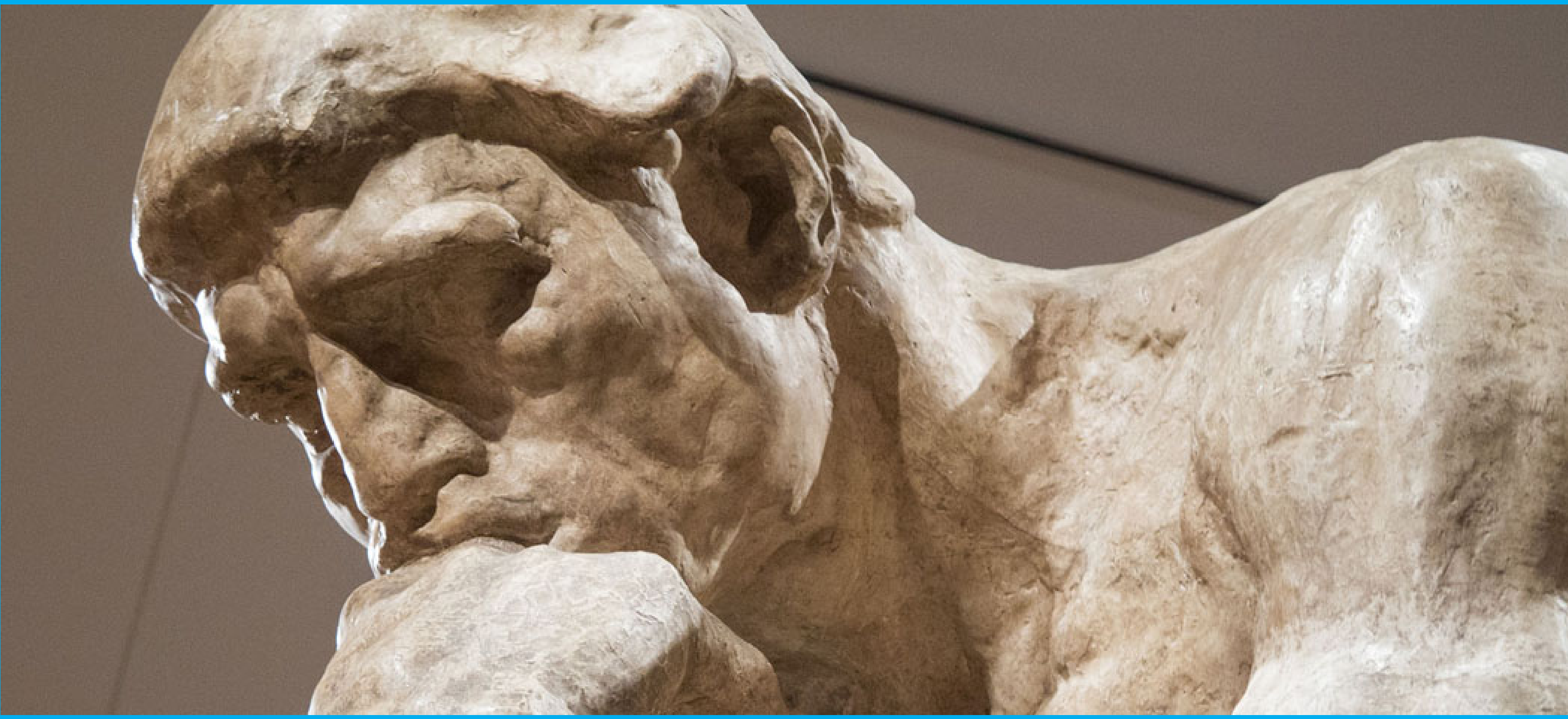
Cable-Driven Parallel Robot Design for Scanning 3D Art Objects

H. M. Waal

24-09-2021



TU Delft

Cable-Driven Parallel Robot Design for Scanning 3D Art Objects

by

H. M. Waal

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday September 24, 2021 at 02:00 PM.

Student number:	4321782	
Project duration:	May 22, 2020 – September 24, 2020	
Thesis committee:	Prof. dr. ir. J. L. Herder,	TU Delft
	Dr.-ing. P. Tempel,	TU Delft, supervisor
	Dr. ir. V. van der Wijk,	TU Delft, supervisor
	Dr. ir. A. Sakes,	TU Delft

Preface

During this thesis I researched the application of Cable-Driven Parallel Robots to the scanning of 3D art objects. A literature review on optimal design of Cable-Driven Parallel Robots and a design study were conducted to find the most suitable robots. At the beginning of this report an introduction is found, then the literature review and design paper are presented and at the end a conclusion, discussion and appendices are included. The idea for this thesis topic came from Dr.-Ing. Philipp Tempel, who investigated the scanning of 2D artwork at the TU Delft.

Conducting this thesis has taught me many lessons about setting up a research, conveying ideas and motivating myself. Moreover, working in python and writing the reports significantly increased my practical skills. With this report, I aspire to obtain the master degree in mechanical engineering at the TU Delft. This thesis was conducted at the High-Tech Engineering department, while my master track was BioMechanical Design department.

Without the guidance of Philipp Tempel and Volkert van der Wijk, this thesis would not have been possible. Their academic experience was of great benefit to my work. Most of all I appreciated their positivity, patience and the way they put things into perspective, this was inspiring and increased my motivation. Furthermore, I am very grateful for the meetings with Philipp Tempel, even though he switched universities. Additionally, I would like to thank Just Herder and Aimée Sakes for joining my graduation committee, assessing my work and giving feedback. Finally, the support of my family, girlfriend and friends throughout this process was invaluable to me.

H. M. Waal
Delft, September 2021

Picture on front page from [56]

Contents

1
Introduction

With the technology of the internet, 3D printing and holograms, 3D art such as statues can now be enjoyed in many ways without visiting their museums. A detailed scan of statues can, however, serve more purposes. First of all, curators can use scans to learn more about previous restorations and plan upcoming ones [51]. To art historians, the surface of a statue reveals many secrets about the techniques and tools of the sculptor or the origin of the material. Furthermore, the surface of the statues reveals the weather conditions that the statue has faced. Finally, researchers and students from all over the world can study the design from every angle and under different lighting, or use it for new creations [39]. The David of Michelangelo, for example, has been created with the purpose of viewing it from below. Viewing it from different angles can offer new insights on composition.

In order to accurately represent the statues, they have to be scanned in great detail. Different scanning solutions have been presented in literature, while some challenges have been faced. The first option for scanning is the use of a scaffold to manually place the camera at every position around an object, such as in Figure 1.3. Yet, this method is time-consuming, sensitive to artwork collisions and prone to vibrations due to walking on the scaffold [11, 66]. Gantries have been used to automate the scanning in [16, 39], but introduce cumulative positioning errors due to their serial nature [39, 64]. An example of a gantry for scanning a statue is given in Figure 1.1. An industrial serial robot arm, depicted in Figure 1.2, can be highly accurate, but the robot has limited scaling possibilities. Furthermore, the arms are expensive and heavy, which makes repositioning and moving them to exhibition halls more difficult. Rotating the 3D objects themselves is a possibility too. Using a rotating disc is, however, not accurate enough [11] and relocating delicate artwork is cumbersome. Handheld scanners have been used as well, but require more manual labor and introduce artifacts in the images which complicate post-processing [39]. Scanning of small statues has also been done with cameras on tripods. In this case, staff readjust the camera similar to the scanning with scaffolds. To minimize time spent in museums and labour costs, most of the work is shifted to post-processing, which is done in an office or is automated [7]. Nonetheless, scanning of the more than 5 m tall statue of Michelangelo's David with a gantry required 33 nights and involved 22 people [39]. A quicker, scalable and automated scanning method could simplify the scanning, reduce the costs and allow for the studying of more statues.

Recently, a Cable-Driven Parallel Robot (CDPR) for the scanning of 2D artwork was introduced by Tempel [64]. CDPRs solely rely on cables to move an end-effector, also called platform, and connect it to the frame. The cables are attached to the platform at distal endpoints and connected to the frame at a proximal endpoint. At the frame, the cables are guided with pulleys to a winch to shorten or extend the cables. The general CDPR lay-out is depicted in Figure 2.5. The CDPR was able to fully cover the painting in Tempels work, but also pan and tilt the scanner with 35° to capture depth. Moreover, CDPRs were advantageous because they are more accurate than most serial robots and compared to conventional parallel robots, are lighter and have a larger workspace. They are also easy to disassemble, build, move and scale to different sizes, because of their modularity [64]. When a CDPR is applied to scanning 3D artwork, two new major challenges need to be faced. First of all, a statue needs to be scanned from all sides with as little repositioning as possible to minimize labor costs. For this purpose, an additional large rotation of the CDPRs platform around the 3D object is required. A second challenge emerges during this rotation: the 3D object in the middle of the robots workspace is obstructing the cables from moving freely. Hence, a CDPR that is able to move around a statue without collisions is needed.

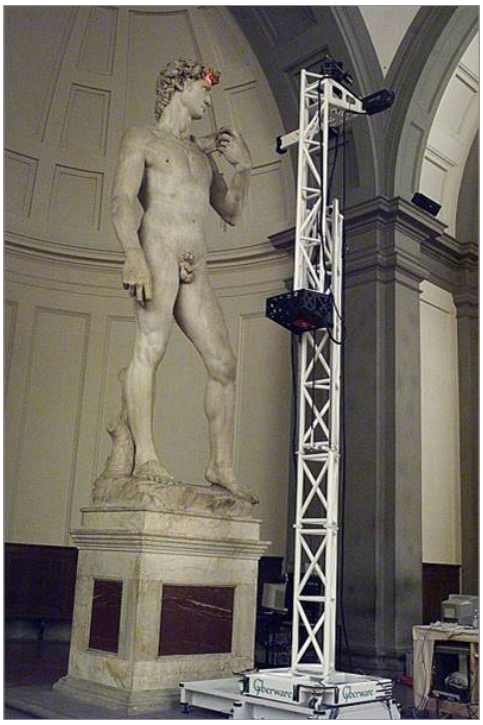


Figure 1.1: The gantry to scan the David of Michelangelo [39].

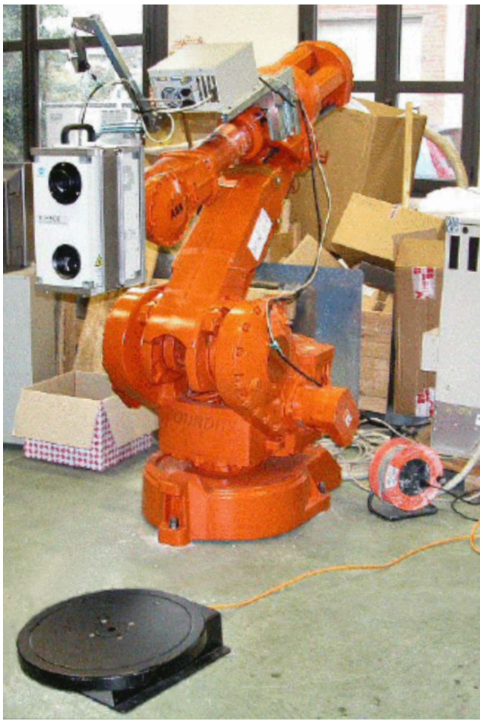


Figure 1.2: An industrial robot arm with rotating platform for the scanning of 3D art. [11].

Figure 1.3: Usage of a scaffold around a statue to position a scanner [30].

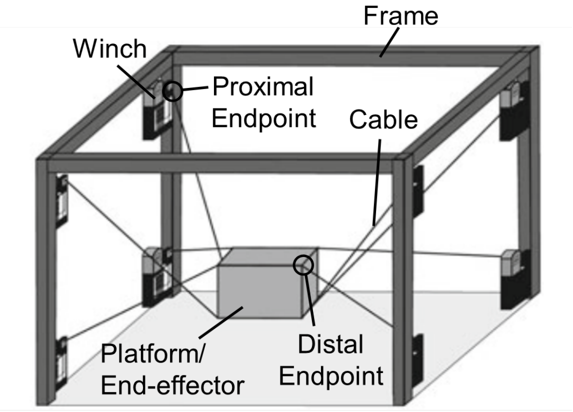


Figure 1.4: The general lay-out and terminology for a CDPR, picture taken from [52].

The advantages of CDPRs for scanning 3D artwork seem promising but their application has not been investigated yet. Since the design of CDPRs is strongly application dependent [52], the goal of this thesis was to provide a good geometrical CDPR design for the scanning of 3D objects and quantify its performance. First, however, a design method had to be found that is tailored for the challenges of scanning 3D art. A literature study was conducted on the state-of-the-art of optimal design of CDPR geometries and the applicability to 3D object scanning. Useful requirements, architectures, parameterization, performance indices and optimization algorithms were extracted. Furthermore, no research was found on non-reconfigurable, non-suspended CDPRs that were designed for rotating around 3D object. With the found methods and research gap, a design process was subsequently executed. First designs were created with a brainstorm. Then a workspace calculation model was built to evaluate the designs and an optimization algorithm was used to polish the results. In the end, suitable designs and their performance were presented.

The literature review is described in Chapter 2. The research paper in Chapter 3 then describes a design process, the found designs and the design performances. In Chapter 4, the discussion puts the outcomes into perspective and makes recommendations on the application of CDPRs to the scanning of statues. This thesis is summarized and conclusions are drawn at the end, in Chapter 5.

2

The State-of-the-Art of Geometry Design to Improve the Workspace of Cable-Driven Parallel Robots and their Application to Large Rotations and 3D Obstacles

Abstract

Because of low weight, modularity and reconfigurability, Cable-Driven Parallel Robots for the scanning of 2D artwork are currently researched at TU Delft. For 3D artwork such as statues, challenges concerning limited rotations of the end-effector and cable-interferences emerge when the end-effector must rotate around an axis with 180° as well as around a statue. To find a good design method to address these challenges, a literature review was conducted. The search focused on all publications regarding the search for the robot's geometrical parameters in an optimal design process to obtain required workspace characteristics. Both the methods and the results of the literature were evaluated to assess whether they contain information beneficial for 3D Object scanning. Subsequently, the content was summarized in four categories: requirements, performance indices, structural synthesis and dimension synthesis. Several conclusions could be drawn from this overview. First of all, most design processes found in literature start by choosing an architecture through intuition and then optimize the geometry for the performance indices. Furthermore, no design has been presented yet that meets the requirements for 3D scanning. Finally, the complement workspace and genetic algorithm present a suitable performance index and optimization algorithm respectively.

2.1. Introduction

To get a good understanding of the direction of this review, first some theoretical background on cable-driven robots will be given. Next the desired application of the Cable-Driven Robots in the CaRiSa project is elucidated. At the end of the introduction, the research design of the literature review is presented.

2.1.1. Cable-Driven Parallel Robots

Cable-Driven Parallel Robots or in this review also called CDPRs or cable robots are a special type of parallel robots since they use cables instead of rigid links to support and actuate an end-effector. An example of a CDPR is presented in figure 2.3 [31]. The cables are attached to the end-effector and are guided by pulleys to winches connected to actuators on the frame to make the cables longer or shorter. The frame is also called the base and the end-effector is called the manipulator or mobile platform as well in this review. The attachment points of the cables at the frame are called the proximal endpoints and the attachment points on the end-effector the distal endpoints.

The CDPRs have many advantages over normal parallel robots. Firstly Cable-Driven Parallel Robots have a large workspace to frame ratio. Furthermore, the low inertia of the cables enable large velocities and accelerations (up to 10G) and in general a very large workspace since the cables can be made very long [28]. Another advantage of the cables are their ease of transport. Additionally, a cable robot can easily change shape or be scaled up or down to other frames, since the cables can be lengthened or shortened to a new position [48]. The frame is not an essential part of the cable robot, any frame, such as an existing building or movable cranes, can be used to attach the actuators and winches to. CDPRs can carry a higher payload too, since the cables are resisting forces very efficiently and their price is relatively low because no specialized parts are needed [69]. The CDPRs end-effector is however more compliant than general parallel robots. Another significant challenge is the unilateral force in the cables. The cables can only exert a pulling force and in order to control all of the degrees of freedom, more cables are attached to the end-effector. The multiple cables moving around can lead to cables interfering with objects or themselves. The end-effector is furthermore not able to show large rotations because of cable interference and singularities.

A subtype of CDPRs is the suspended version, for which cables are only attached to proximal endpoints above the end-effector. This version relies solely on gravity for downward acceleration. The figures 2.1, 2.2 and 2.4 all show suspended CDPRs, while 2.3 is not-suspended. The way the CDPR is constructed, being not-suspended or suspended for example, is called the architecture, lay-out or structure of the CDPR.

Applications

The first cable-driven parallel robot used commercially is the SkyCam [14], shown in figure 2.1. This suspended CDPR is operating in stadiums and is hovering over fields for filming. An example of an application of CDPRs in astronomy is the Five hundred meter Aperture Telescope (FAST) [13], shown in figure 2.2, which is a radio telescope in China with a diameter of 500 m. The bowl of the telescope is embedded in a valley and a receiver is hovering above the bowl as end-effector. Using a CDPR to position the end-effector was necessary in this case because of the exceptionally large workspace and high forces. The suspended cable-driven robot CoGiRo is used for heavy payload handling and is studied by Gouttefarde [25] and shown in Figure 2.4. The suspended CDPR is practical in this case because it is able to move over objects in a large workspace. Another application of cable-driven parallel robots is rehabilitation, then the cable robot is used to either support the arms, legs or the entire human body [36, 37, 47]. Also a robot for the scanning of 3D objects has already been designed [23]. It is a six cable suspended CDPR with a 1D actuator on the end-effector to allow for more rotation around the x/y axis. This robot is mainly scanning objects from above.

2.1.2. CaRISA

At TU Delft, the CaRISA project is focusing on the scanning of artwork with CDPRs. CaRISA is an acronym for Cable Robot for Inspection and Scanning of Artwork. The project is a collaboration between Material Science Engineering and Mechanical Engineering. The scanning of artwork is an important task for multiple reasons. Firstly, it helps understand the techniques of painters and sculptors. Recently, a scan of the Nightwatch of the Dutch painter Rembrandt revealed that he painted with Arsenic, a poisonous compound normally not used for these type of paintings [55]. When more insight is given in their techniques, the history with regard to these famous men can be unravelled and new work could be linked to painters and falsification and fraud can be uncovered. When, additionally, it is more clear how art was created, it is easier to restore pieces of art. Moreover, artwork can be stored as a digital version and be made accessible to anyone that is not able to visit the physical artwork.

