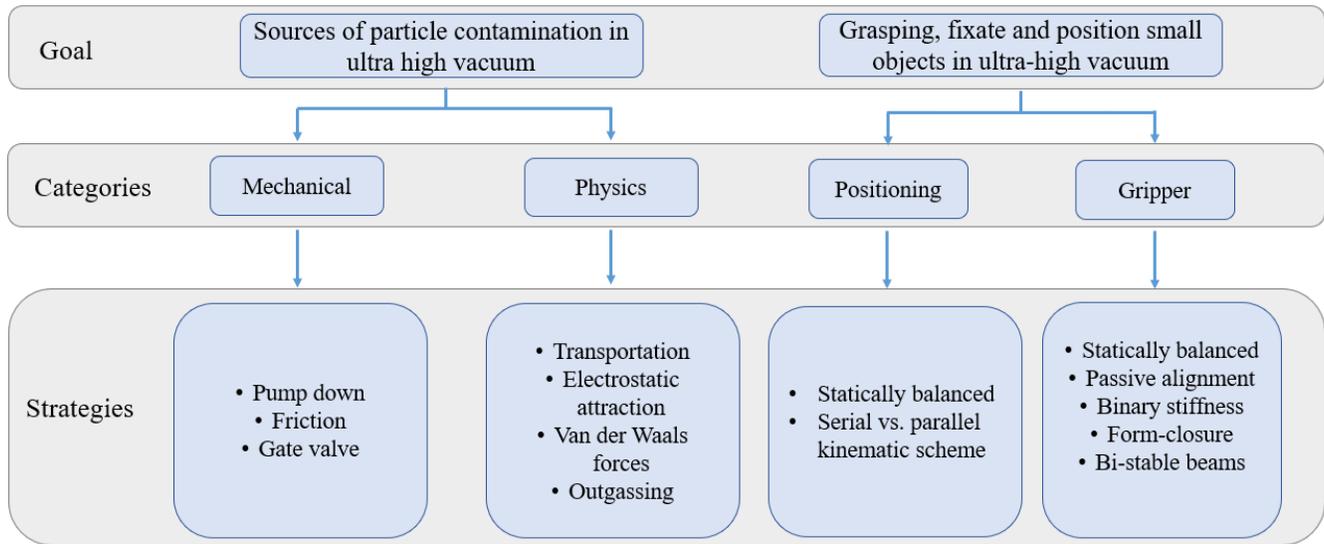


Figure 3. Classification of the results.

sources based on mechanical and physical processes. Second, there will be a focus on a compliant mechanism that is designed following design guidelines for particle-critical applications, and is cleaned and installed following standards designed for particle critical surfaces.

First, a category is added that describes strategies for micro-positioning an object. This category describes strategies for two degrees of freedom positioning stages that can translate in two axes. Second, a category is added that describes strategies for gripper mechanisms to temporarily fixate an object, the definition of a gripper is stated in chapter 1.2. Also, the literature should only focus on grasping objects at the millimeter or centimeter scale level.

## 4. Results

### 4.1 Particle contamination

In figure 4, different types of contamination are shown. Contamination can be divided into two categories, molecular and particle contamination. Molecular contamination consists of organic and molecular contaminants, ions and adsorbed molecules, nonvolatile residue and inorganic surface contaminants. Second, particle contamination consists of fibers, particles and all kinds of metals [3]. The sources of molecular and particle contamination can be based on mechanical and physics processes.

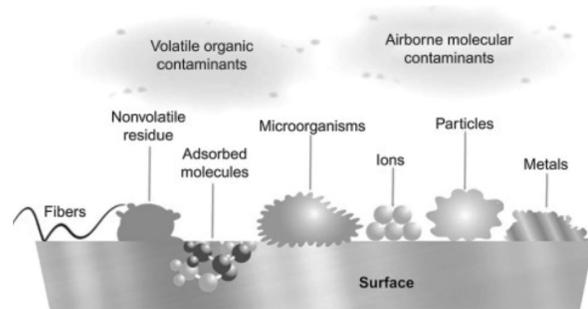


Figure 4. Types of contamination [3]

### 4.2 Mechanical particle generation

#### 4.2.1 Pump down

Pump down is when the pressure in the entire system is ambient and pumped to high vacuum pressure. Kim et al. [17] describes the effect of pumping the vacuum chamber from ambient to vacuum. Vacuum pumps itself generate particles that can travel into the critical environment. In the process of pumping the chamber to high vacuum pressure, the gas becomes supersaturated due to the rapid adiabatic expansion [18]. Here the chemical solutions in the gas exceed the equilibrium solubility, the condensation that follows will generate particles. Chen et al. [19] describes the effect that a higher humidity will lead to higher particle formation, in fact moisture is necessary for nucleation to arise. The mean sizes of the residue particles depend on the pumping speed and range between  $0.1 \mu\text{m}$  to  $0.25 \mu\text{m}$  [20].

#### 4.2.2 Gate valve

Vacuum load locks are used to load parts from an ambient to vacuum pressure environment. A vacuum load lock enables to keep the pressure in the main high vacuum chamber non-adjusted. Therefore, it is not needed to vent the cham-

ber to ambient pressure when a new part is installed. The transition period between ambient and vacuum pressure is the leading factor in contamination source of the system. A longer transition period leads to lower particle contamination of the vacuum system [21].

#### 4.2.3 Shocks and vibrations

Particles can be detached from surfaces when it is subjected to external forces [21, 22]. The external forces can be in the form of shocks, for example knocking of tools on metallic surfaces. Or second, mechanical vibrations can cause a particle to be released off a surface. After release of a very small particle, its trajectory can be described with the Brownian motion [23].

#### 4.2.4 Friction

Particles in a system are generated in various ways. First, friction between two surfaces results in a large amount of newly introduced particles in the system. Even when the parts are cleaned friction leads to dozens of hundreds of particles [21].

The relationship between friction and the wear volume can be divided into the following categories [24]:

- **Abrasive wear:** Asperities, which refers to the peaks of a surface, can penetrate the surface of the softer material, resulting in tearing the material.  
**Adhesive wear:** Tearing of material due to a sticking effect of two surfaces. The adhesive wear can be explained by the effect when the elastic limit of an asperity is exceeded under pressure, at this moment the asperities are welded together. The welding of the two materials can not be noticed on a macroscopic scale, but it can be observed on individual asperity level contacts. Adhesive wear is dependant on the contact surface area, increasing this area can drastically increase adhesive wear [25].
- **Surface fatigue:** Is present in situations where two surfaces have small translations over a prolonged cycle life. Or when a surface is rolling onto a second surface. Here the surfaces pressures are repeatably loaded and unloaded, over a prolonged time period changes to the surface can be noticed.
- **Corrosive wear:** Chemical processes on a surface of a material and corrosion of a surface. [24]

With equation 1 the wear volume can be experimentally found. Here is  $k$  the wear rate which need to be experimentally determined for a given material and the respective roughness, lubricant application and geometry [24]. Formula 1 is valid for modeling abrasive wear and adhesive wear, but the experimentally determined wear rate factor  $k$  is different for both wear mechanisms.

$$V = \frac{kFs}{\sigma} \quad (1)$$

Where:

- $V$  = Wear volume
- $k$  = Wear rate
- $F$  = Normal load
- $s$  = Sliding distance
- $\sigma$  = Hardness of softer material

In Carey et al. [2] it is reported that during the handling and fixating of a wafer in the semiconductor particles can be generated due to gripper pins coming in contact with the backside of a wafer. Subsequently, these particles are able to be transported from the backside of the wafer to critical surfaces in the system, in the described case this is the top side of the wafer.

Besides spontaneous outgassing and desorption of materials as earlier described in chapter 4.3.1 exists another mechanism that is able to release gas to its surroundings. Mechanically affected surfaces, for example due to friction, are a source of gases in a vacuum environment, this phenomenon is also referred to as mechanically stimulated outgassing (MSO) [26, 27]. Repa et al. [26] describe that mechanically stimulated outgassing depends on friction speed, frictional load and the concentration of the dissolved gases in the materials. Second, the paper concludes that it is not affected by the pressure in the vacuum chamber. The gas specifications released by the material contained mostly species dissolved in the material and material of what the tool was made of.

Nevshupa et al. [27] report that the main desorbed gas due to friction induced outgassing is hydrogen. Other commonly desorbed gasses are CO, CO<sub>2</sub>, H<sub>2</sub>O, Ar and CH<sub>4</sub>.

### 4.3 Physics particle generation

#### 4.3.1 Outgassing

Factors associated with outgassing can be divided into two main categories [28]:

1. Outgassing from bulk materials: Molecules of the material itself diffuse through the volume of the material and enter its surface, eventually desorbing from it.
2. Previously adsorbed molecules: Molecules that have most likely entered during venting of the system are absorbed by the surface of the material. When the system is pumped down to a vacuum these molecules desorb again from the surface.

The outgassing rate can be described with the differential equation as given in equation 2, it can be approximated with formula 3 [29]:

$$\frac{d^2 n_v}{dt^2} + \left( \frac{\vec{v}}{4V} (\alpha A_s + A_p) + \frac{1}{\tau} \right) \frac{dn_v}{dt} + \frac{\bar{\alpha} A_p n_v}{4V \tau} = 0 \quad (2)$$

$$\dot{Q} = \sum \left( \frac{i}{h} \right) \bar{\alpha} \quad (3)$$

Where:

- $\dot{Q}$  = Outgassing flow rate
- $\alpha_{1h}$  = Outgassing rate at 1 hour of the material
- $\alpha$  = Decay constant of the material
- $A$  = Surface
- $t$  = Time

Reinhardt et al. [30] describe that outgassing is a significant problem in ultra-high vacuum systems. Systems operating at an ultra-high vacuum ( $10^{-6}$  Pa -  $10^{-10}$  Pa) are highly sensitive for outgassing. The most obvious effect of contaminants in the vacuum chamber is that it risks the performance of a system. Second, contaminants in the vacuum chamber increase pump down time to reach ultra-high vacuum. Most of the contamination originates from different sources used inside the vacuum chamber. First, gases originating from the construction materials that are used mostly consist of  $H_2$ , CO and hydrocarbons ( $H_xC_y$ ). Second, contamination from permeation of gases mostly consists of  $N_2$ ,  $H_2O$  and  $O_2$ . Third, molecules can be adsorbed on surfaces like the chamber walls and eventually be desorbed to the environments, these can be for example moisture and solvents.

In section 4.2.4 another outgassing mechanism is considered. Here a mechanism is described where gas is able to be released to its surroundings induced by friction.

### 4.3.2 Transportation

As described in the previous chapters multiple factors play a role in introducing particles into the vacuum system. This section describes the effect of airborne particles in a vacuum system. At first, when a vacuum system is vented and exposed to ambient air, various new particles are introduced in the vacuum system. It can take multiple days for the particles to settle down on a surface [21]. Especially small particles can take a long time to settle down, due to the low mass and friction. In high vacuum environments, particles settle faster, this is due to the significant decrease in drag force. The settling velocity can be calculated by using the drag force with the gravitational force for a highly laminar airflow  $Re < 1$ . Formula 4 and 5 can be used to explain the settling time of a particle [17].

$$V_s = \frac{\rho_p d^2 g C}{18\mu} \quad (4)$$

$$Re = \frac{\rho_a V_s d}{\mu} \quad (5)$$

where:

- $V_s$  = Settling velocity of particles
- $Re$  = Reynolds number
- $\rho_p$  = Density of particles
- $\rho_a$  = Density of air
- $d$  = Diameter of the particles
- $g$  = Gravitational acceleration
- $C$  = Slip correction factor
- $\mu$  = Viscosity of air

In figure 5 settling velocity versus particle diameter for multiple pressures are plotted using equations stated above.

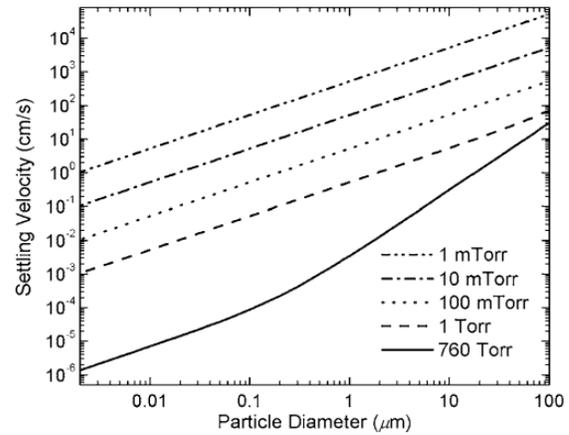


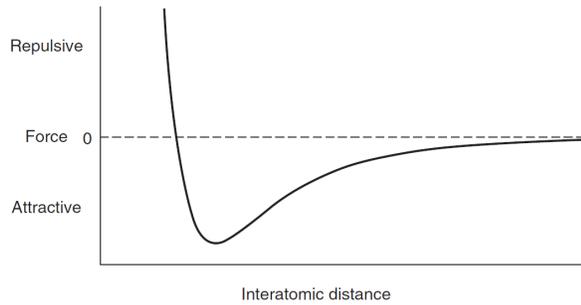
FIG. 1. Particle settling velocity at reduced pressures.

**Figure 5.** Settling velocity vs. particle diameter for different pressures [17].

For particles with a very small mass, the Brownian motion describes its trajectory path. Brownian motion describes the phenomenon that the particles constantly collide with other molecules. The collisions between the smallest particles can transfer enough energy to make a trajectory alteration of the particle possible [23]. Due to these collisions, it is possible for a particle to reach every open surface within a system.

### 4.3.3 Van der Waals forces

Van der Waals forces enable it for non-conducting particles to bind to surfaces [4]. A somewhat larger force than the gravitational force is needed to overcome the Van der Waals forces and therefore come loose from the surface [21]. Van der Waals forces are created every time two bodies come in close proximity of each other. The attraction forces increase when the distance between the two bodies decreases. When the distance between the two bodies decreases further there is a point that the bodies will be repulsed against each other, this is the result of either electron or photon repulsion. This relation is schematically shown in figure 6.



**Figure 6.** Van der Waals repulsive and attractive force versus distance between two bodies [21]

Van der Waals forces are considered to be the weakest form for a non-conducting particle to bind to a surface. The Van der Waals forces for a spherical particle binding to a flat surface can be described with formula 6:

$$F_{adv} = \frac{Ad}{12x^2} \quad (6)$$

Where:

- $F_{adv}$  = adhesive force due to van der Waals interaction
- $A$  = Hamaker constant, depends on material particle and surface
- $d$  = Diameter of a spherical particle
- $x$  = Distance between the particle and the surface

#### 4.3.4 Electrostatic attraction

In Walker et al. [4] the effect of electrostatic attraction for particle contamination is described. Electrostatic attraction (ESA) is a force that can bind particles to a non-conducting surface. The electrostatic attraction force can be described with equation 7. The effect of contamination due to electrostatic attraction is most present for particles with a diameter that does not exceed  $5 \mu\text{m}$ . The charged particle and surface are attracted by Coulombic attraction, but it is not necessary that both particle and surface are charged. It is common that charged particles can be attracted to neutral surfaces:

$$F_e = \frac{K_E q^2}{x^2} \quad (7)$$

Where:

- $F_e$  = Electrostatic attraction force
- $K_E$  = Constant ( $9.0 \times 10^9 \text{Nm}^2/\text{C}^2$ )
- $q$  = Charge of a particle in Coulombs
- $x$  = Particle diameter

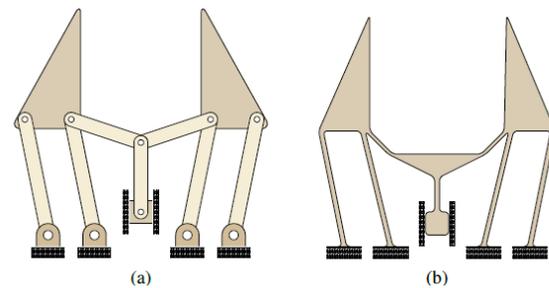
If a surface is contaminated with multiple charged particles, an electrostatic field can be created that will accelerate the deposition of particles that are oppositely charged, resulting in leading to an acceleration of contamination. There are

three factors concerning the potential for particle contamination. First, the contamination is proportional to the charge and concentration of the particle. Second, The charge per area on the surface. Third, the time the surface is exposed to the particles [4].