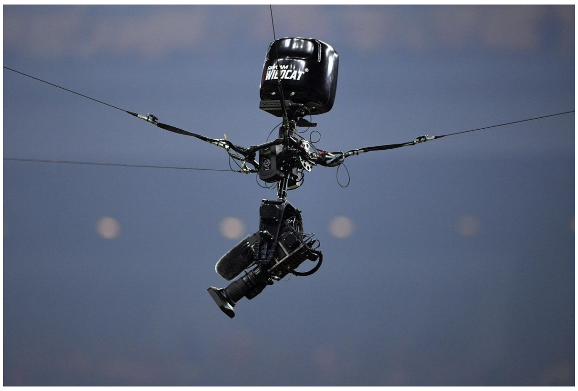


Figure 2.1: The SkyCam [14]

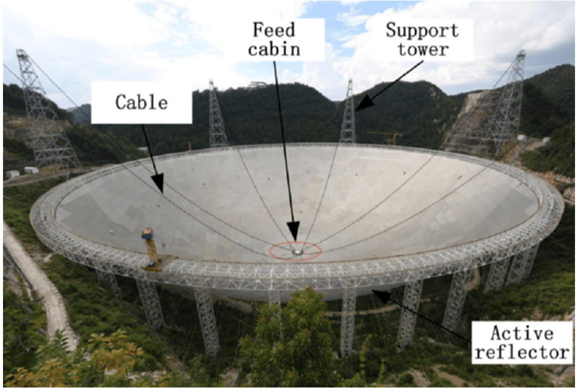


Figure 2.2: The FAST telescope [13]

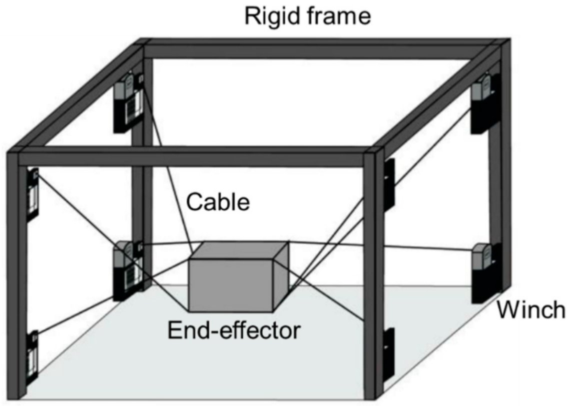


Figure 2.3: The general lay-out of a Cable-Driven Parallel Robot from [31]

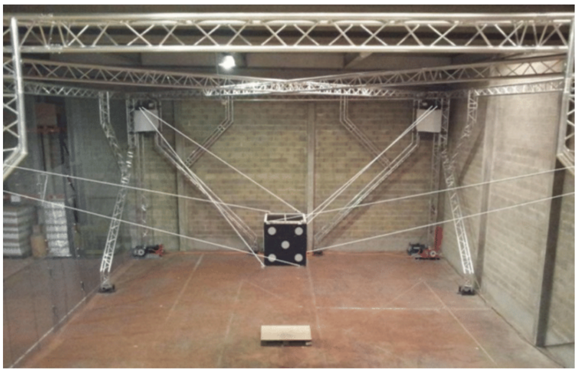


Figure 2.4: The CoGiRo suspended cable robot [17]

The scanning of artwork is nowadays still an elaborate and time-consuming job. For the scanning of the Nightwatch, an entire lifting platform with a 3 axis translational robot has been built in front of the painting. In many exhibition halls it is not possible to install such large lifting platform and robotframe, so a light compact version was searched for. This is where cable robots come into play, they are easy to assemble and disassemble, lightweight and can be adapted to different sizes of artwork. When using CDPRs, extra panning and tilting of the camera is possible as well, when the CDPR is designed for this task. The original scope of CaRISA is 2D artworks, but this literature review will explore the possibilities of using the CDPRs to scan 3D artwork as well. 3D artwork poses new challenges for the CDPR, such as larger rotations and the possibility of collision of the cables with the artwork.

Application Requirements
The scanning of 3D artwork introduces the following requirements;

- A statue approximated by a cylinder of 2m height and 0.5m diameter on a 0.5m high pedestal is subject to the scanning as a proof of concept.
- The statue should be scanned 180° around the statue. In this way, the statue is fully scanned by changing the setup of the cable robot only one time.
- The camera should be able to pan and tilt in the range of ±35° to capture relief and play of light on the statue. Only around the focal axis no rotation is required.
- The exhibition room is 4 meters high and has a width and length of eight meters. This poses a constraint on the size of the robot.
- The 3D artwork is considered as a very delicate object and can't be moved or touched.

- The platform has a weight of 5 kg.
- The camera should be able to move closer and further away in a range of [5...15] cm away from the sculpture.
- The size of one image should be (5x5) cm.
- The scan speed should be around 12 seconds per frame. Due to the combination of this scanning speed and the size of the images, only quasi-static linear velocity is required. Thus, the focus of design should initially be on static equilibria.
- Cable forces should be within the range of [10...150] N.
- The robot design should be modular and available to be used for scanning of artwork of different sizes in different rooms.
- Ideally the platform should only carry the camera and no extra actuators to keep the weight of the end-effector as low as possible and to refrain from an extra cable to feed the actuator.

2.1.3. Research Questions

The posed requirements of the CaRISA project ask for a specific design and no indications have been found that an existing cable robot already meets the requirements. For building a new CDPR it is assumed that actuators and winches are already determined since they will be used for different artwork in different situations. The focus on the design will for this reason be on the size and shape of the frame and manipulator and the cable configuration, parameters that are easily changed for different situations, since pulleys can be repositioned on a frame and cable configurations can be altered by re-attaching cables. At the beginning of the literature review it became clear that the geometrical design parameters of a CDPR are synthesized for a given task with a process called optimal design. Furthermore the most significant challenges compared to scanning of 2D artwork will be the extra rotation of the end-effector and 3d object interference. This knowledge led to the following research question: *What is the state-of-the-art on workspace improvement of Cable-Driven Parallel Robots through the optimal design of geometrical parameters and which methods can be used for obtaining a 3D workspace with large rotations and a 3D obstacle?*

2.1.4. The Literature Review

The review conducted in this literature review is a narrative review according to [57]. The design goals vary highly in found literature and the designs processes are difficult to compare. This review was conducted to discover trends in design methods for workspace characteristics and to find the most suitable methods for a CDPR scanning 3D art. Furthermore, this review's goal is to address the research gaps within found literature that need to be filled in order to achieve a workable design.

No review has been found that addresses every step of the design process of cable-driven robots. Furthermore it became clear that a selection was made by the authors for papers they deemed relevant. The review of [28] for example, describes every step in the optimal design process, but only a selection of performance indices and optimization algorithms. Furthermore no overview of the properties of an architecture and their influence on performance of the Cable-Driven Parallel Robot is given. In [22] optimal design is only briefly mentioned with the remark that there is still little research concerning synthesizing geometrical CDPR designs. The review of [54] is only mentioning some examples of aspects of optimal design to give a general idea of the process, but the successive steps are not categorized, other works are not compared with each other, little examples are given and no arguments for making choices in the design process are provided. In [52], quite some literature on optimal designs are presented, but not categorized in a systematic way following the steps of the design process. Furthermore, there are gaps in the categories of optimization algorithms and performance indices. To conclude, a literature review is necessary for addressing variation in detail of the steps in the design synthesis with optimal design. The possible choices that can be made in every design step should be elaborated with pros and cons. Furthermore extra information should be provided on design choices necessary for the design of CDPR scanner of 3D artwork.

2.2. Search Method

To find the right directions for the literature research, other reviews were consulted first. The reviews of Gosselin [22], Hong [28], Qian [54] and the book of Pott [52] offered insight into the general design process and the steps that should be taken to acquire a geometrical design of a CDPR. Using the background knowledge of the reviews and book and the research question, the following search query was constructed:

((((Cable OR Wire OR Tendon OR Rope OR String) NEAR/2 (Driven OR Based OR Suspended OR Actuated)) NEAR/15 (Robot OR Manipulator* OR Mechanism*) NEAR/15 Parallel)* ***AND*** *(((Geometr* OR Optim* OR Topological) AND (Design OR Dimension* OR Synthesis)) OR "dimension* synthesis" OR "parameter synthesis")* ***AND*** *(Workspace OR Singularit* OR Interfer* OR Avoid* OR Collision*))*

The individual parts of the search query are separated with a bold AND. Their purposes are:

- *((((Cable OR Wire OR Tendon OR Rope OR String) NEAR/2 (Driven OR Based OR Suspended OR Actuated)) NEAR/15 (Robot* OR Manipulator* OR Mechanism*) NEAR/15 Parallel)* Is the part of the search query for filtering out only CDPR related research papers. The NEAR operators are used instead of the AND operators to account for different ways of describing the robot. The synonyms for Cable, Driven and Robot are taken from [52].

- *(((Geometr* OR Optim* OR Topological) AND (Design OR Dimension* OR Synthesis)) OR "dimension* synthesis" OR "parameter synthesis")* is the part of the search query that makes sure all the research paper focus on the optimal design process of CDPRs.

- *(Workspace OR Singularit* OR Interfer* OR Avoid* OR Collision*))* is used to only get search results focussed on the workspace or object avoidance.

A search was conducted using Web of Science to find published papers. The research query yielded 169 results with a substantial part that was not considered relevant. By reading title and abstract and using inclusion/exclusion criteria the selection was narrowed down to 67 papers. The inclusion/exclusion criteria and their reason of usage were:

- A full system design process with optimal design has to be conducted in the papers. This is to make sure the methods or designs in the papers are already validated and have proven their worth. It would be too cumbersome to incorporate for example every paper on a new performance index without the performance index ever showing its practical advantage in a design process. Also papers on only workspace analysis, trajectory planning or control were left out for this reason.

- The objective of the design process or the most important constraint need to be related to the workspace or geometrical parameters and not for example the actuator energy consumption.

- The design parameters have to be geometrical parameters or the cable configuration and not for example the gear ratio.

- Only CDPRs were considered that had no rigid links or springs attached to the end-effector, because those designs compromise the intended ease of transport and applicability to different situations.

- Research using interval analysis was left out of the selection. Interval Analysis is considered to be a difficult topic and the limited time of the project makes interval analysis infeasible for applying.

The remaining 67 papers were read with more care and another 32 did not fully meet the inclusion/exclusion criteria or were copies of papers. Two extra papers that met the inclusion/exclusion criteria were found in the review papers mentioned earlier. The total 37 of papers were read and both categorized and summarized at the same time by keeping track of the following properties on the architecture:

- The motion pattern, so in which degrees of freedom the end-effector can move. Examples are 1R2T, 3T or 3R3T

- The suspendedness, a robot is either suspended or not-suspended

- Restrainedness, depending on the amount of cables with respect to the motion pattern

- The amount of cables

- The dimension and shape of the manipulator and frame

The following properties on optimal design were noted down:

- The geometry subjected to optimization, so for example the frame, the end-effector, both the frame and end-effector or the cable arrangement.
- The parameters describing the parameterized geometry
- The performance indices used
- The constraints used during optimization
- Whether the objective function is single or multi-dimensional
- The optimization algorithm
- The sampling of the workspace properties

These categories were written down in an excel file to offer an intuitive summary of the results and can be used to find gaps and draw conclusions. The names and authors of all 37 read papers are listed in the Appendix. Not all papers will be used during the literature review, only the ones that are relevant. The only missing categories are the algorithms for cable tension and singularity calculation. These algorithms, however, do not change the geometrical design, they can only make the computation of the workspace more efficient.

2.3. Results

In this chapter an overview of the design process of cable-driven parallel robots is integrated with an overview and categorization of conducted optimal design studies on cable-driven parallel robots. This part of the design process is further analysed since it will provide an insight into the most elaborate part of the design process and a rough guideline for setting up a research methodology for a new design process. The CaRISA applications are evaluated as well in the results for guidance during the rest of the literature review. The CaRISA requirements are however not a product of the literature review.

2.3.1. Literature on the Design Process

According to Hong [28], a successful design can prevent the CDPR from having a very large frame, high tension in the cables and a small workspace. The research on CDPR is, however, highly limited to theory, so 3D applications that either focus on large workspaces or high precision still need a multi-step design and optimization procedure [28]. The total design process is not a systematic process yet, because the research field has not been covered sufficiently. The designs are strongly adapted to their purpose, because of the design freedom due to the modularity of the system [52]. This literature review will focus on the first steps of design: defining the requirements and setting up a system design. With these steps a general geometrical configuration of the cables, frame and platform of the CDPR will be determined. Subsequential steps such as hardware design lie outside the scope of this literature review.

The steps that can be distinguished in [52] are:

Defining application requirements The application requirements are constraining the design process. The requirements can be straightforwardly deduced from the desired product description, but some also have to be derived from other application requirements. Straightforward requirements can be the workspace, the lifting capacity and required speed. The workspace however, does not only have to have a certain size, it should also for example be free of any positions in which cables touch each other and entangle. Another less straightforward requirements could be the stiffness of the platform position for example.

Defining Performance Indices The selection of performance indices has to be done before the dimension synthesis, but later on in this chapter they will be immediately discussed after the application requirements section, since the former can be derived from the latter. The performance indices are a way of quantifying the quality of a robot design. This enables an evaluation, optimization and comparison of designs.

Structural Synthesis Structural synthesis is the first step of the two synthesis steps. During this step the general properties of the CDPR are chosen. This step is also called the architecture/archetype/system/layout design/synthesis. The qualities from [52] that are determined during this step are:

- The degrees of freedom and the desired motion
- Under-, completely or over-constrained robot
- A suspended or not-suspended robot
- The number of cables m
- The cable configuration, so in which way are the proximal and distal enpoints connected by cables.
- Frame shape, size and endpoints. The size of the frame should not be too large because this wastes material and space, but the size of the frame is directly related to the workspace. The shape of the frame is determining where the possible proximal endpoints could be. In most literature, either a box or a cylindrical frame is used.[52]

- Platform shape, size and endpoints The size and shape of the platform constrain the possible distal endpoints. A larger platform could interfere more easily with cables but on the other hand, cables could exert a larger wrench on the platform. The wrench is the term that describes the external load on the platform, which consists of force vector There are multiple ways of choosing the architecture. Always some decisions have to be made based on the engineers intuition and knowledge of previous CDPRs. In some publications multiple architectures are suggested and already compared quantitatively with the help of performance indices to find the most suitable one. This can be done with a few educated guesses, but the space of possible architectures can also be more exhaustively searched.

Class parameterization Class parameterization is the process of choosing the design variables that can be altered to improve the performance of the robot. Either parameters such as coordinates are chosen of different parameters are coupled to decrease the amount of design variables. Having less design variables speeds up the process of finding the right design.

Dimension Synthesis During dimension synthesis or parameter synthesis the parameters of the architecture are tweaked to have the best performance as possible. During this step an architecture can be optimized by minimizing/maximizing an objective function containing one or more performance indices through changing the design variables. This is then called optimal design

2.3.2. Application Requirements

All the requirements related to the particular application of Cable-Driven Parallel Robots in this research, can be related to the workspace. In this section an explanation of the workspace, literature review results for the different workspace definitions, their criteria and ways to compute them are presented. The set-up of this section has been taken from [52]. At the end of this section, the exact requirements for the CDPR for scanning of 3D art objects are listed.

Workspace Criteria

Whether a pose of the platform belongs to the workspace, depends on the fulfillment of several criteria. The first and foremost criteria are the wrench-closure and wrench-feasible workspace conditions. These and other important criteria for workspaces in the scope of this research are summarized in this subsection.

Wrench-Closure Workspace The wrench-closure workspace is defined by all possible static poses of the end-effector for which there is a positive tension in the cables and no singularity. Before the force equilibrium is explained, the general symbols for Cable-Driven Parallel kinematics are introduced with the help of figure 2.5.

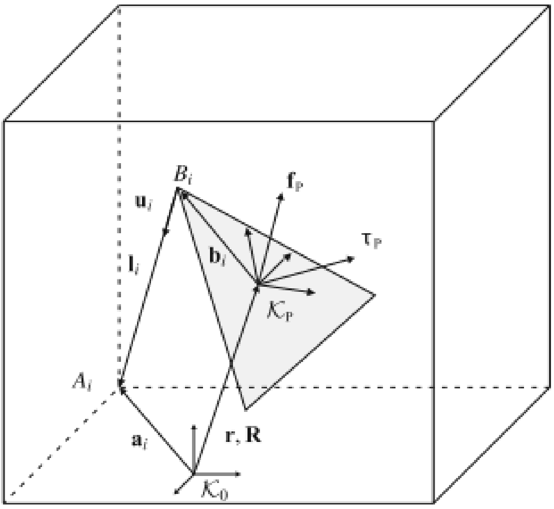


Figure 2.5: The usual symbols used for describing CDPR kinematics [52]

As seen in figure 2.5 there are two coordinate frames $\mathscr{K}_0$ and $\mathscr{K}_p$ which are the general coordinate frame and the end-effector coordinate frame. The displacement vector **r** and rotation matrix **R** express the position and orientation of $\mathscr{K}_p$ with respect to $\mathscr{K}_0$. The symbol A_i denotes the attachment point of the ith cable at the base which has location $\mathbf{a_i}$ with respect to $\mathscr{K}_0$ and B_i denotes the attachment point of the ith cable at the end-effector which has location $\mathbf{b_i}$ with respect to $\mathscr{K}_p$. The vector $\mathbf{l_i}$ is the vector between A_i and B_i that represents the cable. On the end-effector a certain wrench $\mathbf{w_p}$ is applied. The unit vector $\mathbf{u_i}$ of the cables is defined by [52] as

$$\mathbf{u}_i = \frac{\mathbf{l}_i}{\|\mathbf{l}_i\|_2}. \tag{2.1}$$

For a force equilibrium the cables should be able to resist every wrench on the platform. According to [52] this equillibrium can be expressed with [52]:

$$\mathbf{A}^{\mathrm{T}}(\mathbf{r},\mathbf{R})\mathbf{f}+\mathbf{w_p}=\mathbf{0}. \tag{2.2}$$

The vector **f** contains the cable forces and has dimension $(m, 1)$, with m for the amount of cables. The degrees of freedom is depicted with n. The matrix $\mathbf{A}^{\mathrm{T}} \in \mathbb{R}^{nm}$ is called the structure matrix or Jacobian and maps the vector of cable forces on the end-effector **f** to a vector expressing the wrench of the cables exerted to the platform $\mathbf{w_p} = [\mathbf{f_p}, \tau_\mathbf{p}]^{\mathrm{T}}$ [67]. Matrix $\mathbf{A}^{\mathrm{T}}$ looks as follows [52]:

$$\begin{bmatrix} \mathbf{u}_1 & \dots & \mathbf{u}_m \\ \mathbf{b}_1 \times \mathbf{u}_1 & \dots & \mathbf{b}_m \times \mathbf{u}_m \end{bmatrix}. \tag{2.3}$$

The forces in the cables can be computed since the position and applied wrench on the end-effector are known for every pose. When there are more cables than degrees of freedom however, the structure matrix is underconstrained and infinitely many solutions exist for the system of equation of:

$$\mathbf{A}^{\mathrm{T}}(\mathbf{r},\mathbf{R})\mathbf{f}=-\mathbf{w_p}. \tag{2.4}$$

The solution space $\mathscr{S}$ is described by [52] with

$$\mathscr{S}=\left\{\mathbf{f}=-\mathbf{A}^{+\mathrm{T}}\mathbf{w}_{\mathrm{p}}+\mathbf{H}\lambda \quad | \quad \lambda\in\mathbb{R}^{r}\right\}. \tag{2.5}$$

The expression $\mathbf{A}^{+\mathrm{T}}=\mathbf{A}\left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}$ maps the external wrench onto the solution set and **H** is the spanning basis of the Nullspace of the Jacobian. Cable force distributions within this space do not add a nett wrench to the platform and so $\mathbf{H}\lambda$ is a cable prestress that can be freely added until the cable force vector **f** reaches the desired values [5].