#### 4.4 Compliant mechanisms

In this paper only mechanisms are described that use the principle of compliant mechanisms. To prevent confusion all overlapping results concerning the compliant mechanism are described in this chapter. Second, multiple strategies use statically balancing of the mechanism, which is a unique feature a mechanism can have. General properties of statically balancing that can relate to all the statically balanced mechanisms as described in this paper are also described in this chapter.

Compliant mechanisms are monolithic structures, these characteristics result in many advantages. When an input force is applied to the structure it experiences zero friction, no hysteresis, no backlash and no lubrication is needed [5]. In industries where particle generation, micro-precision movement and vacuum environments are important subjects these characteristics play a big role in performance. Compliant mechanisms make use of the deformation of all elements of its monolithic structure, in this way the mechanisms can transmit a motion or force based on an input force [31]. If all the motion of a mechanism follows from deformations of this same mechanism it is said to be fully compliant [32, 33]. In figure 7 an example is shown where a gripper is first visualized where it is constructed with rigid links and a compliant version.



**Figure 7.** Gripper mechanisms a) Constructed out of rigid links. b) Constructed as a fully compliant monolithic structure[33].

##### 4.4.1 Statically balanced compliant mechanisms

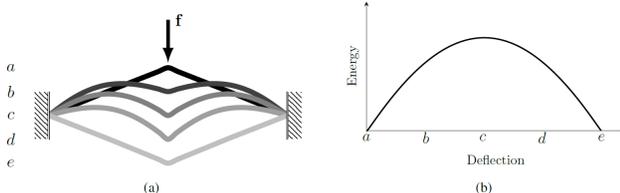
In Gallego [33] statically balanced compliant mechanisms are extensively described. A drawback of compliant mechanisms is that it stores the potential energy of their elements as strain energy, which affects the force input-output relation. This can be counteracted by reintroducing this 'lost' potential energy into the system, these special kinds of compliant mechanisms are also referred to as statically balanced compliant mechanisms. In short, static balancing of a mechanism results in a conservative motion, where for a certain part of the mecha-

nisms range the total potential energy is kept in equilibrium.

Statically balanced compliant mechanisms are capable of improving the energy efficiency of a mechanism [34]. In gravity compensating systems the gravitational force on an object is balanced against its weight. Here very little external force is needed to move potentially very large masses [35]. Second, by restoring the force input-output relation great force feedback of a mechanism can be obtained [16]. At last, statically balanced mechanisms show great characteristics in vibration isolation. In a hypothetical setting, if a mechanism is perfectly balanced and therefore has zero stiffness, the natural frequency of this mechanism is equal to zero. This leads to a mechanism that will not be oscillating when they are perturbed [33].

Statically balancing a compliant mechanism can be done in two ways, using bi-stable beams or pre-loading a member of the mechanism. Note that for rigid-links mechanisms there are more approaches one can take to statically balance a mechanism. Examples can be found where springs are used to create a statically balanced rigid link mechanism [35, 36].

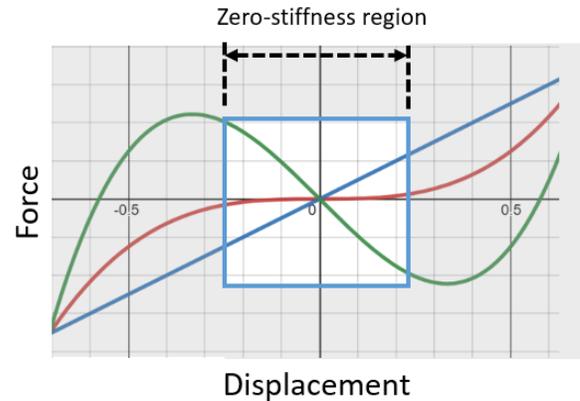
To make a beam a bi-stable beam it should first be preloaded by compressing the beam and clamping and fixating it at both ends. In figure 8 a double clamped preloaded beam and many deflection stages are shown. A bistable beam has two stable points, this is in the figure point *a* and *e*. And it has an unstable point, this is point *c* in the figure. In the first deflection stages *a* to *c* a force needs to be provided to the beam in able to deflect, therefore potential energy is added to the beam as shown on the right side of the figure. After the unstable point the beam will deflect into its second stable form shown as stage *e* in the figure. In this deflection range the beam can deliver a force and the potential energy of the beam will decrease until it reaches its stable form. This latter part can be used to statically balance a mechanism. Other principles to statically balance a system are based on the same principles, where an element delivers potential energy by going from an unstable position to a stable position [33, 37].



**Figure 8.** a) Bi-stable beam and its deflections for different stages. b) Potential energy on the beam during the deflection stages [33].

In figure 9 arbitrary force-displacement plots are visualised for three members. The blue line can represent a leaf

spring where the displacement of an element scales linearly with the applied force. The green line can represent a bi-stable member, where as described above at a certain moment the displacement is beyond its unstable point and experiences a negative stiffness. In this negative stiffness part, the member will provide a force on a given range of displacement. At last, the red line represents a mechanism where previous members and their relative stiffnesses are combined. The end result is a mechanism that can experience a zero-stiffness or constant force region for a certain displacement range [37].



**Figure 9.** Force vs. deflection of an arbitrary statically balanced mechanism. Green: mechanism 1; Blue: mechanism 2; red: combined mechanisms

#### 4.5 Positioning stages

A distinction can be made in micro- and macro positioning stages. In general, there is a trade-off in performance and capabilities in micro or macro positioning stages. Actuators that are used in macro positioning stages can be characterized twofold. Either the mechanism has a large range but limited precision and accuracy, or a small range and capable of high precision positioning [54].

In applications where both large travel range and high-level precision and accuracy are needed, multiple stages can be added in series [6, 55]. Here one stage can translate a platform multiple centimeters, with a low-level precision [56]. The second stage has a short range, but is capable of high-level precision and accuracy. Second, using multiple stages make it possible to develop a high-speed and high-precision system [6].

Most papers concerning micro-positioning stages are about stages that can perform two-translational movements. Thereafter, 3-DOF stages are common where a rotational degree is added [5]. Wang et al. [57] describes that positioning stages can be divided into two schemes to achieve the translation in the *x*- and *y*-axis and their respective drawbacks. The stages can be divided into parallel- and serial kinematics schemes. Serial kinematics when compared to parallel kinematics shows shortcomings in terms of large cumulative

Ref	Range ( $\mu\text{m} \times \mu\text{m}$ )	Design area (mm $\times$ mm)	Coupling X and Y-axis (%)	Coupling error ( $\mu\text{m}$ )	Out of plane deflection( $\mu\text{m}$ )	Ratio design- vs. Range area	Actuation method
[38]	20x20x10 <sup>3</sup>	224x254	0,043	8,6	-	142,24	Linear
[39]	10.000x10.000	125x125	0,03	2,5	20	163,84	Voice coil
[40]	180x180	-	0,83	1,5	-	-	piezo
[41]	1000x1000	24x24x5	-	-	-	576	piezo
[42]	10x10x10 <sup>3</sup>	-	-	-	-	-	-
[43]	700x700	-	-	zero	5,1	-	-
[44]	180x180	4x4	3	5,4 $\mu\text{m}$ @180 $\mu\text{m}$ / 3 $\mu\text{m}$ @90 $\mu\text{m}$	-	493,81	Combdrive
[5]	50x50	170x170	0,91	0,455	0,05	11560000	
[45]	8x8	-	0,6	0,048			
[46]	100.9x100.2	-	0,54	0,544	-	-	piezo
[47]	125x125	131x131	0,6	0,6		0	
[48]	127x127	68.5 $\times$ 68.5 $\times$ 68.5	5.4	6.9	-	0	piezo
[44]	225x225	4x4	6.75	3	-	316	Combdrive
[49]	2300x2300	220x220	0,92	2,3	0,023 @100g	9149	Voice coil
[50]	135 x 135	27x27	-	-	-	40000	Piezo ceramic
[51]	22x22x10 <sup>3</sup>	37x37	-	-	-	2,25	Piezo
[52]	60x60	100x100	0,24	0,4	-	2777777	Piezo
[53]	100x100	40x40	0,02	0,02	-	160000	Piezo

**Table 1.** Overview of compliant positioning stages and their most important characteristics.

errors, low eigenfrequency and high inertia.

A wide variety of two-dimensional compliant stages are described in the literature. All of these described stages describe a compliant mechanism that makes use of the same design strategy in order to manipulate the position of the stage. All obtained stages use a leaf spring design for manipulation of the stage. In table 1 an overview has been given of many mechanisms and their differentiating characteristics. More mechanisms can be found with more or less the same characteristics, in the table it is tried to exclude mechanisms that have many overlapping characteristics from the ones that are already stated.

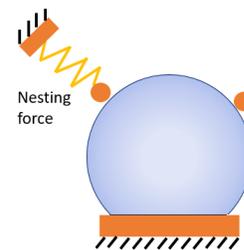
In a situation where many stages are needed it can be beneficiary to statically balance the positioning stages. In these stages the required actuation force can be significantly reduced [37], this would result in a lower required wire gauge. Electrical wires are insulated by a rubber porous material, which have high outgassing rates. A lower wire gauge results in a decrease in surface area, using equation 3 this will linearly decrease outgassing rates. Since the surfaces of one wire is small the effect of one statistically balanced stage is possibly negligible, but when many are present in a system the total outgassing flow can get significant.

#### 4.6 Micro-gripper

In this section multiple strategies are described on how these tasks can be performed, the described papers only focus on a millimeter or centimeter scale objects.

##### 4.6.1 Alignment using externally applied force

In the semiconductor industry, a similar procedure needs to be performed. Here the patterns and features of the wafer need to be measured with an atomic force microscope for quality control. In Duenner et al. [10], a design is proposed to fixate and align a wafer for inline atomic force microscope metrology in nanoscale manufacturing. The paper shows that it is possible to place and align the wafer within one minute with translational repeatability of 1.4  $\mu\text{m}$ . Second, a positioning accuracy of 1  $\mu\text{m}$  is achieved. To fixate the wafer three pins and a nesting force are used in order to remain the contact with the pins, see figure 10.



**Figure 10.** Alignment of a wafer using three contact points and a nesting force.

A flat circular object laid down on a flat surface has three degrees of freedom, two translational and one rotational. It is assumed that the gravitational force on the wafer is sufficient for constraining the out-of-plane rotations and translation. To fully constrain the circular object three pins are required. Each pin creates a line of action on the rigid body, an intersection of the line of actions creates an instantaneous center of rotation (ICR) where a rigid body can rotate about [58]. To constrain the rotational degree of freedom two of the pins should have

parallel lines of actions (LOA) in order to have an instantaneous center placed at infinity. To constrain the translational degree of freedom two of the three pins should intersect with each other to create an instantaneous center located at the wafer [10, 12].

The nesting force should be able to restore and remain the contact with the three pins contacting the wafer. Losing contact of the wafer with one pin implies a rotation about an instantaneous axis of rotation that is determined by the two remaining contacts. The relationship between the reaction forces of the pins and the nesting force is given by equation 8. The nesting force will exert a moment on the wafer in a way it is capable of providing a counter moment and therefore restoring the contact with the pin. It is important the nesting force is placed at a location on the wafer where it can provide a restoring counter moment for all three pins. The created restoring moment about the instantaneous centers can be calculated by equation 9 [12]. In figure 11 a schematic overview of the used variables in the equations is given.

$$\begin{bmatrix} f_3 \\ f_L \\ f_R \end{bmatrix} = F_n \begin{bmatrix} \cos \phi \sec \theta \\ \frac{1}{2}(\sin \phi - \cos \phi \tan \theta) \\ \frac{1}{2}(\sin \phi + \cos \phi \tan \theta) \end{bmatrix} \quad (8)$$

$$M_{ICR1} = \frac{1}{2} F_n (r_{rx} - r_{lx}) \sec(\theta) \sin(\theta - \phi) \quad (9)$$

The object is pushed into the nested position, during this procedure particles are generated. Following section 4.2.4 'Friction', abrasive wear is present at every contact point and depends on the positioning error distance and the reaction forces created by the nesting force. Second, using equation 8 and 9 and the paragraphs above it is concluded that the magnitude of the reaction forces and restoring moment depends on the position of the pins. The magnitude of the restoring moment needs to be maximized to obtain the optimized positioning performance. The number of particles that will be generated can be minimized by optimizing the position of the pins and creating a large restoring moment whilst keeping the required normal force as low as possible. Third, large contact surface areas create large adhesive forces which result in adhesive wear besides the already present abrasive wear. In the described study, the wafer is placed on a flat platform and slides into its nested position. Using this approach creates large adhesive wear and can be prevented by reducing the contact surface area.

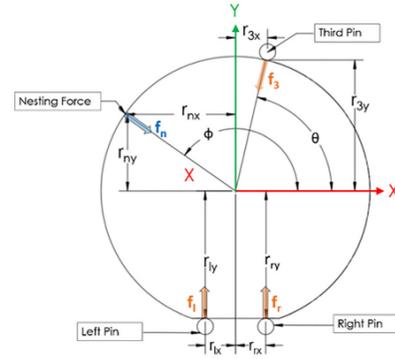


Figure 11. Variables conventions [10].

#### 4.6.2 Passive alignment

Kinematic coupling is a passive alignment and fixating method, which can be capable of accuracy in the order of  $1 \mu\text{m}$  and repeatability in the order of multiple microns [59]. Kinematic couplings take advantage of having a small structural loop, which generally leads to better alignment performance [8]. In general, two alignment methods are used, kinematic and elastic averaging. In kinematic averaging the system is said to be statically determinant, in this situation there are an equal number of contact points as the number of degrees of freedom that are constrained. In kinematic systems, or as they are often recalled to as exact constraint systems, a kinematic coupling is used between the elements. An example is shown in figure 12 where in the left column V-grooves are shown and in the right column V-groove, a trihedral and flat plat are used to align two components. In elastic averaging the system is said to be overconstrained, but every member in the system is able to perform small deflections. In the situation, a force is applied to clamp the system in its place the elements can deform and the errors will average. An example of an elastic averaging system is the well-known LEGO blocks [8].

The performance of a kinematic couple is limited by the tolerances of the machined product. Elastic averaging can also be used to achieve better performance, Factors like accuracy, repeatability and higher load capacity can be improved by using multiple relatively flexible elements [59]. Second, In Slocum et al. [8] it is stated that spring-loaded pins or elements can be used to decrease misalignment.

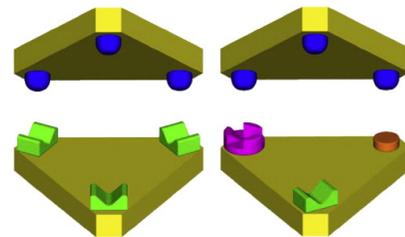


Figure 12. Kinematic couplings Left side) Kelvin clamp Right-side) V-groove [8].

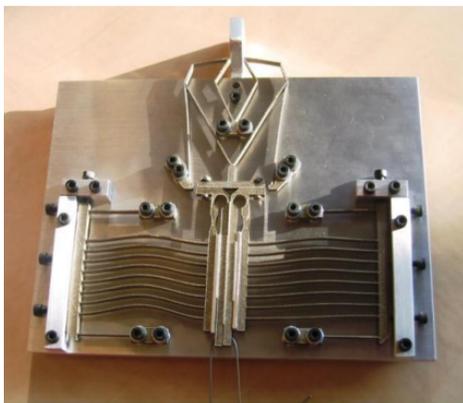
The working principle V-groove or Kelvin clamp resorts

to sliding of the object's pins to the position with minimal potential energy. Both the V-groove and Kelvin clamp have at least three contact points during the positioning of the object, here abrasive wear is present and depending on the reaction forces will create a significant amount of particles. Second, both examples use spherical contact points with small contact surface areas, this is beneficial since it eliminates adhesive forces as much as possible.

#### 4.6.3 Statically balanced gripper

Statically balanced compliant grippers are capable of improving the energy efficiency of a mechanism [34]. In Hoetmer et al. [37] a design method is described where a gripper is statically balanced using a building block approach, the designed gripper is displayed in figure 13. Using this design method a gripper is designed with the goal to lower the actuation force that is needed to operate the gripper. At last, the theoretical approach that is used in the paper should result in a force-displacement behavior of the entire mechanism that is near zero, but most often it shows a small error. The dimensions of the mechanism are tuned manually using an iterative procedure to obtain the best possible result.