To solve for a desired tension distribution, solving equation 2.4 is described by [52] as an optimization problem in the following way:

$$\text{minimize}\, g(\mathbf{f})=\|\mathbf{f}-\mathbf{f}_{\mathrm{ref}}\|_p=\sqrt[p]{\sum_{i=1}^{m}\left(f_i-f_{\mathrm{ref},i}\right)^p} \tag{2.6}$$

$$\text{subject to} \quad f_i\geq 0 \tag{2.7}$$

$$\text{linear constraints} \quad \mathbf{w}_{\mathrm{p},i}=-\sum_{j=1}^{m}A^{\mathrm{T}}_{j,i}f_j. \tag{2.8}$$

In this equation $\mathbf{f}_{\mathrm{ref}}$ is the desired tension in the cables and p is the norm of the solution. This optimization problem can be solved with many different optimization algorithms.

Wrench-Feasible Workspace The wrench-feasible workspace (WFW) is a subspace of the wrench-closure workspace. An added constraint is narrowing the workspace down. For the wrench-feasible workspace the cable tension cannot be less than or exceed a certain value. So for acquiring the WFW in [52] equation 2.7 is extended to:

$$\text{subject to} \quad f_{\mathrm{min}}\leq f_i\leq f_{\mathrm{max}}. \tag{2.9}$$

A minimum value in the cables prevents sagging. Sagging changes the force exerted on the platform in a way that is not modelled and makes the robot control harder [52]. Furthermore having a certain minimum force in all the cables assures stiffness of the end-effector. A maximum value of cable tension is related to a maximum strength of either the cables or the winches and actuators.

Singularities According to [52] there are three types of singularities which are shown in Figure 2.6. The serial singularity only occurs when a distal and proximal endpoint coincide and the architectural singularity just occurs in case of a bad design. Only the parallel singularity is likely to happen with CDPRs. It is detected when the rank of the Jacobian is less than the degrees of freedom of the robot for a certain pose **y**. This is expressed as:

$$\mathrm{rank}\mathbf{A}^{\mathrm{T}}(\mathbf{y})<n. \tag{2.10}$$

When $\mathbf{A}^{\mathrm{T}}$ is rectangular, no determinant can be calculated but a rank deficiency can still be proven when the smallest singular value of the matrix is very small [52]. The singular value is calculated with singular value decomposition. When the workspace is tested by using a defined grid of poses, singularity could still happen for poses in between the grid points.

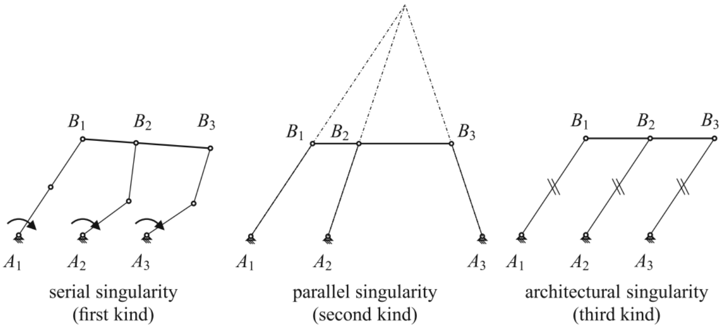


Figure 2.6: The three types of possible singularities according to [52]

Cable-Cable Interference In certain positions of the platform, the cables can collide with each other. The collision makes control more difficult and limits further movement, which makes the collisions unwanted. There are different tests for cable-cable interference detection and they have been summarized already by Andreas Pott [52]. In the end, a way to calculate the total orientation workspace without any cable-cable interference is what matters, so this workspace can then be optimized. A method for determining cable interference is found in [25] and in [47]. The methods rely on modeling the cables as stiff line segments, then computing $\mathbf{n_r}$, the vector of the common perpendicular of two cables and then checking whether the length of $\mathbf{n_r}$ surpasses the diameter ϵ of the cables or not [47]. This method is visualized in figure 2.7.

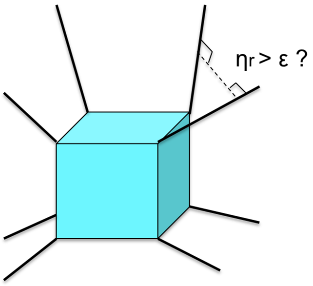


Figure 2.7: The check for cable interference, does the closest distance between the cables surpass the diameter?

Cable-Platform Interference Depending on the architecture Cable-Platform interference can be a problem and should be calculated for every pose just like the cable-cable interferences. This is possible by first computing the n_r, the vector of the common perpendicular of a cable and the vertices of the platform. Then it is checked whether the length of n_r surpasses half of the diameter of the cable [47].

Interference with external object Since the statue in the design problem cannot be hit by either the platform or the cables, there should also be a way of calculating the workspace in which these kinds of interferences are not allowed. Again the object interferences can be calculated in the same way as the cable-platform interferences, but then by taking the vertices of the object. Not many papers found are focussing on external object - platform/cable interference. The only examples found are the scanning of building facades in [29] and the sandblasting of a jacket such as in [19]. Only in the latter paper there is an external object in the middle of the workspace. The problem of moving around the jacket for sandblasting is tackled by Gouttefarde by making the attachment points on the base reconfigurable.

Workspace Computation

The ways the workspace can be computed and presented are ordered into a few categories by [52]. The summary is found below.

The first category are the discretization methods. With these methods, single poses arranged in a grid are evaluated to check whether they belong to the workspace or not. From the work of Pott the following can be summarized about grids: the grid can be regular, random or adaptive. Adaptive grids change fineness depending on the region of the grid. Advantages of the grid-based method are: it is a intuitive task to set up a certain grid, no a priori knowledge about the size and shape of the workspace is required and the points belonging to the workspace can easily be used to estimate the size and shape of the workspace. The most important disadvantages are: the workspace boundary is not very accurate since it is a collection of nodes that lie with full certainty within the workspace, it is a relatively computation heavy method, also increasing the resolution has serious implications on the computation time, because of the coarseness of the grid, the method is insensitive to small iterative changes in design parameters and some poses within the found workspace can still be singular, but can be missed because of the coarseness of the grid [52].

The second category are analytical methods, but analytical formula's for a workspace require long computation times and are not suitable for repeated use during evaluation. They will not be further discussed in this review.

The third category are the geometrical methods. These designs use the properties and limits of the geometrical design to determine a feasible workspace. An example [52] brings up is to find the common volume of all the volumes in which distal end points can reside, when a minimal and maximal cable length is limiting the distance of the distal end point with respect to the complimentary proximal endpoint.

The final category presented by [52] are the continuous methods. These methods use mathematical tools that can evaluate finite volumes that contain an infinite amount of poses. In this way all poses within the workspace are evaluated and are certain to be feasible. With so called interval analysis a workspace can be computed in this continuous way [52]. It has however been decided that any interval methods will be left out of this review, because the methods of interval analysis are too complex to use them within the scope of this review.

A whole different approach is also discussed in [52]: instead of computing the properties of the full volume of the workspace, it is also possible to determine the boundary and then subsequently compute the surface area, volume and centre of mass of the workspace. These kind of methods are called numeric boundary methods. To determine such a boundary, or also called hull, a geometrical shape composed of triangles is placed in the suspected middle of the workspace. The more triangles are used, the more accurate results will be. In the next step the shape in the centre is expanded per triangle and in this way the triangles are mapped to the boundary of the workspace. It can be compared to a balloon that is blown up inside a box, and takes the shape of the box. Instead of the physical walls of the box, the shape is determined by the workspace requirements. Advantages of these methods are: high computational speed and precision, the boundary is represented in a simple and accurate way and opposed to the discrete methods, the boundary methods are sensitive to small changes in the design parameters, so optimization algorithms can more easily find out what good changes in the design parameters are. The disadvantage is that the numeric boundary methods are not as straightforward to use as the discrete methods [52].

Application Requirements for a Statue-Scanning Robot

From the requirements of the CaRISA group for a scanning robot of 3D artwork, some requirements directly applicable to CDPRs can be boiled down. These requirements are:

- The dimensions of the workspace including rotation should have the dimensions of the yellow zone in figure 2.8. The range of rotation should be 180° around the vertical axis and 70° around the axis that is not the focal axis.
- The end-effector should horizontally be at least 0.3m away from the centre vertical axis of the statue.
- The allowable cable tensions for the wrench-feasible workspace are in the range of $[10\ldots150]N$
- A wrench of 50 N is applied on the end-effector in negative Z-direction, because of the weight of the camera.
- Besides singularity and cable-cable interference constraints, cable-object interference with the statue should be taken into account as well.
- The size of the frame should have a maximum width and length of 4 meter and a maximum height of 2m.

Table 2.1: The number of the 39 reviewed papers that used a performance index or constraint based on the property

Performance Index	Obj. Function	As constraint
Stiffness	9	4
Dexterity	9	0
Tension	6	n.a.

- Only the geometrical parameters can be synthesized, the parameters of the drive train are fixed.

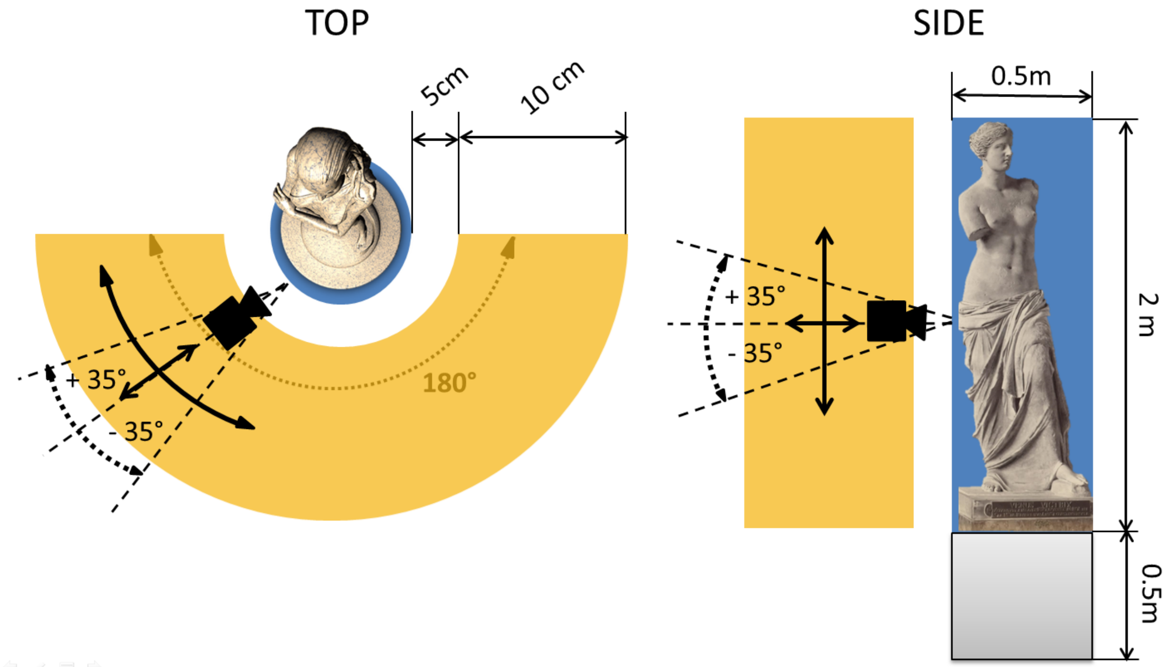


Figure 2.8: A front and side view of the dimensions of the workspace highlighted in yellow, left statue image from [61], right statue image from [65]

2.3.3. Performance Indices

In all found literature the performance indices are somehow related to the workspace, since this was an inclusion requirements. Apart from the workspace volume, many papers also have an index related to either the stiffness, dexterity or tension. How many exactly can be found in table 2.1. Tension is not incorporated as constraint since it is a constraint for the calculation of any workspace.

Performance Indices for Objective Functions

The Workspace Volume The most used peformance index is the workspace volume. There are different ways to describe the workspace volume with respect to translation and orientation. The ones most used in the found literature are:

- Translation workspace: the possible positions of the end-effector for a given rotation [52].
- Total orientation workspace: the possible positions of the end-effector in which a certain range of orientations is possible [52]. In [27] the total orientation workspace is computed. Instead of checking all the orientations for every pose, just the lower and upper limits are checked to see whether the pose is feasible.

The workspace volume is either described quantitatively or qualitatively. A qualitative example would be the percentage of a prescribed workspace reached. To maximize the workspace [47] is minimizing the complement of the wrench-closure workspace (CWCW). Perreault also describes a workspace performance index qualitatively as the percentage of the complement workspace in the total desired workspace.

The most basic description of the workspace is the Wrench closure Workspace. But the workspace has many different forms in literature. In these cases the workspace is subjected to constraints and the constraint workspace volume is in this case the performance index, the Stiffness Feasible Workspace is an example. The most commonly used constraint is the allowable range of cable tension, a workspace under this constraint is the earlier mentioned Wrench Feasible Workspace. Other examples found are the Stiffness Feasible Workspace for which all the poses have a positive definit stiffness matrix [8] or the Gravity Compensation Workspace for which the poses can hold a static ecquillibrium for the wrench caused by gravity [47].

Dexterity as a performance index In [21] the dexterity index is coined as the condition number of the Jacobian matrix. The method to calculate the condition number according to [53] is as follows: First the singular values of the Jacobian matrix **J**: the square root of the eigenvalues of $\mathbf{J^{-1}(J^{-1})^T}$ and $\mathbf{J^{-1}(J^{-1})^T}$ are calculated. The largest and smallest singular value σ_{max} and σ_{min} are then used to calculate the condition number κ:

$$\kappa \equiv \frac{\sigma_{\text{max}}}{\sigma_{\text{min}}}. \tag{2.11}$$

The condition number κ is a value between 1 and ∞. A value close to 1 represents a well conditioned matrix [53]. The condition number tells something about how well the robot is able to exert forces, torsion and stiffness in every direction in that pose. When it is unable to, the pose is close to a singularity and unable to control the manipulator in an accurate way. The global condition index (GCI) reflects the mean condition number of all the poses in the workspace. In [53], the GCI is defined as

$$GCI = \frac{\int_W \frac{1}{\kappa} \mathrm{d}W}{\int_W \mathrm{d}W}. \tag{2.12}$$

The inverse of κ is taken to let the GCI range from 0 to 1.

Another dexterity index introduced by [6] is called the Global Dexterity uniformity index (GDUI) and is expressed by

$$\eta_{du} = \frac{\min(\min(\sigma_i))}{\max(\max(\sigma_i))}. \tag{2.13}$$

The GDUI η_{du} is the lowest singular value found in the entire workspace divided by the highest singular value in the entire workspace, for the calculation of these singular values, the homogeneous Jacobian matrix is used [6]. When the GDUI comes close to its maximum value of 1 the singular values over the entire workspace are very close to eachother and positions in the workspace will have a high dexterity in average.

Stiffness as a performance index Stiffness indices quantify the compliancy of the end-effector to external disturbances. The stiffer the workspace is, the more accurate the end-effector is positioned. Furthermore higher stiffnesses cause higher eigenfrequencies which can be beneficial to not let the robot resonate. The stiffnes is mainly caused by the compliance of the cables [67] Just as the dexterity index, stiffness indices are based on how the Jacobian is conditioned [20, 68]. Different stiffness indices used in literature are listed below.

- The Global Stiffness Index (GSI) is used in [40] and [8]. It is the stiffness index integrated over the entire workspace. The stiffness index is based on the stiffness matrix described by [67]:

  $$\underline{K}(\underline{x}) = k' \underline{A}^\top(\underline{x}) \underline{L}^{-1}(\underline{x}) \underline{A}(\underline{x}). \tag{2.14}$$

  $\underline{L}(\underline{x})$ is the diagonal matrix of the lengths $l_m(\underline{x})$ of the cables. $\underline{A}^\top(\underline{x})$ is the transpose of the Jacobian. k' is the stiffness of the cables in the longitudinal direction. From this formula it becomes clear that the stiffness matrix is depending on the Youngs Modulus, length and diameter of the cables and their orientation with respect to the end-effector [67]. The stiffness matrix has however different units when both including rotations and translations, so has to be homogenized in that case. A method for homogenization is presented in [34]. This method is applied in [3] and yields two matrices, $\mathbf{G_f}$ and $\mathbf{G_m}$. The squareroot of the eigenvalues of $\mathbf{G_f G_f^\top}$ and $\mathbf{G_m G_m^\top}$ yield the stiffnesses in the translational and rotational degrees of freedom, respectively. The stiffness matrix condition number is then calculated as a ratio of the largest and smallest squareroots of the eigenvalues [2]:

$$\kappa = \frac{\sqrt{\lambda_{\max}}}{\sqrt{\lambda_{\min}}} \tag{2.15}$$

The reciprocal of the condition number κ is the stiffness index. The global stiffness index can now be determined by calculating the mean stiffness index over the entire workspace [40]:

$$\mathrm{GSI} = \frac{1}{V}\int_V \left(\frac{1}{\epsilon}\right) \mathrm{d}V \tag{2.16}$$

The number varies from 0 to 1 and says something about how isotropic the stiffness distribution is in all the degrees of freedom. When it is close to 1 the end-effector can resist wrenches in all directions equally well, at every position in the workspace.

- The Overall Stiffness Index (OSI) is used in [40]. For this index [40] first calculates the ratio between the minimum and maximum eigenvalue found in the stiffness matrices **K** in the entire feasible workspace:

  $$\eta_u = \frac{K_{\min}}{K_{\max}} \tag{2.17}$$

  When η_u is close to 1, there are no significant variations in stiffness throughout the workspace. The OSI is then a weighted sum of η_u and the GSI and ranges between 0 and 1, just as its components.

- The lowest natural frequency in a direction compared over all the poses is used in [26]. This method is used since natural frequencies are dimensionless opposed to the components in the stiffness matrix.

- In [18] and [19] the stiffness matrix is used to calculate pose displacement for external wrenches exerted on the platform. These displacements cannot exceed certain limits and are used as constraints. In [45] and [63] the stiffness index is used as well as a constraint. In [63] the stiffness matrix is however not homogenized before calculating the stiffness index.

- In [69] The stiffness in Z-direction for each cable is used as a performance index and a Stiffness Simularity model is used to model the behaviour of the cables since the application is on a very large scale.

- In [2] another performance index M for stiffness is proposed based on the eigenvalues of $\mathbf{G_f}\mathbf{G_f}^\top$ and $\mathbf{G_m}\mathbf{G_m}^\top$. This originally comes from [70] and addresses both the magnitude and isotropy of the stiffness:

  $$M = \frac{\lambda_{\max}\cdot\lambda_{\min}}{\lambda_{\max}+\lambda_{\min}}. \tag{2.18}$$

- In [3] a new stiffness index is proposed which merely focusses on maximizing the product of the smallest eigenvalue of $\mathbf{G_f}\mathbf{G_f}^\top$ λ_{fmin} and smallest eigenvalue of $\mathbf{G_m}\mathbf{G_m}^\top$ λ_{mmin}.

- In [8]another perforance index is used: the Stiffness Feasible Workspace (SFW) as a subset of the Wrench Closure Workspace. It describes the part of the WCW in which the stiffness matrix is positive definite. In the SFW increasing the cable tension increases the stiffness of the end-effector as well.

Tension as performance index The constraint of the minimal and maximal cable tension to compute the wrench feasible workspace is not considered in this part. When tension in the cables can be decreased, the power delivered by the actuators can be decreased as well.

- In [36] the tensions in the cables are added up for every pose. Then the highest value of tension found for a pose in the workspace is used as a performance index. the tensions used are the lowest possible maximum tensions to achieve a static balance at a pose. the highest cable tension among the cables instead of the sum of tensions is used in [29].

- The cable tension is minimized in [37] by minimizing the following normalized quadratic sum of the cable tension vector $\mathbf{T} = [t_1, t_2, \ldots, t_m]$:

  $$F_2(\boldsymbol{I}) = \frac{1}{\max(\boldsymbol{T})}\cdot\left(\boldsymbol{T}^{\mathrm{T}}\cdot\boldsymbol{T}\right) \tag{2.19}$$

- A tension distribution index that evaluates the variance of the tension distribution of all the cables at a certain pose is presented in [63]. It is called the All Cable Tension Distribution Index (ACTDI) and is brought up by [63] to offer a better reflection of the tension distribution at a pose than the tension factor and global tension index. The ACTDI has to be minimized to be able to control the end-effector more easily. The formula from [63] consist of the tension t_i in cable i and the mean tension $\bar{t}$ and looks as follows:

$$e_t = \frac{\sqrt{\sum_{i=1}^{m} \left(t_i - \bar{t}\right)^2 / (m-1)}}{\min\left(t_i\right)} (i = 1, 2, \ldots, m) \tag{2.20}$$

- The Tension Factor and Global Tension factor are mentioned by [63] and proposed by [49]. The tension factor is the ratio between the maximum and minimum cable tension at a pose. A value close to 1 is implying a good tension distribution and is favourable, when the tension factor approaches zero the pose is close to the workspace boundary [49]. The global tension factor is the mean tension factor over all the poses in the workspace. This index has not been used for the optimal design of CDPRs yet.

Other Performance Indices There is a wide variety of other performance indices. Some examples are:

- A performance index taking both the volume and the location of desired workspace into account is the Distance Cost introduced by [4].
- The amount of cable-cable / cable-object collisions for a prescribed trajectory is used as a performance index by [47].
- The frame size can be a performance index as shown in [18].
- Gouttefarde introduces the distance between the geometrical center of the end-effector and the center of gravity of the end-effector with payload as a performance index which is minimized for a certain workspace. This index indicates how well the payload can be balanced by the end-effector [25].
- In [32] the h-index is introduced. This performance is indicating how large the set of possible wrenches in the workspace is.

Performance Indices for Constraints
The performance indices can be used as constraints as well. In this case one performance index is optimized, while some constraints for other performance indices have to be met. For example: the workspace volume is used as a constraint when a task requires a certain workspace but this workspace can easily be reached. In this case the design can be optimized to acquire a workspace with good stiffness or cable tension properties or a design that is easier or cheaper to build. An example is given in [18] in which the volume of a robot for painting and sandblasting structures is minimized while still satisfying the constraints posed by its task. Other examples of performance indices used as constraints are: the stiffness as constraint when optimizing the WFW [48] and the cable length and lower limit for a frame size when optimizing the GCI and the WCW [15].

Flowchart of the Performance Index Evaluation
In Figure 2.9, a flowchart is presented on how the performance indices are determined with respect to the workspace. In literature, the desired workspace is sampled with a discretized grid, since interval analysis has been left out of the literature review. The conditions for a workspace pose are first checked in the flowchart. When the pose belongs to the workspace, extra performance indices for the pose can be calculated. As soon as the entire grid of the desired workspace has been sampled, the total workspace volume and global performance indices can be calculated.

2.3.4. Structural Synthesis

Structural Synthesis is determining the lay-out of the CDPR. It doesn't define exact dimensions, but the fundamental properties instead. It is not straightforward to choose an architecture type since no literature exists so far that links design specifications to the choice of a certain architecture [28]. First the properties of archetypes with respect to geometrical design and their pro's and con's are listed after which some strategies from literature to choose a certain archetype are presented.

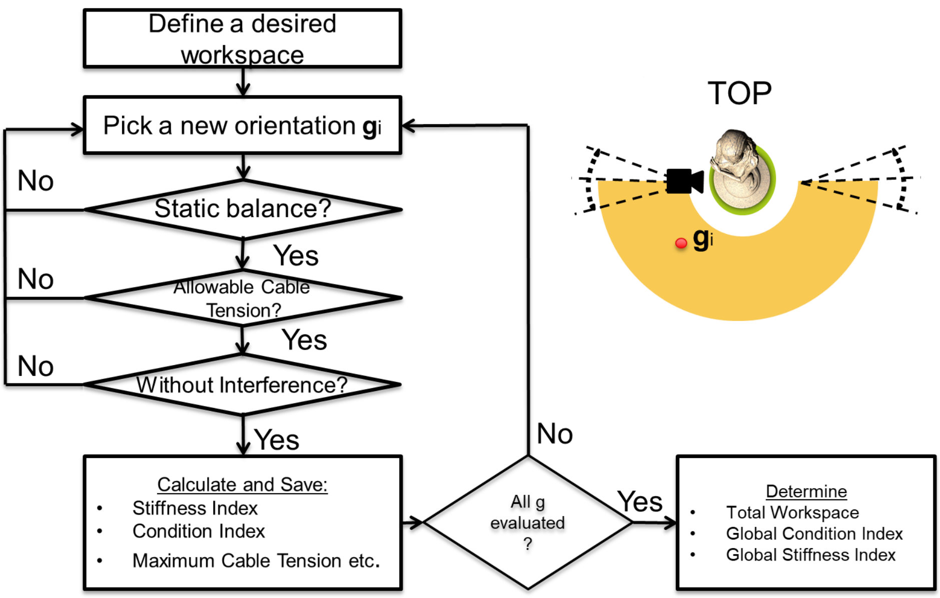


Figure 2.9: Flowchart describing how the performance indices are determined. Statue image from [61]

Categories

The found literature on optimal design processes for Cable-Driven Parallel Robots can be categorized based on two aspects: the architecture subjected to the optimal design process and the optimal design process itself. The differences in architectures used in the 37 optimal design papers reviewed can be found in the motion pattern, suspendedness and restrainedness. The 16 CDPRs with a 2D motion pattern are, except for two CDPRs, not-suspended and at least completely restrained. For the 21 CDPRs with a 3D motion pattern nine suspended versions are presented. Of the nine 3D suspended CDPRs six were incompletely restrained. All the not-suspended 3D CDPRs are at least completely restrained.

Type of restrainedness The advantages of completely restrained positioning mechanisms (CRPMs) and redundantly restrained positioning mechanisms (RRPMs) over incompletely restrained positioning mechanisms (IRPMs) are a high level of control and easier control of the manipulator, more achievable positions and higher accuracy, stiffness, velocities and accelerations [52]. The degree of restrainedness is determined by the amount of cables used.

Amount of Cables Choosing the amount of cables is trade-off between different qualities of the CDPR. In the case of a manipulator with 6 degrees of freedom, a minimum of seven cables is needed to fully constrain the end-effector, because of unilateral force transmission of the cables. In [52] the following advantages and disadvantages caused by the amount of cables are discussed: choosing the minimum amount of cables means that a minimum amount of components has to be used and the costs are reduced, also cable-cable interferences are less likely to happen. But increasing the amount of cables to eight or more is vastly increasing the size of the workspace with respect to the frame size. Because of this reason, an architecture with eight cables is used in [47]. Moreover, adding cables makes the workspace more customizable and decreases the chance of singularities. Some cables can, however, be slack at some times because of the redundancy. In addition, extra cables can carry more load [52]. In [15, 27] an example is presented for which the total orientation workspace increases because of adding an extra cable.

Attachment Points of Cables By providing relations between the attachment points of the cables, the kinematic equations can be simplified. This kind of designs are called non-generic design [52]. Possible options would be symmetry, shared endpoints or letting the proximal endpoints or distal endpoints lie within a plane, line or point. In reality a perfect non-generic design cannot be achieved, but it can be approximated or serve as a starting point for further design steps [52].

The sharing of endpoints is described in [52] and in most literature with the X-Y notation. X are the number of proximal endpoints and Y the number of distal ones. When a CDPR has 8 cables, a possible configuration would be an 8-4 configuration, which means that there are eight separate proximal endpoints and just four distal ones, so some cables share distal endpoints. In the case of an 4-8 configuration, some proximal endpoints are shared by cables. Sometimes it is assumed just distal endpoints have to be elucidated, then the configuration is described with the amount of cables at every endpoint. So a square with two cables at every corner would have a 2-2-2-2 configuration [52].

When endpoints are shared by cables it reduces the likelihood of cable-cable interference, which increases the set of the possible poses and the forward kinematics become more straightforward to calculate [52]. On the other hand some control is lost when endpoints are shared and the dexterity decreases. When for example all distal endpoint lie share one point, a platform could not withstand any torsion. When cables are crossed, so when a proximal endpoint is not per se connected to the closest distal one, large ranges of orientation can be achieved and furthermore the total orientation workspace can be enlarged [24].

Many different cable configurations can generally be chosen from, four examples are shown in figure 2.10. In general for a robot with eight cables and eight distinct frame connections, there are 8^8 = 16.777.216 possible connections to the end-effector, but many have to be discarded because of cable-cable or cable-object interferences and the fact that at least three points of the manipulator have to be connected to cables to resist a wrench in every direction [32]. When cables cannot share endpoints, the amount of possible configuration for an eight cable robot are 8! = 40320 configurations, but this can be reduced to 384 when there is the demand of a symmetry axis [25].

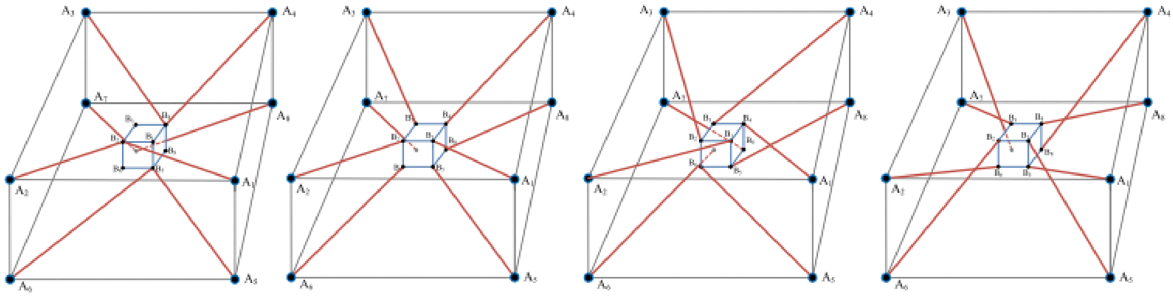


Figure 2.10: Four possible cable configurations for a 3D design with eight cables [32]

Suspension The advantages of suspended CDPRs according to [52] are: there are more cables available for load carrying and there is no collision between the cables and object underneath and next to the manipulator. The disadvantage is: the downward acceleration is limited by gravity, so in general accelerations and velocities are more limited. Also the CDPR is not able to provide any stiffness for some wrenches, for example an upward force [52]. To have more control in horizontal directions, the cables need to have a small incline and are hanging almost horizontally across the workspace. This is not the case for not-suspended robots since more force can be exerted by the cables in a vertical direction, because the vertical counterparts can balance this vertical force to end up with only a horizontal component.

Platform and Frame Structure The box structure of both fixed and movable frame are preferred by researchers since it increases the reliability and symmetry [59]. When performing an optimal design procedure, it might roll out of the equation that the platform should be non-generic and should be degenerated to a lower dimensional shape [48]. A certain shape should be defined as starting point for the optimal design phase. Some structures are deemed to have certain properties. Some non-generic designs such as the 1D or 0D platform structures have less singularities for example.

Symmetry Symmetrical cable configurations are desirable since they decrease the amount of endpoint locations that need to be determined so the computation time in the optimal design phase is shortened because of the decrease in parameters. Furthermore the performance of symmetrical architectures is more homogeneous in case of symmetry [25].

Parameterization

In the case of m cables, there are $2m$ endpoints, also called geometrical parameters, and every one of them has to be assigned to a coordinate within a 2D/3D space. This means that there are $6m$ (3D) or $4m$ (2D) values to be solved by a optimizer. This can be reduced with $2 \cdot 6$ (3D) or $2 \cdot 3$ (2D) to leave out rotation and translation of the whole base and frame, because this does not influence the shape of the geometry [52]. Optimizing 36 parameters is still cumbersome and the DoFs should be reduced by parameterization to decrease the computing time. Only the selected design parameters then need to be optimized by an optimization algorithm. A description of how the architecture is parameterized and the design mapped into a lower dimensional space is described in [52]: the procedure of mapping the design parameters $\mathbf{g}$ to the geometric parameters $[\mathbf{a}_1, \ldots, \mathbf{a}_m, \mathbf{b}_1, \ldots, \mathbf{b}_m]$ with the vector function $\boldsymbol{\Phi}^{\mathrm{G}} : \mathbb{R}^{n_{\mathrm{D}}} \rightarrow \mathbb{R}^{6m}$ is shown in the following equation:

$$[\mathbf{a}_1 \ldots \mathbf{a}_m \mathbf{b}_1 \ldots \mathbf{b}_m] = \Phi^{\mathrm{G}}(\mathbf{g}) \tag{2.21}$$

A straightforward example would be describing the end-effector as a box with width, height, and length instead of 8 endpoints with each 3 degrees of freedom. Then the parameters concerning the distal endpoints are decreased from 21 to 3.

In literature the amount of design parameters varies mostly between 1 and 6, with four exceptions of 11 [5], 13 [29], 16 [1] and 16 [42].

Common Archetypes

In several categories quite some archetypes have been built and tested already. In this paragraph some existing archetypes that fit the application requirements quite well are presented. To summarize, a workspace with a large yaw motion included without singularities and cable-cable interference has to be realised, while there is an object in the middle. The most challenging part are the significant rotations of 180° around the z-axis that most probably will be needed. CDPRs are known for having relatively small orientation ranges [68]. The archetypes presented here will have eight cables since it will be a 3R3T robot and 7 cables will not suffice, since the workspace will be rather small then, an extra cable will decrease the chance of singularities in the workspace and will make the workspace symmetrical. Using any more cables seems unnecessary at this point. Possible archetypes in the category of 8 cables found in [52] are listed below.

- The IPAnema has a rectangular base and frame. The proximal endpoints go to the distal endpoints of the closest vertical edge of the platform, but the lower distal endpoint is connected to the upper proximal endpoint and vice versa. The vertical switch of endpoints improves the stiffness and provides less cable-environment collisions. Within the base no cable-cable collisions occur for zero orientation when using the lay-out of the IPAnema[9].

- The CoGiRo is a project from Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM) and Tecnalia, Montpellier, France [52]. It is a suspended CDPR that has an orientation range of 105° around the z-axis and 80° around the horizontal axes [25]. These results were achieved, however, without having orientation range as a performance index. In general the CoGiRo design is considered as a design with a large yaw range, furthermore the WFW is relatively large and it has a reasonable stiffness despite the suspended configuration. A downside of the CoGiRo is the low hanging cables, because of the suspendedness.

- The IPAnema-Falcon is based on the Falcon design of 7 cables with a platform that is a 1D rod connected with 7 cables to a 2D frame which lies orthogonal to the longitudinal axis of the rod [33]. In the book of Pott [52], the IPAnema-Falcon version is introduced, which has 8 cables to increase the workspace and symmetry. Additionally, the rod does not have a 1D shape anymore, but has become a 3D rod with separable corners instead. This introduces the possibility of a movement of the rod around its longitudinal axis. Just as the Falcon design it has a large orientation workspace for tilting, which would be a large movement around the z-axis, when the planar frame would be placed vertically [52]. Furthermore the translational workspace is larger than the one of the original Falcon because of the added cable and the lay-out of the IPAnema provides a high stiffness [52].

- CableSimulator is a design with a hexagon shaped platform, which is not a suitable design for the statue scanner since the possible yaw of the platform is limited to ±5° because of singularities [52].

Other 3R3T designs with eight cables that were the subject of an optimal design process and mention large yaw angles are:

- The CDPR built for locomotion support by [47]. It is based on the premise that crossed cables are helpful for creating large orientations. While it is not solely optimized for the yaw angle, it already shows a promising range of ± 45°. The optimization algorithm eventually turns the end-effector into a long rod.
- A yaw angle of ± 10° is achieved by [46]. The optimization focuses on a total orientation workspace with ± 10° rotation in every direction. A standard configuration of a cuboid frame and cuboid end-effector with eight distal and proximal endpoints and no cable crossing is used.
- The research presented in [32] focuses on designing different cable orientations to optimize the possible wrenches in a specified workspace, so yaw angles are not the focus of optimization. The rotations achieved with the CDPR are ± 20°.
- A small yaw angle of ± 2° is a WFW constraint in [29], while larger angles are not necessary for the task presented in the paper.

2.3.5. Dimension Synthesis

In theory there are an infinite ways of placing the cable endpoints in different constellations. To find the best solution, the possible configuration should be systematically analysed. The following method is described by [10]: the first step is to define a design space with all the possible solution. Then the previously mentioned performance indices are defined to be able to measure to what extend the robot meets the requirements. The requirements are combined in an objective function which can then be optimized with an optimization algorithm [10]. The optimization loop can be described with the flowchart in figure 2.11. The typical optimization steps are as follows [15]:

1. Create a design space by defining boundaries for all the design parameters. How to determine the set of design parameters was discussed in 2.3.4.
2. Choose a set of design parameters.
3. Evaluate a set of design parameters with the objective function.
4. Based on the evaluation in the previous step, choose the next set of design variables, the optimization algorithm is determining which set.
5. Repeat step 2 to 5 until the optimization algorithm converges or terminates.
6. The set of design variables with the highest objective function outcome presents the optimal solution.

The most important differences in the optimal design process could be found in usage of optimization algorithms and usage of performance indices for the objective function and constraints.

Formulating the Objective Function

The objective function is the function to be minimized by an optimization algorithm for optimal performance of the Cable-Driven Parallel Robot. In most found literature, just one performance index at a time is evaluated with the objective function. When multiple performance criteria are used separately for an optimal design, they could however have reverse effects on the design parameters. An example is given in [15]: optimal design for a high global condition index, leads to a decreased workspace. This is called pareto-optimality: an increase in one performance index automatically leads to a decrease in an other one. Multi-Objective Optimal Design offers a solution. A weighted sum of the performance indices is then used, which quantifies how important optimizing one performance index is with respect to optimizing another one. An example from [47] of a weighted sum with weighing factor k_i of a performance index η_i for n performance indices is:

$$\mathrm{sum} = \sum_{i=1}^{n} k_i \eta_i \tag{2.22}$$

Also constraints can be added to an objective function by multiplying a performance index that does not meet its constraint with a high penalty factor. An example of using constraints in a multi-objective function is found in [37]. In [47] the amount of interferences are included in the objective function. The more interferences encountered, the higher the value of the objective function becomes in this case.