The statically balanced gripper shows that the required actuation force is significantly decreased for opening and closing the gripper. The non-balanced gripper needed 4.5N for actuation, the balanced gripper need -1.2 N for actuation. This also shows that the gripper is overbalanced, this can be caused and mitigated by multiple factors like improving manufacturing tolerances and decreasing misalignment. Second, both the balanced and the non-balanced grippers show a strong hysteresis, which can be partly explained by the backlash of the force transducer.



**Figure 13.** Statically balanced compliant gripper [60].

In Lamers et al. [16] a statically balanced gripper is described, see figure 14. In this study a gripper is designed that needs a perfect ratio between the applied input force and delivered clamping force. An input force is applied, and part of the applied energy is stored as elastic strain energy. This elastic strain introduces a stiffness that will affect the

input/output force relation of the gripper, statically balancing the gripper restores the affected input/output force relation.

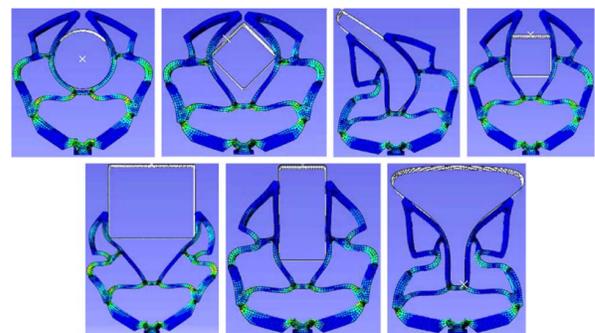


**Figure 14.** Statically balanced gripper with a perfect ratio between applied input and clamping force [16].

A statically balanced gripper has the same benefits with respect to particle generation when compared to the previous and in more detailed described benefits for statically balanced positioning stages in chapter 4.5. Lowering the required actuation force can decrease the outgassing rates originating from the wires feeding power to the actuators.

#### 4.6.4 Form-closure gripper

In Petkovic et al. [11] a gripper mechanism is described that can morph into many different shapes of the objects that are being grasped, as shown in figure 15. Conventional grippers that can grasp a wide variety of objects with different shapes have multiple actuators and sensors. This mechanism is passively underactuated, in other words, fewer active inputs are needed than the number of degrees of freedom of the gripper mechanism to actuate the open and close motion of the gripper and so to fully constrain the object. An added benefit of using an underactuated mechanism is that it uses fewer sensors and actuators, thus reducing the number of control variables. Reducing the number of control variables results in a less complex control system.

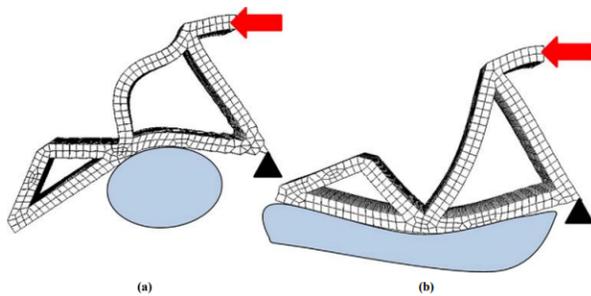


**Figure 15.** Form-closure gripper [11].

In figure 16, the paper also shows that the mechanism is able to morph into a convex and concave shape. The two-

finger gripper mechanism as shown in the above figure 15 shows unstable behavior during the grasping and holding of an object. A multi-fingered gripper with at least three or more fingers greatly enhances stability. Also, the described gripper is made of a flexible material in order to allow the gripper to morph into different shapes. In this case, the gripper is made of silicone rubber Elastosil R420/70 MH E. Using a flexible material makes it so that the gripper is unable to hold heavy objects.

A gripper that is made of flexible material is of a rubber or silicone-like substance, these materials are very porous. As described in chapter 4.3.1 'Outgassing', all porous materials have notoriously high outgassing rates. A form-closure gripper would result in a significant contamination source due to outgassing.



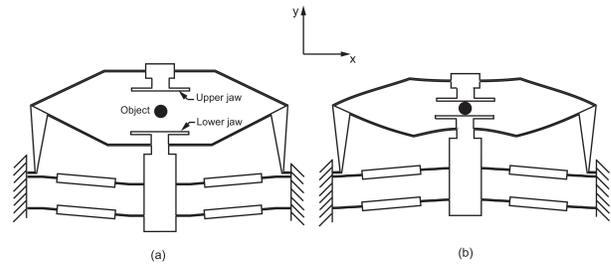
**Figure 16.** Gripper morph shapes convex (a) and concave (b) [11].

#### 4.6.5 Bi-stable locking mechanism

In Chang et al. [13] a gripper is described that can maintain the grasp on an object while it no longer consumes any energy. This is done by using a bi-stable mechanism that is also referred to as a double-arch suspension [33], here the two stable positions of the mechanism resemble the closed and open state of the gripper. In figure 17 the gripper mechanism as described in the paper is shown. The gripper is able to grasp objects with diameters ranging from 4 to 8 mm. The objects should be grasped close to the center of the gripper to prevent a tilt angle of the jaw.

Bi-stable beams snap through when it goes from an unstable position to a stable position. This snap accelerates the jaws and creates an impact force when it grasps an object. Following equation 1 in chapter 4.2.4 'Friction' the relationship in applied force and particle generation is explained. The extra force will result in an unnecessary increase in particle generation, but if there is no slip between the object and gripper than no proof is found that particles will be generated. In section 4.2.4 'Friction' contamination due to well-controlled wafer handling is also described. Here during a well-controlled grasp of the wafer particles are generated, this shows that particles are still generated although a no-slip condition could be assumed. This could implicate that an uncontrolled grasp when using a bi-stable beam gripper could have a negative

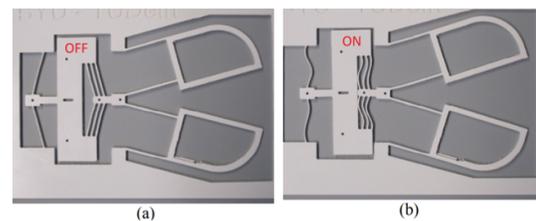
effect on particle generation.



**Figure 17.** Bi-stable gripper [13].

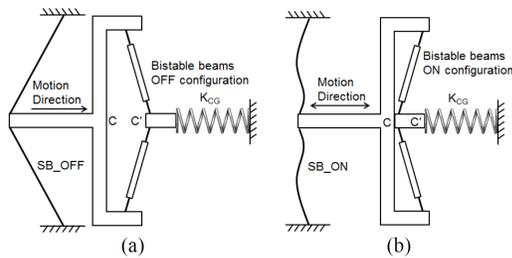
#### 4.6.6 Switchable binary stiffness gripper

In Plumiers et al. [15] a micro-scale gripper mechanism is described that has a switchable binary stiffness gripper. In figure 18 the gripper is displayed in two different states. The mechanism can switch between a state where the entire mechanism has high stiffness, thus constraining any motion of the gripper. In the second state, the gripper is in a soft state where the gripper is able to translate. In this state, the mechanism is also statically balanced in order to reduce the operating force for opening and closing the gripper. The paper describes that the required actuation force was reduced by 91%, from 4.4 N to 0.4 N. The binary switchable stiffness characteristic is added due to the high risk that is involved in damaging a zero-force structure during handling. Similar concepts are found where this method is also successfully incorporated [61, 62, 63].



**Figure 18.** Configurations of the gripper in the OFF (a) and ON (b) state [15]

In figure 19 a simplified representation is shown for the binary statically balanced mechanism. To switch between the states a motion is applied as an input to the system. This will result in buckling of the beams on the left side of the figure. The gripper mechanism itself, represented as  $K_{CG}$  has a relatively large stiffness on the horizontal axis. Therefore, the bi-stable beams on the right side will move into their other stable position. In this position, it will contribute an opposite stiffness compared to the gripper mechanism ( $K_{CG}$ ).