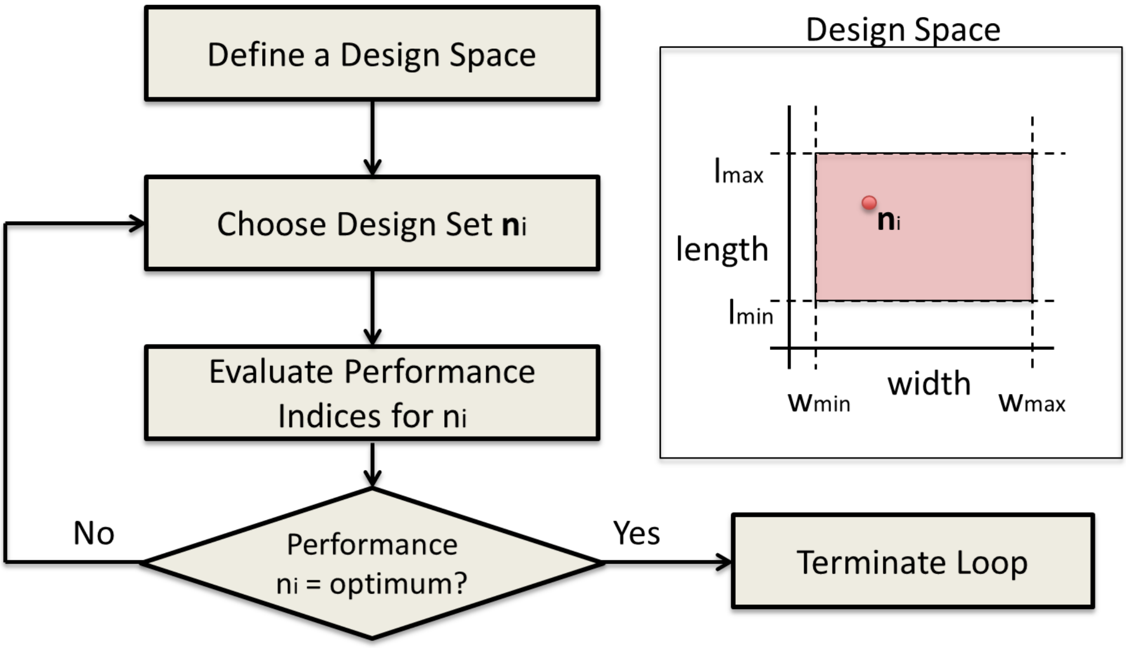


Figure 2.11: The flowchart of the optimal design process with a depiction of a design space.

Optimization Algorithms

The optimization algorithm determines which design sets in the design space are evaluated in a new iteration of the optimal design loop. The design space dimensions and landscapes can wildly vary, which influences the choice of the best optimization algorithm. When choosing the optimization algorithm, a trade-off between viability, accuracy and computing time has to made. In literature many different algorithms are proposed and a selection will be presented in the following section. The most frequently used optimization methods found in literature are the Coordinate Descent Method (GCD), the Genetic Algorithms (GAs) and exhaustive search as shown in 2.2. The exhaustive search method is however not really an algorithm. The heuristic algorithms are mentioned separately, just as the exhaustive search. Local optimization algorithms tend to converge to the closest minimum found in a design space. Global algorithms also sample the rest of the design space to find even lower optima.

Table 2.2: The algorithms used in the 39 research papers for optimal design.

Optimization Algorithm	**No. of papers found**
Exhaustive Search	17
Grouped Coordinate Descend	3
Genetic Algorithm	10
Other	10

Exhaustive Search During an exhaustive search the design space is sampled for different design sets but there is no loop such in picture 2.11. Beforehand a fixed number of design sets is chosen and then evaluated to keep the best solution. This is done by either discretizing the design space by forming a grid and sampling at the grid points or by taking a fixed amount of random samples in the design space. The advantage of an exhaustive search is that the design space is searched thoroughly but it is likely that a true optimum will not be found. The more design sets sampled from the design space the closer an exhaustive search will come to an optimum. In case of a high dimensional design space an exhaustive search cannot maintain a fine grid for every design parameter because the amount of sampled points will be too large. In around half of the found research papers an exhaustive search was found. Either every combinations of design variables is sampled such as in [15], or design variables are sampled one by one, while keeping the other design variables constant at a certain value such as in [43]. The latter method is not finding a global optimum and is more useful for understanding the relation between a design variable and the performance of the robot. The found optimization algorithms are divided into three parts: the exhaustive search, algorithms that always converge to a local optimum and heuristic methods that have no proof of converging.

Local Optimization

Grouped Coordinate Descent Method The Grouped Coordinate Descent (GCD) method suits itself well to find local optima, but this algorithm will remain stuck in a local optimum and then it will not find a global optimum. Apart from this, the algorithm does not suit itself for optimization problems that are non-differentiable [10]. This method is also called the sequential algorithm by [47]. An example of using the Grouped Coordinate Descent method is found in figure 2.12. In the figure either x or y are altered every iterations with the other parameter fixed until an optimum is found [46]. Ouyang presents the following method for the GCD:

$$\begin{bmatrix} W_m^{k+1} & L_m^{k+1} & H_m^{k+1} \end{bmatrix} = \arg \max_{W_m,L_m,H_m} F\left(W_f^k, L_f^k, H_f^k, W_m, L_m, H_m\right), \tag{2.23}$$

$$\begin{bmatrix} W_f^{k+1} & L_f^{k+1} & H_f^{k+1} \end{bmatrix} = \arg \max_{W_f,L_f,H_f} F\left(W_f, L_f, H_f, W_m^{k+1}, L_m^{k+1}, H_m^{k+1}\right). \tag{2.24}$$

First he optimizes the width, height and length of the manipulator W_m, L_m, H_m for a fixed width, height and length of the frame W_f^k, L_f^k, H_f^k. He then uses the optimized the width, length and height of the manipulator $W_m^{k+1}, L_m^{k+1}, H_m^{k+1}$ to find the optimized width, length and height of the frame $W_f^{k+1}, L_f^{k+1}, H_f^{k+1}$. Ouyang performs this separate optimization with an exhaustive search within a range from the value from the previous iteration. So a new L_f^{k+1} is searched for on the interval $L_f^k - 1$ and $L_f^k + 1$ for example. This is repeated multiple times until the algorithm converges to certain dimensions of the frame and platform. Although this is a straightforward algorithm, Ouyang takes 9.39 hours to converge to a solution for optimizing width and heigth of both the base and platform [46]. So care should be taken when deciding for termination conditions and resolution of the exhaustive search within the GCD steps. An enhanced GCD method is also proposed by Ouyang when the algorithm converges to an optimum to make sure there are not better optima close-by. Sampling is then done once outside of the search space of the last iteration.

Dynamic-Q-method The Dynamic-Q-Method is used in [27] and is a gradient based optimization algorithm: the local gradients help to make a decision for selecting the next design set. In [40] however it is criticized because of the likelihood to find only local optima.

Sequential Quadratic Programming The Sequential Quadratic Programming Method is used in [29]. This method mainly focusses on solving optimization problems with non-linear constraints. In the design synthesis of [29] it takes a couple of hours to compute a solution on a standard computer for a 3D case with 13 parameters.

Complex Optimization Algorithm This algorithm is used in [48]. It selects and evaluates a number of design sets and the worst performing one is replaced by its reflection on the centre of all the other design sets. Then this method is repeated until it converges.

The Core Concept Method In [32] the design space is defined by different cable-arrangements. According to Kamali the problem of different cable arrangements can be described by a zero-one multi-dimensional knapsack problem. This problem is defined as a linear sum that's maximized while every part of the linear sum introduces costs that cannot exceed a certain limit. A method called The Core Concept is used in [32] to solve the optimization problem

Heuristic Methods

Heuristic methods aren't able to find minima with 100% certainty, but in practice they often do. During each iteration step their behaviour is determined by predefined rules and the performance of the found solution sets.

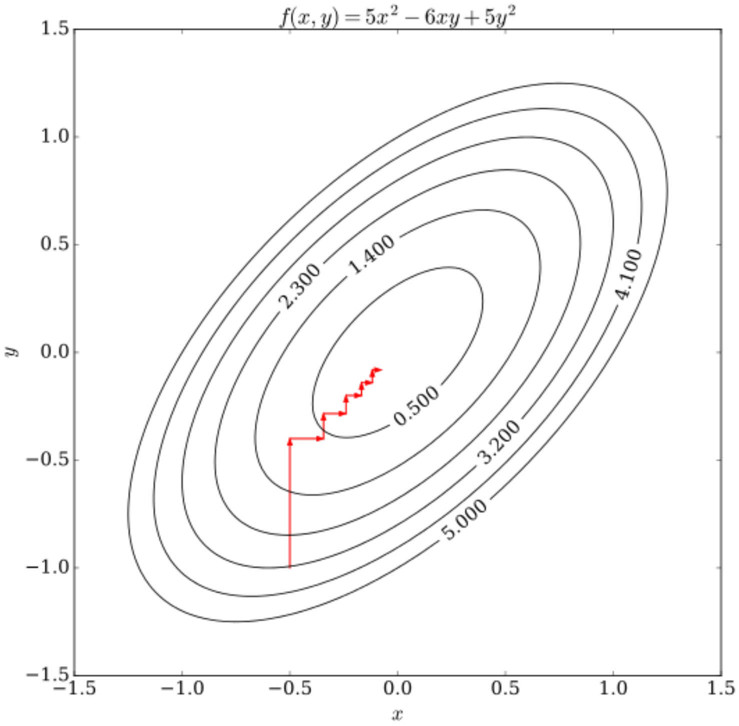


Figure 2.12: Visualization of a coordinate descent method [44]

Genetic Algorithms The standard genetic algorithms are used for discrete problems, but variations for continuous problems do exist [10]. The algorithm is able to solve highly non-linear and discontinuous problems and can also work with constraints [40]. Furthermore it is increasing the likelihood of finding a global optimum. The disadvantages are high computational power because of the high number of design sets evaluated during every iteration [47].

The algorithm is based on evolutionary biology. In [41] In every iteration a group N design space samples are evaluated. The $k * N, k = [0 \ldots 1]$ best ones are selected for the next iteration and will be called parents. To replace the $(1-k) * N$ discarded design sets, new design sets are created by combining traits from the parents in a stochastic way. This is called cross-over and produces a children set. Furthermore there is a chance of random changes in both parents and children, which is called mutation. The set of N mutated parents and children are then evaluated in the next iteration. The best performing parents are sometimes left out of the mutation stage to make sure their beneficial properties aren't lost, this is called elitism and can be very beneficial for convergence. Elitism is used in [38] for example. The four important steps: evaluation, selection, cross-over and mutation are visualized in figure 2.13. in [40] the genetic algorithm is proposed for the optimal geometry design of cable-driven parallel robots for the first time.

The genetic algorithm has a couple of rules or parameters to be determined to set up the optimization. These parts are [40, 58] :

- Chromosome representation: the way in which the design sets that are being optimized will be represented as something that can cross-over and mutate.
- Selection function: according to which rules will certain design sets be selected as parents for the next generation based on their performance.
- Genetic operators: these determine the stochastic processes for how the parents and children cross-over and mutate.
- Population Size: How many design sets will be sampled every generation.

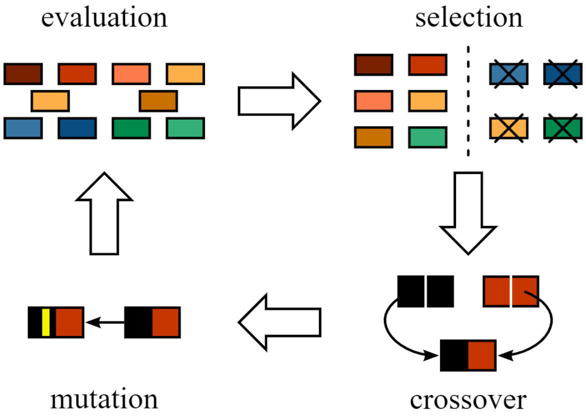


Figure 2.13: Visualization of the steps of an iterations of a genetic algorithm. [12]

- Termination criteria: when to stop the optimization loop. A common criterium is a certain amount of generations.
- Objective function: how will the performance of a design set be evaluated.

The last prerequisite for setting up a GA optimization is the selection to start with, when the selection of design sets is already close to optimum, the algorithm will converge more quickly.

Particle Swarm Optimization Algorithm Just as the genetic algorithms, the particle swarm optimization (PSO) algorithms are suited for non-linear, non-differentiable objective functions. It is a global optimization algorithm and its searching behaviour is based on the behaviour of swarms in biology. A set of multiple design sets is used as a starting value, then new design sets are chosen as if animals are swarming through the design set. The swarming behaviour of an animal, so which set it chooses next, is depending on three factors: his previous speed, the location of the best performing design set it has found so far and the location of the best performing set found by any animal so far. [1].

Pattern Search Algorithm The Pattern Search algorithm is used in [4] and [6] and suits itself well for discontinuous solution spaces. The algorithm searches for better solutions on a mesh around the previous found solution. A better solution is taken as next solution. During the search, the mesh is becoming increasingly fine.

Differential Evolution Method The differential evolution method is used by [50]. The differential evolution method is just as the GA and PSO method a biology based heuristic method that suits itself for discontinuous non-linear problems [60]. Just as the genetic algorithm, the differential evolution method has cross-over and mutation as steps. In [50] a variation of the differential evolution method is presented which is giving preference to design sets that meet all the constraints.

Mixtures of Optimization Algorithms

The different optimization algorithms can be used together to make up for each others weaknesses. A mixture of the GCD algorithm and a genetic algorithm is proposed in [47]. The GCD algorithms converges to an optimum with less computational effort and the genetic algorithm is used to keep exploring the remaining design space to discover other possible minima [47]. The sequential algorithm is used first, after which the genetic algorithm is used. Then they are used within a loop until they both converge to the same solution. In [4] and [6] a mixture of a genetic algorithm and a pattern search algorithm is used. First only the Genetic Algorithm is run to explore the entire design space, but its stochastic properties let the results of different runs vary. To fix this problem a Pattern Search Algorithm is run with the results of the genetic Algorithm to make sure the system converges.

2.4. Discussion

The goal of the result section was to eventually answer the question: *What is the state-of-the-art on workspace improvement of Cable-Driven Parallel robots through the optimal design of geometrical parameters and which methods can be used to obtain the geometrical CDPR design for scanning around a tall 3D obstacle with large rotations?* In the result section an extensive overview is presented on the design process of cable-driven robots. Furthermore the relevant papers and methods in optimal design are discussed, such as different performance indices and optimization algorithms. The design process is focused on obtaining a system design and to be more specific a cable, frame and manipulator configuration. The design process was explained in the results with the following steps:

1. This review focused on optimal design of CDPRs with workspace properties as application requirements, so first the workspace, its properties and its computation were explained. Then the requirements of the CaRISA 3D scanning CDPR were presented

2. The most frequently used performance indices to quantify workspace properties were discussed

3. The design choices for a CDPR architecture were explained. Furthermore, the method to describe the architecture with fewer parameters was presented. In the end, an overview was presented of found architectures related to the application requirements of the CaRiSA project.

4. The optimal design process for CDPRs was explained. First the objective functions and subsequently the found optimization algorithms in literature were reviewed.

In literature, the optimal design process could be categorized by either the architecture or the design process used. The architectures with a 3D motion pattern offered the closest resemblance to designing a cable-driven robot for the scanning of 3D artwork. From remarks found in literature on the architecture it became evident that eight cable not-suspended cable-driven parallel robots are most suitable to have a relatively large workspace with large yaw angles and no cable interferences. Architectures of either the CoGiRo robot of Gouttefarde [25], the IPAnema-Falcon from Andreas Pott [52] or the CDPR for locomotion by Perreault [47] are enabling the largest yaw angles. They do not meet the application requirements for 180° scanning around an object, however, and are not solely focussing on rotation around large yaw angles, so still a lot could be gained here. Furthermore none of the optimal design processes found focus on moving around an external object that is positioned in the middle of the workspace. The architectures are in most literature parameterized by only a maximum of 6 design variables and there are only 4 design processes presented with a higher amount.

2.4.1. Performance Indices

From categorizing literature with respect to the performance indices and optimization algorithms, it became clear that many different performance indices are already available. The four most important categories are: the workspace volume, dexterity, tension and stiffness. The performance of the CDPR with respect to these categories is either separately optimized or in the form of a multi-objective function to have a suitable combination of properties for the CDPR. All of these performance indices could be useful for the scanning of 3D art object, but some are more preferable than others. Firstly the main goal is to achieve a certain rotation around a statue, which means a certain total-orientation wrench-feasible workspace without singularities and cable-cable or cable-object interferences is the most important performance index. Since such a workspace has not been achieved in literature so far, it is not clear whether this will be achieved. In this case the complement of the desired total orientation wrench-feasible workspace could be used as a performance index. When it is possible to achieve such a workspace the stiffness could be incorporated into the objective function as well. A stiff end-effector would be beneficial for the accuracy and to make sure the camera would not swing into the statue because of any disturbances. For this purpose the Overall Stiffness Index introduced by [40] can be used to have a workspace average but to also make sure there are no wild variations within the workspace. Secondly the Global Condition Index should be incorporated to check whether the end-effector is not close to singularities at any position in the workspace so the control will be easier. Tension or energy saving is deemed less important at this stage and should only be added when the CDPR would be performing well considering workspace, stiffness and dexterity.

2.4.2. Optimization

For the optimization, three methods are used most frequently: first the exhaustive search, then the genetic algorithm and as third the Coordinate Descend Method. All the optimization algorithms have their strengths and weaknesses. The exhaustive search is easy to implement and is exploring the entire search space. This is done very coarsely in a rigid grid, so it is not sure whether it will come close to an optimum. Furthermore, when the amount of design parameters increase, it becomes very computationally expensive to explore the entire design space. The genetic algorithm is 'smarter'. It is still able to take samples from the entire workspace but makes educated guesses on which samples. This way it is computationally less expensive, but is still looking around the entire workspace. It is however not completely sure whether the genetic algorithm will eventually converge to an optimum and whether this is the global one. The Coordinate Descend Method is able to converge to an optimum with ease, but only locally and could miss many different optima. It is nonetheless computationally less expensive then genetic algorithms. Since there are no examples in literature of robots that are either capable of moving around an object in the middle of the workspace or showing large yaw angles up to 180° the design will be quite specific, this means that quite some geometrical parameters should be altered. The design space will be multidimensional and will have many local optima. This means that it will be most efficient to use a genetic algorithm, with the possibility of using an extra optimization algorithm that efficiently can converge for local optima.