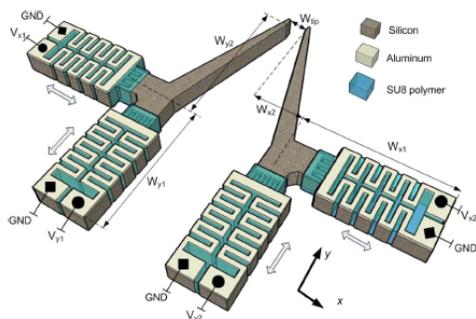


**Figure 19.** Simplified representation binary stiffness mechanism in a) Off state b) On state [15]

#### 4.6.7 Micro-gripper based on two independent positioning stages

A mechanism that can grip small objects is in its basics an extended feature of a micro-positioning stage. Especially in compliant mechanisms, many characteristics in gripper systems are overlapping with positioning stages. In this chapter, mechanisms are described that can grasp an object and is able to position the object with high micron-scale precision. Multiple papers in the literature describe a gripper mechanism that can independently translate both gripper arms. It should be noted that in the obtained papers micro-positioning was not the exact goal of the authors, but rather describe the ability to grasp and further rotate the object.

In Duc et al [64] a micron-scale gripper is described that can grasp objects between  $6 \mu\text{m}$  to  $40 \mu\text{m}$ . The described gripper is displayed in figure 20 and uses four silicon-polymer laterally stacked electrothermal microactuators. The gripper can translate the gripper arms in the x- and y-axes up to 17 and  $11 \mu\text{m}$  respectively [64, 14].



**Figure 20.** Two DOF micro-gripper [64]

In Speich et al. [14] a mechanism is described that is able to translate the gripper arms with a 10 mm range of motion with micron precision. The mechanism is constructed by a small compliant mechanism that is able to grasp the object, thereafter a rigid-link mechanism is constructed to manipulate the position of the gripper. The entire size of the structure is in the range of  $30 \text{ cm}^3$ . This mechanism focuses on manipulating an object rather than positioning it. The discrepancy between manipulating and positioning is most notable when evaluating the closed-loop stability of a mechanism. In manipulation

tasks, the mechanism can utilize kinematic constraints coming from the surroundings to guide the motion of the manipulator. In positioning tasks, these kinematic constraints are mostly ignored, for example the notion of disturbance rejection.

## 5. Overview relationship strategies and particle generation

In this section an overview is given of the relationship between particle contamination and object handling strategies which were obtained in this literature review. In table 2 for each described strategy, a shortened summary of the results is described and is subdivided into negative and beneficial relationships with respect to particle contamination.

Subject / Strategy	Negative relationship	Beneficial relationship	Ref.
<b>Positioning stages</b> <i>Statically balancing</i>		Lowering the required actuation force leads to a smaller required wire gauge. A smaller wire gauge leads to lower outgassing rates. If this effect large lengths decrease in outgassing can become significant	[28, 29]
<b>Gripper</b> <i>Statically balancing</i>		Identical to statically balanced positioning stages.	
<i>Alignment using externally applied force</i>	1) During the alignment process abrasive wear is present at every contact point following equation 1. The wear depends on the positioning error and reaction forces. 2) The study describes a wafer laid upon a flat surface. Large contacting surfaces greatly increase adhesive wear.	The magnitude of the restoring moment needs to be maximized to obtain the optimized positioning performance. Abrasive wear can be minimized by optimizing the position of the pins and creating a large restoring moment whilst keeping the required normal force as low as possible.	[2, 10, 12, 24, 25, 58]
<i>Kinematic couplings</i>	The working principle V-groove or Kelvin clamp resort to sliding of the objects pins to the position with minimal potential energy. Both the V-groove and Kelvin clamp have at least three contact points during the positioning of the object, here abrasive wear is present and depending on the reaction forces will create a significant number of particles.	Kelvin clamp and V-groove use spherical contact points with small contacting surface areas, this is beneficial since it eliminates adhesive forces as much as possible.	[8, 24, 25, 58, 59]
<i>Form-closure</i>	1) Requires the use of flexible material, which is in general a rubber or silicone-like substance, which are very porous materials. These materials have notoriously high outgassing rates. 2) The described design results in a grasp with large contact surfaces. This leads to significant adhesive wear. The design could be adapted to obtain a form-closure grasp with only point contacts	A form-closure mechanism is passively underactuated, therefore fewer actuators are required than the number of DOF that is needed to actuate the open and close state. Reducing the number of required actuators leads to fewer wires. The rubber insulation of the wire has high outgassing rates.	[11, 24, 25, 28, 29]
<i>Bi-stable locking</i>	The bi-stable beam accelerates when going from its non-stable to stable position. This will create a relatively high impact force on the object. This can create an increase in particle generation depending on the sliding distance between the gripper and the object.		[13, 24, 25]

**Table 2.** Summarized overview of the relationship between described object handling strategies and particle generation.

## 6. Discussion

### 6.1 Main findings

This literature review had the goal to provide a bridge between particle generation sources and positioning and gripper strategies. As stated before, limited information was available that resembled a similar goal or describe design rules for object handling operations. Therefore, the goal was divided into multiple categories in order to generate a clear classification of the multiple topics. First, various sources for particle generation were described, which was later used to provide a bridge between the object-handling strategies. For each object handling classification, strategies were described that circumscribe different approaches the selected papers took.

This literature study focused on object handling in a vacuum environment, the benefit of object handling in a vacuum instead of venting the system and installing an object could be demonstrated by the results of section 4.2.1 and 4.2.2. Each pump down to vacuum pressures results in many new introduced particles. In current practices the industry uses a load lock that can load a new part from ambient pressures without alternating the vacuum pressure in the main system, this addition to the system helps reduce new particles. An additional particle contamination source that is not described in the mentioned chapters is ordinary 'dirty' air originating from outside the system and entering the system.

In table 2, an overview of the obtained relationships between the object handling strategies and particle contamination is given. This table shows that only a plausible link can be made with abrasive and adhesive wear and outgassing. Although other described contamination sources mentioned in section 4.1 do not have a direct link with the object handling strategies, they are useful to provide insight into how contamination can eventually result in a performance loss of a product. In subsection 4.3.2 it is described that due to the Brownian motion a small particle can reach every open surface in a system. This enables contamination generated far away from critical surfaces to reach these surfaces. A full enclosure of the critical surface or particle-generating volumes could improve performance by restricting particle transportation. Also, as described in this subsection it can take multiple days for a particle to settle down on a surface. Therefore, the effect of generated contamination in a system can take a long time before it can be noticed.

In an interview with a contamination expert within VDL ETG [25] a rule of thumb is given that the contacting area needs to be kept minimal to minimize adhesive wear. Using this rule of thumb, nothing is said about the relationship between applied normal force and adhesive forces. On the contrary, equation 1 can also be used to describe adhesive wear and here the applied normal force does have an influence on the adhesive forces. One could assume that spreading the load over a larger surface would lower this applied force and

following the equation leads to lower adhesive forces. The influence of contact area in this formula is not completely clear, since the contact area is not used as a parameter. These differences could possibly be explained due to the fact that the literature that describes equation 1 does not specifically focus on situations in ultra-high vacuum applications and low-wear rates. The rule of thumb of keeping the contacting surface area minimal is based on experience with particle generation in vacuum systems with low-wear rates and experiments performed by VDL. Therefore, this rule of thumb gives a more confident assumption to model and predict adhesive wear than the given equation 1.

In literature, very little is publicly available about the interaction of two objects when grasping and the resulting particle contamination due to different design choices. No information was found that described the failure mode in more detail. In Tribology literature, a type called fretting wear can be found that describes a repeating displacement of two surfaces with a small amplitude in the range of 100  $\mu\text{m}$ , which can lead to damage over a prolonged period [65, 66]. Possibly, when grasping an object a vibration between the object and gripper is induced, which can explain the wear that is still present in a stable grasp as described in Carey et al. [2]. The particle generation due to grasping could also be argued due to the manufacturing tolerances and accuracy of the actuators of the gripper. In the situation where one constrain slightly later come in contact with the body, a motion can be set in place between the two gripper and object.

### 6.2 Limitations

Multiple limitations were set or found that are present in this literature review. In this literature review, the aim was to narrow down the treated subjects as far as possible, still it has an inherently broad character. Therefore, the search plan may exclude search terms that could give interesting new insights.