3

Geometry Design of Cable-Driven Parallel Robots with Large Rotations for the Scanning of 3D Art Objects

Geometry Design of Cable-Driven Parallel Robots with Large Rotations for the Scanning of 3D Art Objects

H.M. Waal

Abstract— **Cable Driven Parallel Robots, or CDPRs, might provide art historians and curators with an improved automated way of scanning 3D art objects. This application requires CDPRs to rotate around a 3D object with large panning and tilting, while avoiding collisions. While no designs exist for this purpose, this article investigates and proposes new cable robot geometries. First, we built a workspace model to quantify the performance of a design. A brainstorm and the ACCREx method then provided us with over 100 promising robot architectures. Their subsequent testing with the workspace model revealed the nine most promising architectures. Lastly, a geometry optimization resulted in the most favourable CDPR geometries. It appeared that mainly cable force limits, cable-statue and platform-statue collision are limiting the workspace of the CDPRs. In the end, the best enclosing and non-enclosing design are capable of reaching 29% and 16% respectively of the required 180° workspace around the statue. Without extra panning and tilting these percentages are 79% and 66% respectively. Both designs show to which extend a statue can be scanned by a CDPR from one configuration.**

I. Introduction

A digital copy of 3D artwork is necessary to understand its history, preserve it, and to make it available for a larger audience. The techniques of the sculptor and the conditions in which artwork was preserved can be deduced from its surface properties through scanning [1], [2]. Furthermore, a detailed digital copy, that, for example, has the same properties under lighting as the original, enables fans and researchers all over the world to enjoy and study the objects [1]–[3]. Finally, the need for restoration and its expected influence can be identified [4]. So far, different constructions have been used to scan 3D artwork. Either a scaffold with a tripod or gantry, only a gantry, or a robotic arm are used to position the scanner. First of all, scaffolds are built around artwork to reach all positions. Operators, however, have to manually place the scanner in different positions on the scaffold, which is time consuming and prone to artwork collisions [5], [6]. Stand alone gantries, for example used in [1], [7], have the advantage of being automated, but are susceptible to cumulative positioning errors because the joints are in series [1], [8]. An industrial serial robot arm can acquire a high accuracy, but is challenging to move and reposition due to its weight and has limited scaling possibilities. Although a rotating disc underneath 3D art solves repositioning, it is not accurate enough [6] and requires possible heavy and delicate artwork to be relocated.

For the scanning of 2D art objects, Cable Driven Parallel Robots, also called CDPRs or cable robots, have been proposed in the work of Tempel [8]. CDPRs are a special type of parallel robots that use cables instead of rigid links to move their end-effector. Figure 1 shows its most important parts: the end-effector or platform, the cables, the winches which coil and uncoil the cables, and a frame which supports the winches. The attachment points of the cables to the frame are called the proximal endpoints and the attachment points to the platform the distal endpoints. A CDPR is called suspended when all cables are moving upwards to their frame attachment points in most positions [9]. A generally well-known example of a CDPR is the SkyCam [10] hanging above the field in stadia to capture events from above. A special property of cables with respect to rigid links is that they are only able to exert forces in unilateral direction. For this reason, cables are efficient in exerting forces and consequently are lightweight. Therefore, CDPRs can accelerate quickly because of the low cable inertia. A CDPR additionally offers a high workspace to frame ratio, is easy to build, reconfigure and scale, and costs less than conventional parallel robots [11], [12]. The unilateral force in the cables, however, restricts the control of the end-effector. Hence, a completely restrained CDPR needs one more cable than the amount of DOFs. A redundantly restrained version requires even more cables. The many cables can interfere with themselves, the end-effector and the environment. Another disadvantage is the flexibility of the cables, as this decreases the accuracy of the end-effector [9] and its stiffness [13].

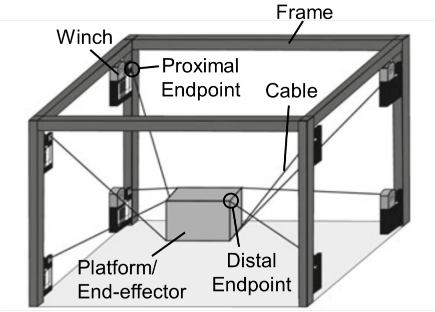


Fig. 1: A schematic image of a CDPR [9]

CDPRs can be advantageous for scanning of 3D objects because they are more accurate than serial robots. When being compared to parallel robots, they are more scalable, have a larger workspace and are easier to build [8]. The design proposed by Tempel is not only able to move along the surface of a painting, but also inspect it from different angles by panning and tilting the end-effector. A next step in line of this research is the scanning of 3D artwork such as statues, vases or pillars. For this application, the two main design challenges are moving the end-effector in a three dimensional space to different sides of the object and enabling large rotations of the end-effector. With these qualities, the CDPR scans the art object from as many angles as possible, so repositioning of the CDPR is minimized.

Moving around external objects, so not merely facing an external 2D plane, is a challenge described in various papers and different categories of solutions are presented. In [12] and [14], cable connections to the frame are reconfigured to operate at different sides of a jacket structure or along the surface of a plane respectively. In other works a robotic arm is attached to the platform to reach locations a static platform is not able to [12], [15], [16]. In [17], the CDPR moves around objects by letting the cables interfere with other external objects, while guarding the original object. To avoid cables from touching external objects while changing positions, planning the path of the platform has been studied too, for example in [18] and [19]. Path planning is more straightforward when the platform is able to move freely above an object, and cables are less likely to interfere with the object when using a suspended configuration. Given these advantages, the suspended configuration of the CoGiRo robot [16] has been used to move around and above objects in a warehouse setting. In [20], a suspended CDPR scans objects from multiple sides.

In literature, other solutions have been found for enabling large rotations of the end-effector. A first solution is reconfiguring the geometry: a circular frame with reconfigurable proximal endpoints can be used for infinite rotations [21] or distal endpoints can be reconfigured passively [22]. A reoccurring problem for large rotations is the collision of cables, which is tackled by releasing cables in [23], [24]. Other designs offer mechanical solutions for rotation, besides actuators on the platform: in [25], a part of the platform is rotated through a differential attached to cables, in [20] a platform is introduced that has a passively rotating part and in [26]–[28] a platform functions as a crankshaft to facilitate infinite rotation. Alternatively, the geometry of the robot can be designed with large rotations in mind. Examples of geometries with large rotations are the Virtual Tennis design [29], the French-German design [30] and the 3-3-2 design [31], which is called the Nishioka design in [13]. In [8], [16], [32], examples are shown of optimizing the geometric design with certain platform orientations as a requirement.

These solutions for object avoidance and large rotations can be part of a final design for 3D art scanning, but they each have different downsides. For the 3D object avoidance, reconfigurations cost time and external objects to deflect the cables make the model more complex and require more material and construction time. CDPRs with rotary actuators or robot arms mounted on the end-effector could offer effective solutions for both rotation and moving around an object, but they add extra weight and therefore decrease the workspace. Suspended CDPRs can only navigate around an object in case of high rooms and low external objects. Furthermore, the end-effector is not as stiff as for a non-suspended configuration and only low downward acceleration is possible.

For large rotations, releasing cables has been proposed besides reconfigurability. This solution, however, complicates the modelling and does not deal with the artwork collision avoidance of cables. Mechanical solutions for rotation, such as a differential, infer added weight to the platform and more cables are necessary for actuation, which could increase the cable collision.

To conclude, solutions for large rotations and 3D obstacles are present, but come they all come with downsides. Furthermore, the existing geometric designs for large rotations are not suitable for rotating around high external objects yet, because of collisions. An optimal design of an 'elementary' CDPR merely consisting of a rigid frame, cables, winches and a rigid end-effector can be beneficial to provide a basis for further research into CDPRs rotating around statues, or tall 3D object in general. However, a geometrical design of such an elementary non-reconfigurable, non-suspended, CDPR without collision, achieved by optimization, has not been presented yet.

The goal of this paper is to introduce a geometric design of a Cable-Driven Parallel Robot for moving around a tall 3D object with large rotations. For the first time, a geometric CDPR design is proposed for this purpose. The used performance index is novel as well: platform rotation is an optimization goal, while it is directly coupled to workspace size. The presented geometries can be used for further enhancements and design for rotation around tall 3D objects.

To find an highly suitable geometric design, first a CDPR workspace calculation model and a performance index were created for evaluation. Then, a collection of possibly well-performing CDPR architectures,or layouts, were designed. The performance of the designs in tests was subsequently used to iterate towards better designs. As a final step, the geometries of the best-performing designs were improved with optimization algorithms. These optimized designs were evaluated and optimized anew when there was room for improvement. The design choices were documented as this is missing in many papers on CDPR design [9].

The kinematics of CDPRs and the corresponding notations are discussed in Section II. Then, the method for formulating a workspace and a performance index can be found in Section III. All CDPR requirements are presented in Section IV to provide a framework for the creation of designs. In Section V, the method for the generation of new robot architectures is presented. With the help of the optimization strategy proposed in Section VI, the design geometries are optimized and redesigned in Section VII.

II. Kinematics

Because of the unilateral forces in the cables, the CDPR is fully constrained when the number of cables m equals the number of degrees of freedom n + 1. With fewer cables, the CDPR is underconstrained and with more, overconstrained.

The notations for the kinematics of a CDPR are taken from the work of Pott [9] and their visual representation is shown in Figure 2. The proximal endpoint A_i of cable i is expressed in the global reference frame $\mathcal{K}_0$ with the position vector $\mathbf{a_i}$. The distal endpoint positions B_i with respect to the platform's local reference frame $\mathcal{K}_p$ are expressed with vector $\mathbf{b_i}$. The distance vector $\mathbf{r}$ and the rotation matrix $\mathbf{R}$ describe the position and orientation of $\mathcal{K}_p$ in $\mathcal{K}_0$. Finally, the cable l_i runs between B_i and A_i and is represented by the vector $\mathbf{l_i}$ in the global coordinate frame. As can be seen in Figure 2, all the discussed position vectors represent a loop. Hence, the an expression for every cable vector can be deduced [9]:

$$\mathbf{l}_i = \mathbf{a}_i - \mathbf{r} - \mathbf{R}\mathbf{b}_i \text{ for } i = 1, \dots, m. \tag{1}$$

the wrench on the platform $\mathbf{w_p}$ consists of the external forces $\mathbf{f_p}$ and the external torques $\tau_\mathbf{p}$. For a static equilibrium, the wrench equals the force and torques that the cables exert on the platform center. These cable forces on the platform center are expressed by the multiplication of the structure matrix $\mathbf{A}^\mathrm{T} \in \mathbb{R}^{n,m}$ and the cables force vector $\mathbf{f}$. The matrix $\mathbf{A}^\mathrm{T}$ is composed of $\mathbf{b_i}$ and the cable unit vectors $\mathbf{u_i}$. When combined, the static equilibrium equations are

$$\mathbf{A}^\mathrm{T}\mathbf{f} + \mathbf{w_p} = \mathbf{0}. \tag{2}$$

Inserting all the expressions transforms this equation into

$$\begin{bmatrix} \mathbf{u}_1 & \dots & \mathbf{u}_m \\ \mathbf{R}\mathbf{b}_1 \times \mathbf{u}_1 & \dots & \mathbf{R}\mathbf{b}_m \times \mathbf{u}_m \end{bmatrix} \begin{bmatrix} f_1 \\ \vdots \\ f_m \end{bmatrix} + \begin{bmatrix} \mathbf{f}_\mathrm{P} \\ \boldsymbol{\tau}_\mathrm{P} \end{bmatrix} = \mathbf{0}. \tag{3}$$

The cable mass and the associated sagging will not be considered during this research because this level of accuracy is not required for geometric design [33].

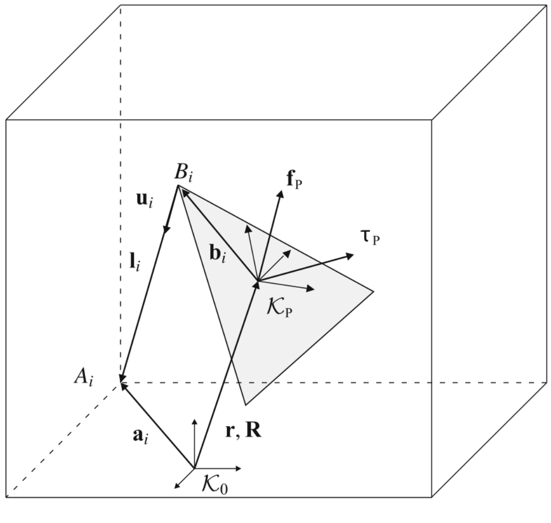


Fig. 2: The notation for the kinematic analysis of CDPRs, taken from [9].

III. Workspace Model

In this article, the workspace is defined by all the poses of the end-effector that are reachable from the platform's starting position. To incorporate both positions and rotations in a 3D dimensional workspace representation, the total orientation workspace $\mathcal{W}_\mathrm{TO}$ is used. It comprises of all the positions where the range of required orientations $\mathcal{R}_0$ is possible. A mathematical description of the total orientation workspace, with $\mathbf{y}$ representing the pose, is [9]:

$$\mathcal{W}_\mathrm{TO}\left(\mathcal{R}_0\right) = \left\{\mathbf{r} \in \mathbb{R}^3 \quad | \quad \mathbf{y} = (\mathbf{r}, \mathbf{R}) \, \forall \, \mathbf{R} \in \mathcal{R}_0\right\}. \tag{4}$$

At a location in the workspace, the camera in its neutral pose needs to face the longitudinal axis of the statue and pan and tilt around the point on the statue surface in between the camera and center line. This is illustrated in Figure 3. The negative and positive panning and tilting poses are all separately evaluated at their extremes. The different poses 0 - 4 to evaluate for a platform location are depicted in Figure 3. Possible interference while moving to the neutral pose and from the neutral pose to the panning and tilting poses is evaluated too. Because the required workspace is symmetric, only half of it has to be evaluated, which is illustrated in Figure 4. This only applies, however when the CDPR is symmetric too.

To explain the workspace model, the Hull method is introduced first to determine the workspace boundary. Subsequently, the minimum and maximum cable force and singularities are presented as workspace constraints. When moving to another pose, collisions between cables, platform and external objects can take place. They are presented as the other workspace limitations. At the end of this section, a performance index is proposed to relate the workspace model to the requirements.

A. The Hull Method

The hull method introduced from [36] is used to accurately calculate the workspace boundary in a quick, accurate and efficient way. During the first step of the hull method, the middle of a icosahedron is placed at the hull method's starting position inside the workspace. The resolution of the hull is determined by how many times the polyhedron's triangles are subdivided into four smaller triangles. Another word for resolution is the recurrence level. At zero, the lowest level, the icosahedron consists of 12 vertices. Next, the vertices of the shape are extrapolated outward until the the workspace requirements are violated. Not only the neutral vertex pose is evaluated, but also the separate poses of the panning and tilting positions. In the end, the vertices are triangulated to form a workspace boundary or hull. To start the hull method in the middle of the required workspace, the CDPR must be able to move from its starting position to the starting position of the hull method, which is visualized in Figure 4. To define the location of the starting point of the hull method, the starting point of the platform is shifted around the longitudinal axis of the statue with a certain angle.

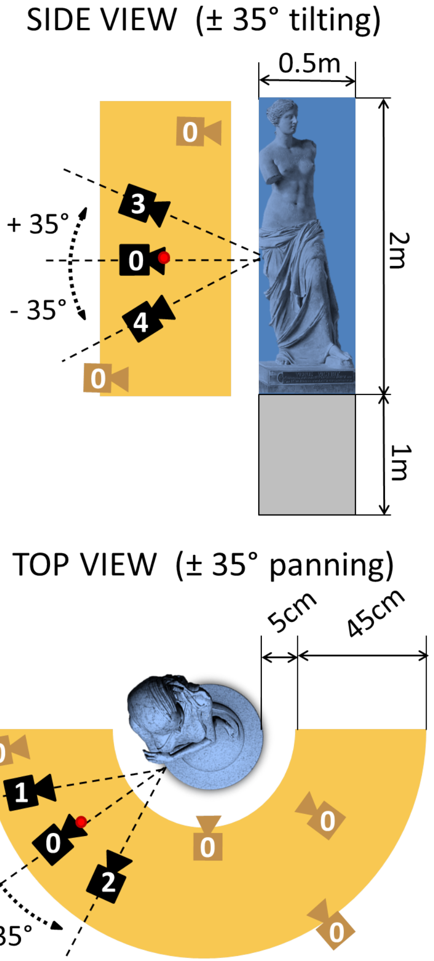


Fig. 3: The most important workspace dimensions for the CDPR and the depiction of the five poses 0 - 4 that need to be evaluated for one random location, depicted with a red dot. The top view of the statue was taken from [34], the front view from [35].

B. Wrench-Feasible Workspace and Singularities

In operation, a minimum and maximum cable force is dictated to prevent significant cable sagging, as well as damaging the cables, pulleys, winches, and actuators. When using a CDPR for scanning artwork, the end-effector will move with low velocities and accelerations since this decreases vibrations and the covered distances will not be large. For this reason, only statics will be considered during this research. The Wrench-Feasible Workspace (WFW) is defined by all the poses for which the cable forces are in static equilibrium with external wrench $\mathbf{w_p}$ and do not exceed the cable force limits. When the number of cables m exceeds the degrees of freedom n, the system of equations in Equation 2 is under-constrained and there are an infinite number of varieties of cable force distribution. At a pose, Linear Programming is used to evaluate whether a force distribution within the cable force limits is possible under a certain wrench. Linear Programming is a useful method because it works for any redundancy or cable force and has a bounded computation time. Yet, it does not calculate force distributions that are continuous along a trajectory and is not capable of doing real-time calculations [37]. These downsides are, however, not relevant for this work. The negative and positive panning and tilting poses are all separately evaluated at their extremes.

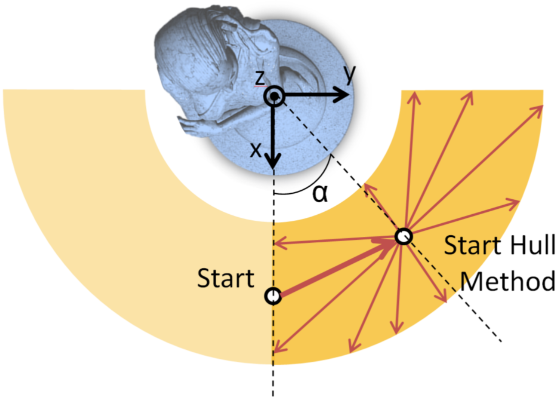


Fig. 4: Depiction of evaluating whether the hull starting point can be reached and of the Hull being expanded in the required workspace. The statue picture originates from [34].

C. Collisions

Collisions, or interferences, need to be considered to provide a model that is limited in its movements in the same way as the real-life model. Interference is either damaging the components or adding conditions that hinder the control of the robot. For this research, interference detection was implemented in multiple ways and subsequently tested. The model finds any possible interferences on the path between the center of the hull and the vertices, as well as between the vertex location and the corresponding panning and tilting locations. This means, however, that on other trajectories in the workspace, interferences can still occur.

1) Cable-Cable Interference: To detect cable-cable interference, two methods are combined. The first method is the numerical method introduced by Perreault [32], which detects cable-cable collisions at a pose and in between. Two cables collide when all of the following three conditions relating to Figure 5 are satisfied [32]:

1) $0 \leq n_1 \leq$ Length Cable 1
2) $0 \leq n_2 \leq$ Length Cable 2
3) $|n_r| \leq \epsilon$ or n_r changes sign while moving between two poses.