To narrow down the scope of positioning and gripper mechanisms the decision has been made to focus on only papers that describe compliant mechanisms, since it is commonly known that these have excellent characteristics in having a small design space, low particle contamination and high precision. Also, the mechanisms only focus on a millimeter or centimeter scale applications.

To narrow down the scope of particle contamination it is chosen to only look for sources of contamination in the context of an ideally designed, cleaned and installed mechanism in an ultra-high vacuum. The severity of different kinds of contamination is excluded from this literature review, whilst it is known that some sources can have a different kind of impact on the performance.

Literature regarding particle contamination is sparsely

found and is relatively old, which could lead to a publication bias. A significant part of the used papers in this literature review regarding particle contamination dates back to the year 1990-2000, a few are in the years 2010-2018. As particle contamination is a very important and large subject in the industry. It is expected that most of the research and gained knowledge is kept confidential in private companies and is not published publicly.

### 6.3 Recommendations

Many positioning stages are found in literature, some of them have unique characteristics like statically balanced actuation. These unique characteristics let them excel in certain use cases. In table 1 an overview is given of positioning stages and their most important characteristics. With this table it can be concluded that there are many two-degrees of freedom positioning stages that have low coupling between X- and Y-axis over a wide spread of different ranges and design spaces. Second, there are not many positioning stages described for applications where available design space is very limited. There are only a few compact positioning stages with a design area of around 30x30 mm.

Third, papers concerning gripper mechanisms do not focus on particle generation when gripping an object in an ultra-high vacuum. Therefore, it is unknown which design rules can be taken into account to minimize particle generations when two surfaces grasp each other.

Fourth, the Severeness of performance limiting risk for different kinds of molecules, contamination and generation mechanisms in the system is not researched in this literature review. This is partly due to the fact quantifying the amount or severeness of particle generation is very difficult and highly depends on the system that is observed. Second, in publicly available sources this information is not described. Also based on interviews with a contamination expert from VDL ETG [25], quantifying how large particle generation will be in a mathematical model based on the design parameters of the mechanism is very hard to do or not possible. An insight on the relationship between particle generation and design parameters of the mechanism is in the current way of working done by doing empirical research, this could be used to create a trade-off analysis between concepts.

At last, no mechanisms are found in the literature that can first grasp a millimeter or centimeter-scale object, position the object on a micron scale and can subsequently release it afterward. Especially no mechanism is found that had a special focus laid on small design space and low particle generation.

## 7. Conclusion

This literature review has provided an overview of various strategies for mechanisms that can position or grasp an object with micron-level accuracy and precision, and their respective relationship with particle generation. It was determined that the design of the kinematic coupling could influence three different particle generation methods. First, abrasive wear can be prevented by decreasing sliding distances and normal forces at the points where contact is made. Second, to eliminate adhesive wear the contacting surface area should be kept minimal. At last, some of the described strategies can influence the outgassing rate. All porous materials should be avoided, a form-closure gripper is hard to establish since it uses flexible porous materials like silicon or rubber. Statically balancing can reduce the required power and therefore lowering outgassing rates by reducing the required wire gauge.

## References

- [1] M. Kanetomo, H. Kashima, and T. Suzuki, "Wafer-transfer robot for use in ultrahigh vacuum," *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films*, vol. 15, p. 1385, 6 1998.
- [2] S. Carey, D. M. Knotter, E. Ooms, J. Boersma, E. Van Der Sar, R. Cop, W. Gerritzen, H. Van Zadelhoff, and H. Bouma, "Yield Impact of Backside Metal-Ion Contamination," *Solid State Phenomena*, vol. 187, pp. 287–290, 2012.
- [3] K. v. d. Broek, "[Internal] WOW - DESIGN FOR CONTAMINATION CONTROL," tech. rep., VDL Enabling Technologies Group, 2021.
- [4] R. W. Welker, R. Nagarajan, and C. E. Newberg, "Contamination and ESD Control in High-Technology Manufacturing," *Contamination and ESD Control in High-Technology Manufacturing*, pp. 1–498, 11 2005.
- [5] L. Zhao, X. Yu, P. Li, and Y. Qiao, "High-precision compliant mechanism for lens XY micro-adjustment," *Review of Scientific Instruments*, vol. 91, p. 035004, 3 2020.
- [6] T. Tuma, W. Haeberle, H. Rothuizen, J. Lygeros, A. Pantazi, and A. Sebastian, "A dual-stage nanopositioning approach to high-speed scanning probe microscopy," *Proceedings of the IEEE Conference on Decision and Control*, pp. 5079–5084, 2012.
- [7] A. Misra, "Challenges in Delivery of Therapeutic Genomics and Proteomics," *Challenges in Delivery of Therapeutic Genomics and Proteomics*, 2011.
- [8] A. H. Slocum and A. C. Weber, "Precision passive mechanical alignment of wafers," *Journal of Microelectromechanical Systems*, vol. 12, pp. 826–834, 12 2003.
- [9] A. C. Weber, "Precision Passive Alignment of Wafers," 2002.

- [10] A. C. Duenner and M. Cullinan, “Passive semiconductor wafer alignment mechanism to support inline atomic force microscope metrology.”
- [11] D. Petković, N. D. Pavlović, S. Shamshirband, and N. B. Anuar, “Development of a new type of passively adaptive compliant gripper,” *Industrial Robot*, vol. 40, no. 6, pp. 610–623, 2013.
- [12] T. F. Yao, A. Duenner, and M. Cullinan, “In-line dimensional metrology in nanomanufacturing systems enabled by a passive semiconductor wafer alignment mechanism,” *Journal of Micro and Nano-Manufacturing*, vol. 5, 3 2017.
- [13] P. L. Chang, I. T. Chi, N. D. K. Tran, and D. A. Wang, “Design and modeling of a compliant gripper with parallel movement of jaws,” *Mechanism and Machine Theory*, vol. 152, p. 103942, 10 2020.
- [14] J. Speich and M. Goldfarb, “A compliant-mechanism-based three degree-of-freedom manipulator for small-scale manipulation,” *Robotica*, vol. 18, no. 1, pp. 95–104, 2000.
- [15] P. J. Pluimers, N. Tolou, B. D. Jensen, L. L. Howell, and J. L. Herder, “A Compliant On/Off Connection Mechanism for Preloading Statically Balanced Compliant Mechanisms,” *Proceedings of the ASME Design Engineering Technical Conference*, vol. 4, pp. 373–377, 9 2013.
- [16] A. J. Lamers, J. A. Gallego Sánchez, and J. L. Herder, “Design of a statically balanced fully compliant grasper,” *Mechanism and Machine Theory*, vol. 92, pp. 230–239, 10 2015.
- [17] J. H. Kim, C. Asbach, S.-J. Yook, H. Fissan, K. J. Orvek, A. Ramamoorthy, P.-Y. Yan, and D. Y. H. Pui, “Protection schemes for critical surface in vacuum environments,” *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films*, vol. 23, p. 1319, 7 2005.
- [18] J. Zhao, “Thermodynamics and particle formation during vacuum pump-down,” 1990.
- [19] D. Chen and S. Hackwood, “Vacuum particle generation and the nucleation phenomena during pumpdown,” *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films*, vol. 8, p. 933, 6 1998.
- [20] Y. Ye, “Ph.D Thesis,” tech. rep., University of minnesota, 1992.
- [21] L. Lilje, “Controlling Particulates and Dust in Vacuum Systems,” *Glumslöv*, 2017.
- [22] J. L. Dorier and N. Hilleret, “Description of a dust particle detection system and measurements of particulate contamination from shock, gate valve, and ion pump under ultrahigh vacuum conditions,” *Review of Scientific Instruments*, vol. 69, no. 11, pp. 3818–3827, 1998.
- [23] K. A. Floyd, A. R. Eberly, and M. Hadjifrangiskou, “Adhesion of bacteria to surfaces and biofilm formation on medical devices,” *Biofilms and Implantable Medical Devices: Infection and Control*, pp. 47–95, 2017.
- [24] V. L. Popov, *Contact Mechanics and Friction*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017.
- [25] K. van den Broek, “Interview contamination expert VDL ETG,” 5 2022.
- [26] P. Řepa and M. Rott, “Outgassing of metals stimulated by friction,” *Vacuum*, vol. 48, pp. 775–778, 9 1997.
- [27] R. A. Nevshupa, J. L. De Segovia, and E. A. Deulin, “Outgassing of stainless steel during sliding friction in ultra-high vacuum,” *Vacuum*, vol. 53, pp. 295–298, 5 1999.
- [28] K. Jousten, “THERMAL OUTGASSING,” 1961.
- [29] J. M. Lafferty, *Foundations of Vacuum Science and Technology*. Wiley, 1998.
- [30] K. A. Reinhardt and W. Kern, *Handbook of Silicon Wafer Cleaning Technology*. Elsevier Inc., 2018.
- [31] L. Li and Z. J. Chew, “Microactuators: Design and Technology,” *Smart Sensors and MEMS: Intelligent Sensing Devices and Microsystems for Industrial Applications: Second Edition*, pp. 313–354, 2 2018.
- [32] L. L. Howell, “Compliant mechanisms,” p. 459, 2001.
- [33] J. Andres Gallego Sánchez, “(PDF) Statically Balanced Compliant Mechanisms: Theory and Synthesis..”
- [34] G. Radaelli, J. A. Gallego, and J. L. Herder, “An energy approach to static balancing of systems with torsion stiffness,” *Journal of Mechanical Design, Transactions of the ASME*, vol. 133, no. 9, 2011.
- [35] B. M. Wisse, W. D. Van Dorsser, R. Barents, and J. L. Herder, “Energy-free adjustment of gravity equilibrators using the virtual spring concept,” *2007 IEEE 10th International Conference on Rehabilitation Robotics, ICORR’07*, pp. 742–750, 2007.
- [36] D. A. Streit and E. Shin, “Equilibrators for Planar Linkages,” *Journal of Mechanical Design*, vol. 115, pp. 604–611, 9 1993.
- [37] K. Hoetmer, G. Woo, C. Kim, and J. Herder, “Negative stiffness building blocks for statically balanced compliant mechanisms: Design and testing,” *Journal of Mechanisms and Robotics*, vol. 2, 9 2010.
- [38] P. Gräser, S. Linß, F. Harfensteller, M. Torres, L. Zentner, and R. Theska, “High-precision and large-stroke XY micropositioning stage based on serially arranged compliant mechanisms with flexure hinges,” *Precision Engineering*, vol. 72, pp. 469–479, 11 2021.
- [39] “Compact design of a planar compliant XY precision positioning stage — TU Delft Repositories.”
- [40] Y. Li and Q. Xu, “A novel design and analysis of a 2-DOF compliant parallel micromanipulator for nanomanipulation,” *IEEE Transactions on Automation Science and Engineering*, vol. 3, no. 3, pp. 247–254, 2006.

- [41] Y. Shimizu, Y. Peng, J. Kaneko, T. Azuma, S. Ito, W. Gao, and T. F. Lu, “Design and construction of the motion mechanism of an XY micro-stage for precision positioning,” *Sensors and Actuators A: Physical*, vol. 201, pp. 395–406, 10 2013.
- [42] S. Awtar and J. E. Ustick, “Design and Experimental Characterization of a Large Range XYZ Parallel Kinematic Flexure Mechanism,”
- [43] P. Wang and Q. Xu, “Design of a compact compliant constant-force XY precision positioning stage,” *MESA 2016 - 12th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications - Conference Proceedings*, 10 2016.
- [44] M. Olfatnia, L. Cui, P. Chopra, and S. Awtar, “Large range dual-axis micro-stage driven by electrostatic comb-drive actuators,” *Journal of Micromechanics and Microengineering*, vol. 23, p. 105008, 9 2013.
- [45] Y. Tian, C. Liu, X. Liu, F. Wang, X. li, Y. Qin, D. Zhang, and B. Shirinzadeh, “Design, modelling and characterization of a 2-DOF precision positioning platform:” <http://dx.doi.org.tudelft.idm.oclc.org/10.1177/0142331214540692>, vol. 37, pp. 396–405, 7 2014.
- [46] “Scopus - Document details - Modeling and experimental testing of a composite bridge type amplifier based nano-positioner.”
- [47] S. C. Huang and T. P. Dao, “Design and computational optimization of a flexure-based XY positioning platform using FEA-based response surface methodology,” *International Journal of Precision Engineering and Manufacturing* 2016 17:8, vol. 17, pp. 1035–1048, 8 2016.
- [48] X. Zhang, Y. Zhang, and Q. Xu, “Design and control of a novel piezo-driven XY parallel nanopositioning stage,” *Microsystem Technologies*, vol. 23, pp. 1067–1080, 4 2017.
- [49] Q. Xu, “Design of a Constant-Force Flexure Micropositioning Stage With Long Stroke,” *Proceedings of the ASME Design Engineering Technical Conference*, vol. 5A-2015, 1 2016.
- [50] JPE Innovations, “CLS – Cryo Linear Scanner - JPE.”
- [51] Physik instruments, “U-723 PILine® XY Stage.”
- [52] Physik instruments, “P-612.2 XY Piezo Nanopositioning System.”
- [53] Physik instruments, “P-620.2 – P-629.2 PIHera XY Piezo Stage.”
- [54] A. T. Salton, Z. Chen, and M. Fu, “A class of preview control for dual-stage actuator systems,” *2009 IEEE International Conference on Control and Automation, ICCA 2009*, pp. 110–115, 2009.
- [55] L. Zhang, Z. Long, J. Cai, and J. Fang, “Design of a linear macro–micro actuation stage considering vibration isolation:” <http://dx.doi.org.tudelft.idm.oclc.org/10.1177/1687814015584541>, vol. 7, pp. 1–13, 5 2015.
- [56] J. Fang, Z. Long, L. Zhang, and L. Nian, “Driving process and control analysis in macro-micro dual stage,” *2013 IEEE International Conference on Information and Automation, ICIA 2013*, pp. 37–42, 2013.
- [57] P. Wang and Q. Xu, “Design of a flexure-based constant-force XY precision positioning stage,” *Mechanism and Machine Theory*, vol. 108, pp. 1–13, 2 2017.
- [58] H. Soemers, *Design Principles for precision mechanisms*, vol. 978-90-365-3103-0. 2010.
- [59] A. Slocum, “Kinematic couplings: A review of design principles and applications,” *International Journal of Machine Tools and Manufacture*, vol. 50, pp. 310–327, 4 2010.
- [60] J. L. Herder and F. P. A. van den Berg, “Statically Balanced Compliant Mechanisms (SBCM’S): An Example and Prospects,” in *Volume 7B: 26th Biennial Mechanisms and Robotics Conference*, pp. 853–859, American Society of Mechanical Engineers, 9 2000.
- [61] N. Tolou, V. A. Henneken, and J. L. Herder, “Statically Balanced Compliant Micro Mechanisms (SB-MEMS): Concepts and Simulation,” *Proceedings of the ASME Design Engineering Technical Conference*, vol. 2, pp. 447–454, 3 2011.
- [62] G. Chen and S. Zhang, “Fully-compliant statically-balanced mechanisms without prestressing assembly: Concepts and case studies,” *Mechanical Sciences*, vol. 2, no. 2, pp. 169–174, 2011.
- [63] N. Tolou, P. Pluimers, B. D. Jensen, S. Magleby, L. Howell, and J. L. Herder, “Constant Force Micro Mechanism out of Carbon Nanotube Forest,” 2011.
- [64] T. Chu Duc, G. K. Lau, and P. M. Sarro, “Polymeric thermal microactuator with embedded silicon skeleton: Part II- Fabrication, characterization, and application for 2-DOF microgripper,” *Journal of Microelectromechanical Systems*, vol. 17, no. 4, pp. 823–831, 2008.
- [65] A. R. Warmuth, S. R. Pearson, P. H. Shipway, and W. Sun, “The effect of contact geometry on fretting wear rates and mechanisms for a high strengthsteel,” *Wear*, vol. 301, pp. 491–500, 4 2013.
- [66] R. Merhej and S. Fouvry, “Contact size effect on fretting wear behaviour: Application to an AISI 52100/AISI 52100 interface,” *Lubrication Science*, vol. 21, no. 3, pp. 83–102, 2009.