When the values n_r, n_1 and n_2 are multiplied with their corresponding unit vectors, they form the vectors $\mathbf{n_r}$, $\mathbf{n_1}$ and $\mathbf{n_2}$ from Figure 5. Their formulas are as follows [32]:

$$\mathbf{n}_r = n_r \frac{(\mathbf{n}_1 \times \mathbf{n}_2)}{|\mathbf{n}_1 \times \mathbf{n}_2|} = n_r \frac{(\mathbf{u}_1 \times \mathbf{u}_2)}{|\mathbf{u}_1 \times \mathbf{u}_2|} \tag{5}$$

$$\mathbf{n}_1 = -n_1 \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|} \tag{6}$$

$$\mathbf{n}_2 = -n_2 \frac{\mathbf{u}_2}{\|\mathbf{u}_2\|} \tag{7}$$

To find the values for n_r, n_1 and n_2, the system of equations

$$\mathbf{n}_1 + \mathbf{n}_r - \mathbf{n}_2 - \mathbf{r} = 0 \tag{8}$$

has to be solved [32].

The maximum allowable distance between the center lines of two cables is defined by ϵ. A sign change of n_r between two poses indicates that the two cables have crossed.

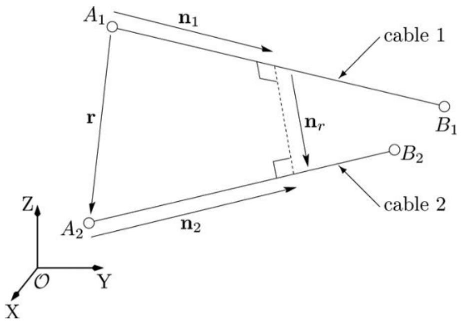


Fig. 5: Analyzing the distance between two cables, taken from [32].

For large rotations and translations, the sign of n_r could change even while a collision occurred outside of the lengths of the cables. In this case a false detection takes place. The method from Nguyen [38] offers a different approach for evaluating interference between poses. Two cables interfere between platform poses when the following conditions are both satisfied:

1) There is a change in s_{ij}, defined by [38]:

$$s_{ij} = \text{sign}\left(\left(\overrightarrow{A_iB_i} \times \overrightarrow{A_iB_j} \right)^T \cdot \overrightarrow{A_iA_j} \right). \tag{9}$$

So in case of a sign change, cable A_jB_j moved to the other side of the plane $A_iB_iB_j$. This is visualized in Figure 6

2) When the cables are projected onto each other at the new pose, they intersect within their own lengths. This is depicted in Figure 7 The projection of cable j (A'_jB_j) on plane $A_iB_iB_j$ is calculated with the three following equations [38]:

$$A'_j = A_j - t\,\mathbf{n}, \tag{10}$$

$$\mathbf{n} = \overrightarrow{A_iB_i} \times \overrightarrow{A_iB_j}, \tag{11}$$

$$t = \frac{\mathbf{n}^T \cdot \overrightarrow{A_iA_j}}{\mathbf{n}^T\mathbf{n}}. \tag{12}$$

The downside of Nguyens method is that a minimum cable distance is not considered, while cables should maintain a minimum distance in the workspace for safety reasons.

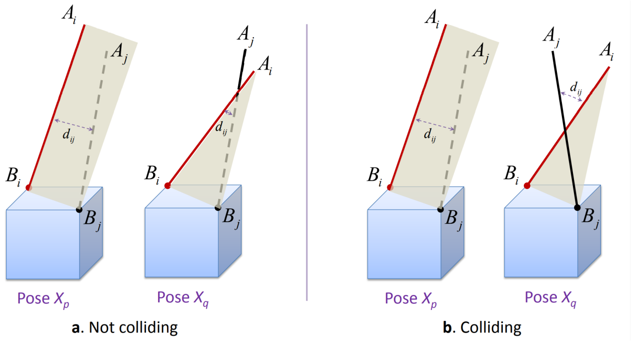
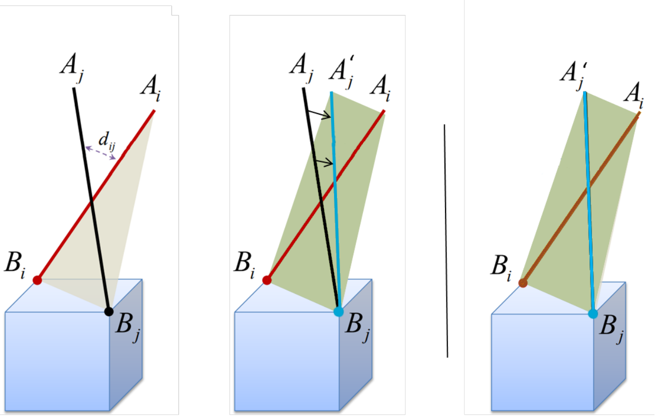


Fig. 6: In **b.** a collision has taken place between pose X_p and pose X_q. Taken from [38].



A_iB_i and A_jB_j A_jB_j projected onto A_iB_i $A_j{}'B_j$ crossing A_iB_i

Fig. 7: Before, during and after the projection from left to right. Edited version of a picture of [38].

Hence, Nguyens method was used, combined with the cable distance conditions of Perreault, so his first two conditions. In the new combined algorithm, cable pairs that share common connection points are not considered because they cannot cross. Furthermore, parallel cable pairs are not evaluated either, because their cross product is zero. The new algorithm first tests whether cables have collided while moving to the current pose and then whether they are colliding at this pose. Poses with rotated orientations always have the neutral pose as the previous pose and neutral poses always have another neutral pose as previous pose.

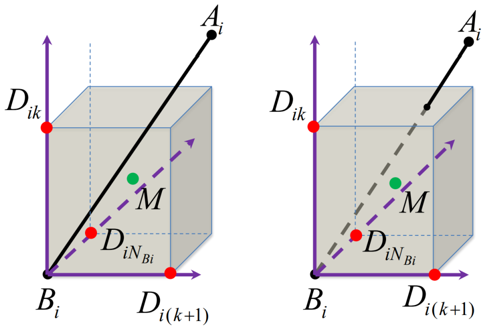
To test the algorithm, a benchmark algorithm is used that takes incremental steps that are smaller than the cable width, instead of using the LineSearch method. In this case, the algorithm tests only at a pose for cable collisions. The comparison with the benchmark algorithm reveals that the proposed algorithm falsely detects collisions in some cases, because the projection method cannot flawlessly reveal what has happened between poses. In very rare cases collisions were not found because cables changed sign twice between two poses. Hence, the algorithm was considered to be on the conservative side. The lack of precision compared to the benchmark algorithm was compensated for by a significant increase of speed. The comparison of different cable-cable detection algorithms with the benchmark algorithm are found in the appendix.

2) Cable-Platform Interference: Depending on the platform, two different cable-platform interference detection algorithms are used based on the platform type. The first algorithm was introduced by Nguyen [38] and is computationally efficient and reliable. However, it only works for convex shapes. This method selects for each distal endpoint B_i the closest vertices of the platform, which are D_{ik}, $D_{i(k+1)}$ and $D_{iN_{Bi}}$ in Figure 8. Together with its closest vertices, a distal endpoint represents a convex cone in which the platform lies. The position of a point M inside the cone with respect to the faces of the cone can be represented by calculating the sign between M and the faces. This is done with the following formula:

$$S_{Bi}(k) = \text{sign}\left(\left(\overrightarrow{B_iD}_{ik} \times \overrightarrow{B_iD}_{i(k+1)}\right)^T \cdot \overrightarrow{D_{ik}M}\right). \quad (13)$$

At every pose, the same is done for all the proximal endpoints A_i instead of M. When the signs of a proximal endpoint A_i match with the signs of point M inside the cone, the cable i collides with the platform. When one of the signs is zero, cable i collides as well.

For platforms that are not a convex cuboid, a method introduced by Perreault [32] was used. It considers the platform sides as cables and evaluates whether collisions happen between the cables and the platform edges. For this purpose, the formerly introduced combined cable-cable collision detection algorithm was used. This method requires longer computation times, however, when the number of cables and platform edges increase. Moreover, the flaws of the cable-cable algorithm still apply.



a. Not colliding **b.** Colliding

Fig. 8: Visualisation of the detection of a cable-platform collision, taken from [38].

3) Cable-Statue Interference: To research whether a CDPR can rotate around a structure, a cylinder was used to approximate a statue as was done in [39]. This approach simplifies the cable-platform collision detection significantly. In this way, the cylinder can be seen as a thick cable and cable-cable collision detection can be used. The cable cable algorithm was extended to detect cables that cross the top and bottom of the cylinder, while having their closest point to the statue outside of the statue length. The cable-statue collision detection algorithm evaluates at every position whether a collision takes place or has taken place while moving there.

For statues with protruding parts or cavities, an algorithm has to evaluate many faces for possible collisions. This increases the number of computations. In order to save computational effort, objects can be approached by volumes with less faces that fully incorporate the object [38].

4) Platform-Statue Interference: The last type of collision occurs between the platform and the statue. To detect most of the collisions, interference of the platform edges with the statue is detected with the same principle of the cable-statue collision detection algorithm, while considering the platform edges as cables. Additionally, the cable-platform algorithm evaluates whether vertices of the platform interfere with the statue at a certain pose. Because the statue is large and interference is not easily missed, the platform-statue collision between poses is not evaluated by the means of sign change detection. To not miss platform-statue collisions during the first large steps of the LineSearch, the LineSearch algorithm evaluates at multiple points during the first two iterations. Similar to the cable-platform interference, this detection method benefits from the simplification of the statue with a cylinder.

D. Accelerating the Interference Calculation

To decrease the computation time, collisions were evaluated only for relevant cases only for relevant cases. For each design, cable pairs for cable-cable interference, cables for cable-statue interference and platform edges for platform-statue and cable-platform interference were selected. It was assumed that only this selection is in risk of collision.

E. Combining the hulls

Because the workspace is calculated thousands of times during one optimization, a quick calculation of the workspace accelerates the process significantly. The expression 'useful workspace' is used here, which describes the volume in which the final workspace should lie. It is limited by the required workspace, as well as the minimum of all the limiting workspaces, e.g. the WFW or the cable-cable interference workspace. To calculate the workspace efficiently, the useful workspace served as outer boundary for the LineSearch method. While starting at the outer boundary, the LineSearch algorithm then moves inward into the useful workspace for evaluations. In this way, the LineSearch method does not evaluate unnecessary locations which will not belong to the final workspace. The WFW is calculated first while it is the most limiting workspace. Hence, subsequent workspace calculations, such as the one of cable-cable interference, are done in a smaller useful workspace. For many vertices of the hull, the LineSearch will then converge at the first evaluation already.

F. Performance Index

Since this article investigates to what extend a CDPRs can rotate around an external object, the design process focusses on maximizing the resemblance of the required workspace and not workspace quality. Because the workspace is symmetrical, only half of it is evaluated. A reference hull is created within the required workspace, with every hull vertex coinciding with the required workspace boundaries. The CDPR's hull is created from the same starting point. The ratio of the CDPR's hull with respect to the reference hull is the complement of the workspace, a performance index introduced in [32].

IV. Geometry and Workspace Requirements

Before creating the designs, boundary conditions were set. The exhibition hall has a width and length of six meters and a height of four meters, so the proximal endpoints have to remain within this space. The platform weighs 5 kg and does not exceed 2m in length, width or height. Additionally, only xz-plane-symmetric and non-suspended designs were evaluated. A frame on only one side of the statue and cables that do not enclose the statue are preferred to make the design usable for different statue shapes. A maximum of eight cables was first chosen as a soft constraint, representing a trade-off between a larger workspace [13], and collisions and the cost of extra components. Designs with crossing cables were ommitted since they are difficult to design and optimize for the large rotations of this application.

The statue is approximated by a cylinder of 2 m high and 0.5 m in diameter on a 1 m high pedestal. In the end, the pursued workspace is half of a hollow cylinder around the statue. The minimum distance to the statue is determined by the maximum focal depth of the microscope [40] and the thickness of the pursued workspace is chosen so the SpecimIQ spectral camera [41] can take pictures from a maximum of 0.5 m distance. The vertical size of the required workspace is determined by the statue size. An overview of the dimensions is shown in Figure 3.

At a pose, the camera faces the statue central axis in the neutral position. Moreover, it is required to tilt and pan $\pm 35^\circ$ degrees in both directions, while facing the same statue surface location as in the neutral position. While moving to a pose, or moving to a panning and tilting position of a pose, interference by cables or the platform are not allowed. Furthermore, the cables can only exert forces between 5N and 150N. The starting point of the platform is positioned in the middle of the workspace: at a height of 2 m, and 0.525 m away from the longitudinal axis of the statue. The angle which determines the starting position of the hull method is depending on the platform size. For platforms with dimensions smaller than 1 m, the angle is 5 degrees, for platforms smaller than 2 m 10 degrees and 20 degrees for rotations without panning and tilting. This angle α is shown in Figure 4.

V. CDPR Design

During the design phase, feasible cable lay-out and platform shape combinations are determined. The theoretical solution space of cable layouts for eight cables (k) and eight possible proximal and distal endpoints (A, B) is already more than 4 billion possibilities, which was calculated with the formula for combinations:

$$\frac{(A \cdot B)!}{k!((A \cdot B) - k)!}. \tag{14}$$

The number of possibilities can be reduced with rules and symmetry, but the performance of the remaining cable layouts is also depending on the geometry. This makes the solution space too large to explore with an optimization algorithm. Instead, a design process method is used. First, a brainstorm was conducted with the help of literature. Inspired by the ACCREx method [42], the brainstorm results were categorized, extended and an initial selection was tested. Based on the results, seven promising designs were selected to be optimized.

A. Design Creation

For the first designs, inspiration was taken from existing designs and research on designing CDPRs. Steps and considerations for designing a geometry, such as decoupling the rotation of the platform, were taken from the work of Lafourcade [43]. Verhoeven [13] introduced several design guidelines as well. One of them tells that endpoints of cables should be put together as much as possible to enable large rotations and prevent cable-cable collisions. Another states that the shape of the workspace can be improved by adding redundant tendons. Beside giving guidelines, Verhoeven introduced and analyzed different designs in his work, which served as inspiration. Especially the Nishioka design from [31] has the useful properties of large rotations and absence of cable-cable collisions. The designs from [8], [32], [44] served as inspiration as well. Three different kinds of design templates were created: 2D T-platform designs, with three distal endpoints, 3D H-Platform designs, which have four distal endpoints to exert a rotation on both sides of the platform and 3D C-Platforms, which can have distal endpoints on the edges and vertices of a cuboid. A brainstorm on cable lay-out was done using pen and paper with printed templates. The templates and schematic versions of the platform types are shown in Figure 9.

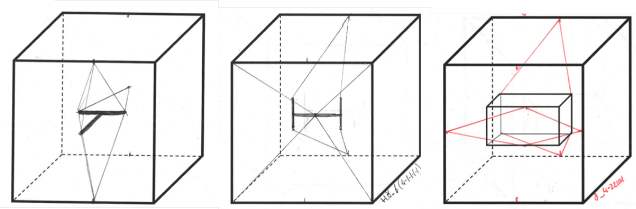


Fig. 9: Templates that were used for the brainstorm. From left to right the T-platform, H-platform and C-platform are depicted.

After the brainstorm, infeasible designs with obvious collisions, interferences or singularities were filtered out and the remaining designs were categorized. Firstly, designs were sorted in three tables based on platform type, with the number of cables on one axis and a distinctive feature on the other. For T-platforms, the feature was the number of cables sharing a proximal endpoint at the mirrored endpoints of the T. The number of shared distal endpoints on the other side of the statue categorized the H-platforms. The categories of the C-Platforms were based on the cable lay-out and the design that served as inspiration. Within the table slots, the designs were grouped once again for common features. Features include the location of the endpoints, the number of shared distal and proximal endpoints and the lay-out of the cables. Inspired by the ACCREx method, missing combinations of features that pose feasible designs were added to the tables. Furthermore, the categories of the features were extended. To illustrate, when three cables share an endpoint, solutions with two, four, five or more cables at that endpoint could be good solutions too. The tables per platform can be found in the Appendix.

B. Testing the Designs

During the evaluation of the designs, the added wrench on the platform was 49.05 N in the y-direction and a moment around the x- and y-axis based on the platform dimensions. The complement workspace with ±35° panning and tilting was used as a leading requirement, but not all the designs could meet this requirement, so different combinations of panning (p) and tilting (t) requirements were tested. These combinations were: $0p0t$, $35p0t$, $0p20t$, $35p20t$, $0p35t$ and $35p35t$. Features of well performing designs were combined to produce better designs. In this way, not all feasible combinations of features had to be tested, merely combinations of well performing features. In the end, more than 100 designs were tested.

C. Results

The tests revealed insights for designing. First of all, crossing cables can enlarge the WFW, which is depicted in Figure 10. The positive effect on the workspace was, however, undone because the cable-cable interference workspace decreased. When adding extra cables to an eight cable design, cables that went straight to the top or bottom of the frame increased the WFW most significantly. From tests it also appeared that half or more of the cables should be going up to withstand the wrench caused by the weight of the platform. Cable lay-out features that provided good results culminated in the designs chosen for optimization.

From every platform type, at least one well-performing design was selected for optimization. Every design has eight cables to make an honest comparison. Additionally, eight cables is a good trade-off between workspace size and number of cables [13]. To limit time spend on optimizations seven designs were chosen for optimization. The seven selected designs and reasons for selection are listed below. Pictures of the optimized designs can be found in Figure 11.

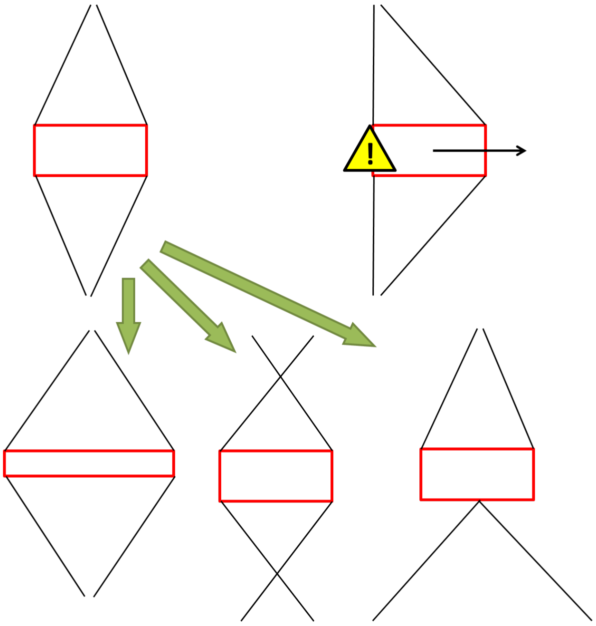


Fig. 10: Different solutions to implement cable-cable-platform triangles for rotations, without decreasing the translational workspace.

- **T-Nishioka**: A variation on the Nishioka design from [31] was chosen, because it was recommended for large rotations by Verhoeven [13]. This orientation of the Nishioka design produced the best results for ±35° panning and tilting.
- **T-Falcon**: this version produced the best results of all eight cable T-platforms for large tilting.
- **T-Lafourcade**: this design had the largest workspace of all eight cable T-platforms for panning without tilting.
- **H-Platform**: The best results for H-platforms with eight cables came from this design. Not more H-platforms were selected, because variation between the H-platforms was low.
- **C-IPAFalconX**: Chosen as a variation on the IPAnema-Falcon design which capable of both large translation and orientations [9]. The proximal endpoints are in the xz-plane.
- **C-IPAFalconY**: an IPAnema-Falcon version with its proximal endpoints on a plane facing the statue. The camera is somewhere within the volume spanned by the endpoints.
- **C-Carisa**: Based on the CDPR introduced by Tempel for the scanning of 2D art [8].

VI. Optimization

The differential evolution algorithm of the python SciPy Optimize library was used for geometry optimization, because it searches for global minima of continuous, multi-dimensional black box problems [45]. Gradient based optimization algorithms are not suitable for this problem. The local minimization at the end of a differential evolution optimization could not converge and was left out. The population size was set to 15. A larger population searches the solution space more extensively, but requires more computational time. The differential algorithm evaluates the objection function a maximum of $population\_size \cdot optimization\_dimension \cdot maximum\_allowable\_iteration$

times. The allowable number of function evaluations was limited to not span more than 48 hours, but took less in most cases. Since the differential algorithm is a heuristic method, convergence to the global minimum is not guaranteed. The input of the differential algorithm is the changeable parameters, their bounds, constraints and the workspace model. The computation time of one run of the workspace model was 15 - 20 seconds for a recurrence level of 1. The performance index used is the complement of the pursued workspace, introduced in section III-F.

Parameterisation, Bounds and constraints

A minimum in the solution space is depending on the coordinates of all the endpoints, these are called the dimensions of the problem. By changing one dimension only a local minimum can be found. For all the distal and proximal endpoints, there would be 42 parameters. The differential evolution would take too much time to converge to a solution for all these parameters. The algorithm then needs to evaluate too many points to sample the search space thoroughly enough for convergence at a local minimum in the vicinity of the global one. Furthermore, bounds and constraints for limiting the search space are easier to impose when a general idea for a design has been conceived already. To achieve this, many parameters can be discarded by taking symmetry into account. With extra assumptions and choices, the number of parameters was kept as low as possible, varying between three and eight. Highly symmetrical designs such as the T-Falcon design needed less parameters than the T-Lafourcade design for example. The parameters of the designs are shown in Figure 11. For the final choice of parameters different reasons dominated:

- **T-Nishioka**: From tests it appeared that a thin platform produces the best results. The cables are attached to the vertices of the triangle. One vertex coincides with the camera, the position of the other two are described with $h1$, $w1$, $h2$ and $w2$. The vertex with two distal endpoints is always above the one with three. The two proximal endpoints at the bottom are fixed relatively to each other, but can move along the x direction. The other two groups of proximal endpoints are mirrored in the xz-plane. The y-value or width of the frame was maximized in earlier trial optimizations, so it is left out of the parameters for faster convergence.
- **T-Falcon**: The four groups of distal endpoints are only described by x_0 and y_0. They are mirrored in the xz-plane. In earlier optimizations, it already appeared that the frame height should be maximized, so 4 m in this case. Tests showed that the platform height should be low, 0.02 m here. The parameters wp, lp and lp_b are used to describe the platform dimensions and camera position.
- **T-Lafourcade**: The three proximal endpoints on the grey rectangle were assumed to have the same x-value to lower the dimensions of the optimization. The rectangle height is 4 m. The height of the platform is 0.02 m, to leave a small space between distal endpoints. Tests revealed that the height should be as small as possible. Besides shaping the platform, the parameters lp and lp_b also determine where the camera is positioned.
- **H-Platform** The height of the proximal endpoints is 4 m, because of earlier testing results. All proximal endpoints have a y-value of zero, so only x_0 and x_1 are frame parameters. The lengths hp, wp and lp determine platform size and camera position.
- **C-IPAFalconX**: For the same reasons as for the H-Platform, the proximal endpoints are described by x_0 and x_2. For this design, the height is a parameter as well. The length of the platform and the position of the camera is determined by lp and lp_b.
- **C-IPAFalconY** The proximal endpoints are similar to the ones of the T-Falcon design and have the same parameters. The platform shares the same platform parameters with the C-IPAFalconX design.
- **C-Carisa**: The proximal endpoints are described in the same way as for the earlier Falcon designs. Earlier optimizations showed that the platform should have minimized height. Another previous optimization showed that a protruding camera does not produce better results. Hence, wp and lp_b describe the platform.
- **T-Carisa**: The parameters are the same as for the C-Carisa design. This design was added during the optimization phase, reasons for chosing this design are found in Section VII.
- **C-IPAFalcon_Encap**: has the same parameters as the C-IPAFalconY. The platform height is left out, because it was minimized in the optimization of the C-IPAFalconY. Similar to the T-Carisa design, this design was added during the optimization phase.

The bounds on the parameters help the differential algorithm to converge to an optimum. A trade-off has to be made while choosing the bounds. When the bounds are too loose, the algorithm evaluates many infeasible designs. Convergence then takes longer or does not occur, because no feasible designs are sampled for many iterations in a row. Yet, when bounds are too strict, the solution space is restricted and an optimum might be missed. When a parameter was on its bound in the optimization results, the optimization was run again with loser bounds. The bounds, however, still had to comply to the design requirements. The bounds for the parameters can be found in the appendix.

Unfeasible combinations of parameters can be filtered out of the solution space with constraints. The workspace is then not calculated for this combination and time to convergence decreases. Constraints were added to limit the size of the platform and remove geometries with obvious singularities in the required workspace. To illustrate, the parameter x_0 of the T_Falcon design has to remain between the x-values of the platform attachment points.

VII. Optimization Results

First, the optimized geometries of the seven designs were calculated. The results led to the new T-Carisa and C-IpaFalcon_Encap designs and their optimization. The insights

TABLE I: The performance index as percentage for all CDPR geometries

Design	Large Platform Workspace (%)		Small Platform Workspace (%)	
	$35p35t$	$0p0t$	$35p35t$	$0p0t$
T-Nishioka	**14.25**	63.63	**6.93**	50.25
T-Falcon	**28.02**	76.23	**5.75**	57.19
T-Lafourcade	**25.30**	69.52	**2.10**	50.65
H-Platform	**15.42**	69.65	**0**	n.a.
C-IPAFalconX	**9.29**	64.06	**0**	n.a.
C-IPAFalconY	**20.16**	68.68	**6.88**	53.46
C-Carisa	**16.45**	65.90	**6.26**	52.28
T-Carisa	**15.95**	67.71	**5.87**	54.09
C-IpaFalcon_Encap	**28.97**	78.90	**7.78**	63.63

from optimal parameter values of every design influenced the choice of optimization parameters for the next designs. In Section VI, however, the final parameters are presented. The designs were additionally optimized for a maximum platform width, height and length of 1 m. In the last paragraphs, the results are discussed.

A. *Optimization of large platform designs*

The optimized designs are shown in Figure 11. For all designs, a minimum distance of 2 cm between distal endpoints and a distance of 10 cm between proximal endpoints was left to provide space for cable attachment and winches. A final overview of the design performances for both ±35° panning and tilting ($35p35t$) and no panning and tilting ($0p0t$) is presented in table I. The results without panning and tilting serve as extra information about the designs, but this was neither a design goal nor an optimization goal.

Of the first seven designs, the T-Falcon design has the highest performance. Since its attachment points are fully enclosing the statue, the idea of a more symmetric enclosing design arose. Hence, C-IPAFalcon_Encap (Figure 11i), based on the Ipanema-Falcon design, was created. The other new design, T-Carisa (Figure 11h), was created in an attempt to improve the C-Carisa design (Figure 11g). It has the distal endpoints arranged in a T-shape to mimic the successful T-Falcon. The best performing design is the statue enclosing C-IpaFalcon_Encap. Enclosure of the statue, however, is sometimes not feasible. When omitting enclosing designs, the C-Carisa has the best performance. These two CDPR geometries differ in size; the proximal endpoints C-IpaFalcon_Encap cover a vertical rectangle of 2.4 m wide while C-Carisa's endpoints cover a vertical rectangle of 5.3 m wide. Although the difference in result for the T-Falcon and C-IpaFalcon_Encap is not large, the frame of T-Falcon is 4.1 m wide. A wider frame occupies more space, which is not always available. The frame and platform coordinates of the optimized geometries of C-IpaFalcon_Encap and C-Carisa are presented in Table II. The other coordinates can be found in the Appendix.

As seen in Figure 12, the workspace of C-IpaFalcon_Encap is not a radial slice of the required workspace. This occurs while the platform cannot approach the statue without platform-statue collision. Furthermore, its workspace is limited in the radial direction by both the WFW. Other total workspace depictions are found in the appendix.

During the optimizations, some parameters reached their bounds. However, due to design requirements, the bounds could not be relaxed in many cases. Hence, the parameters were assigned a fixed value for a next optimization of the design, which reduced the computing time. For all designs, the dimension of the platform that was important for panning was maximized by the optimization. For C-Carisa, IPAFalconX and IPAFalconY the platform height reached its lower bound. Hence, it was not used for similar designs during later optimizations. For all platforms except T-Nishioka, the height of the frame was maximized and later removed from the parameters. For C-Carisa, the width of the platform was minimized up to its bound of 0.207, but this was discovered at a later stage and did not influence parameterization. This bound was determined by the width of the SpecimIQ spectral camera.

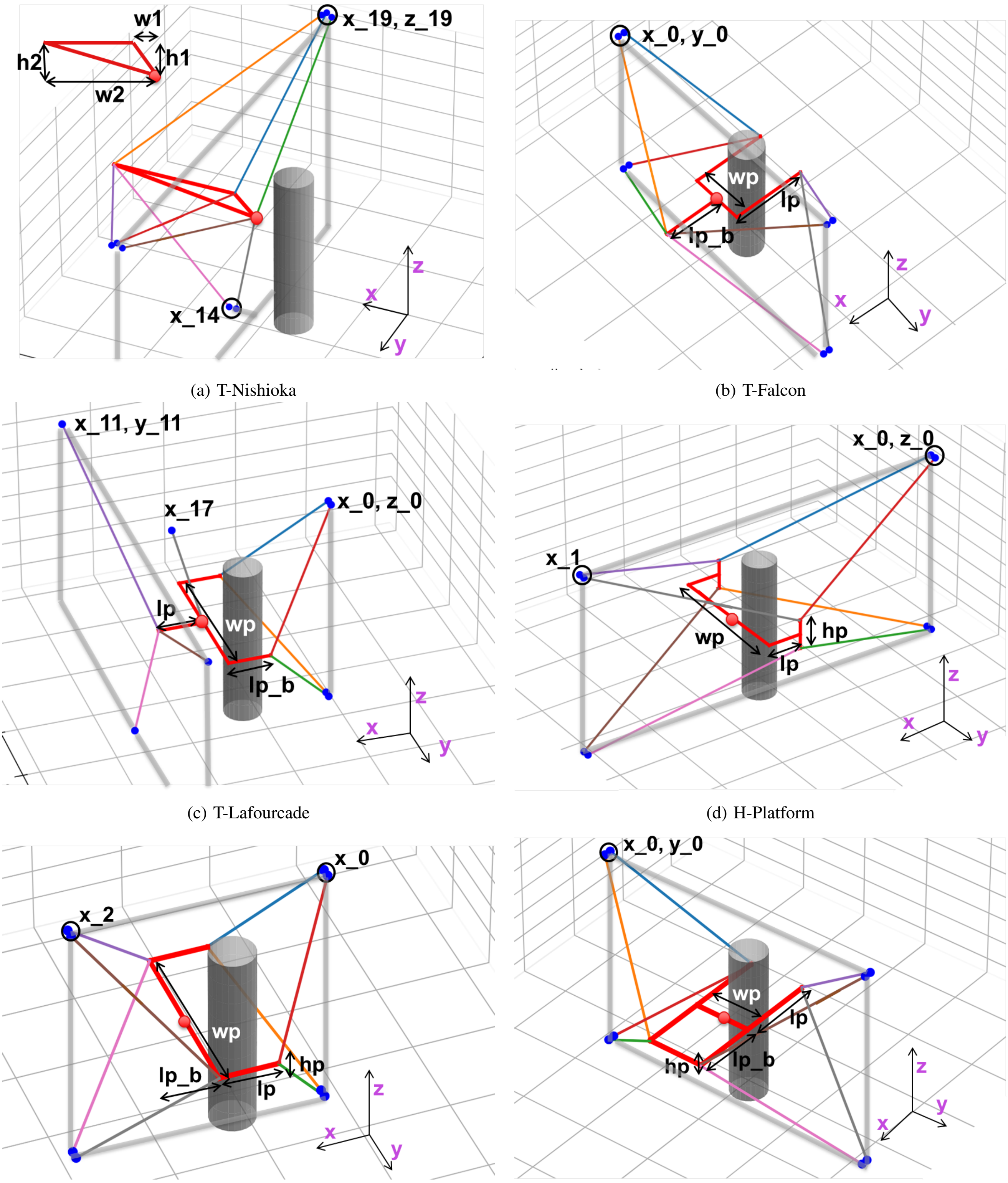
B. *Optimization of small platform designs*

Since one dimension of every platform reached its maximum value, the influence of a more compact platform on the workspace size was tested. Because of the insights of the previous optimization, the maximized platform parameter was taken away and set to 1 m instead of 2 m. The parameters that described this length in couples, e.g. lp and lp_b for T-Lafourcade, were made dependent so they would add up to 2 m. The performance of the small platform designs is compared with the larger platforms in table I. Figures with their geometries can be found in the Appendix.

C. *Discussion of the results*

For the large yaw rotation of the platform, cables that share a distal endpoint and form a triangle with the platform can facilitate large rotations. The longer the platform is, the more the platform can move sideways without singularities. The cable-cable-platform triangle is illustrated in Figure 10. Moreover, the optimization results reveal that the height of the room is fully utilized by the frame. In order to exert vertical forces on the platform, the cables always need to have a vertical component. This can be achieved by placing the endpoints far enough above and below the platform.

The LineSearch algorithm is fast, but does not exhaustively sample the workspace. So when calculating the WFW, the space between sampling points is not evaluated and singularities or impermissible forces can remain undetected. This effect is shown in Figure 13, in which two hulls with different LineSearch evaluations are compared with each other and the benchmark hull. The latter is calculated by evaluating along the vertex directions with small steps of 0.002 m. Because the Line Search algorithm distorts the results, the performance index from the tests and optimization is not fully representative. This effect was the same for all designs, however, so a distinction on performance could still be made. The final results in Figure 12 and Table I are calculated while using the benchmark algorithm for the WFW.



(a) T-Nishioka

(b) T-Falcon

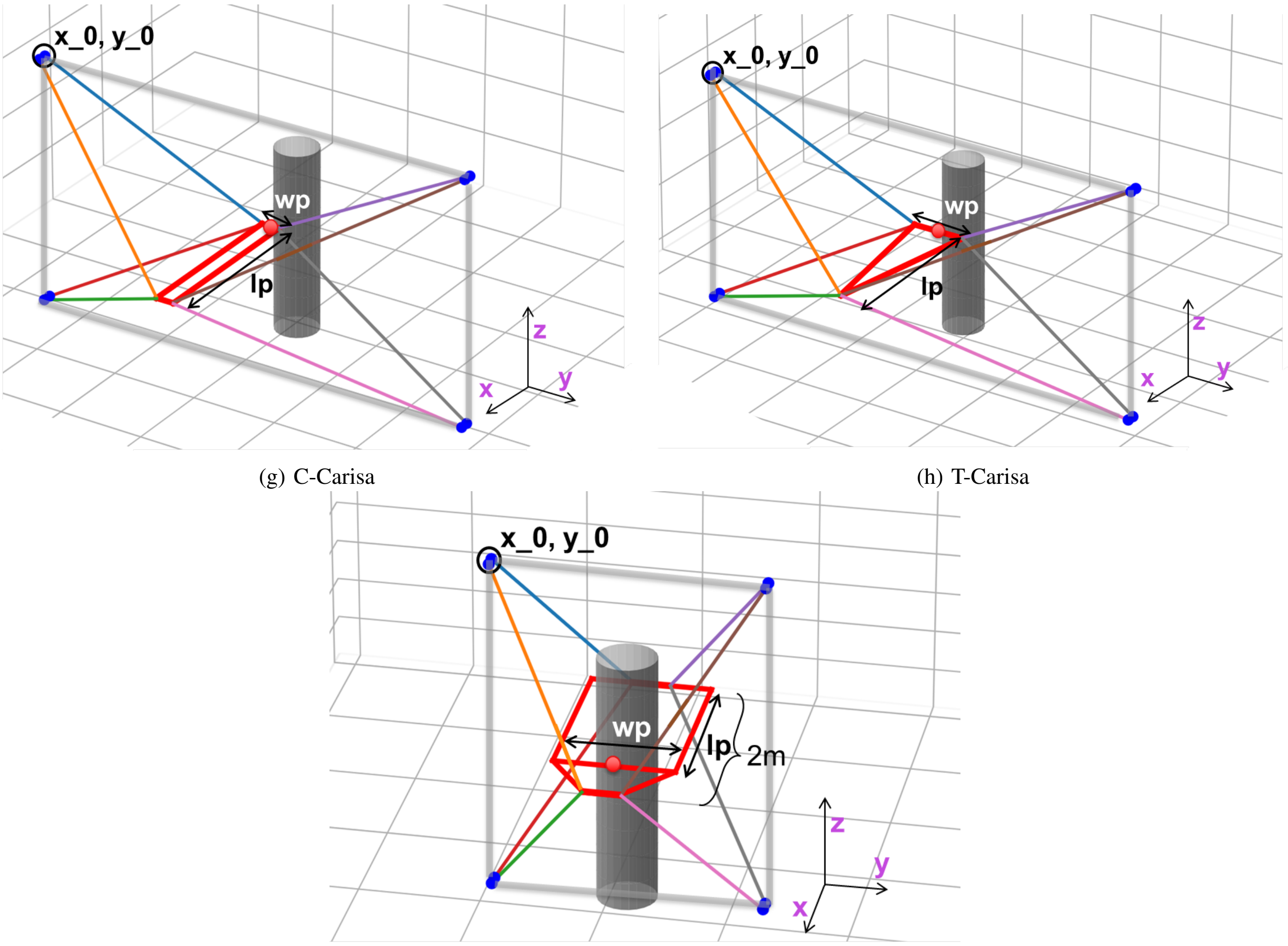
(c) T-Lafourcade

(d) H-Platform

(e) C-IPAFalconX

(f) C-IPAFalconY

Fig. 11: All optimized designs with large platforms and their parameterization. Frame geometry is emphasized with grey lines and the red dot points out the location of the camera.



(g) C-Carisa

(h) T-Carisa

(i) C-IPAFalcon_Encap

Fig. 11: All optimized designs with large platforms and their parameterization. Frame geometry is emphasized with grey lines and the red dot points out the location of the camera.

TABLE II: Frame and platform coordinates of the final CDPR designs

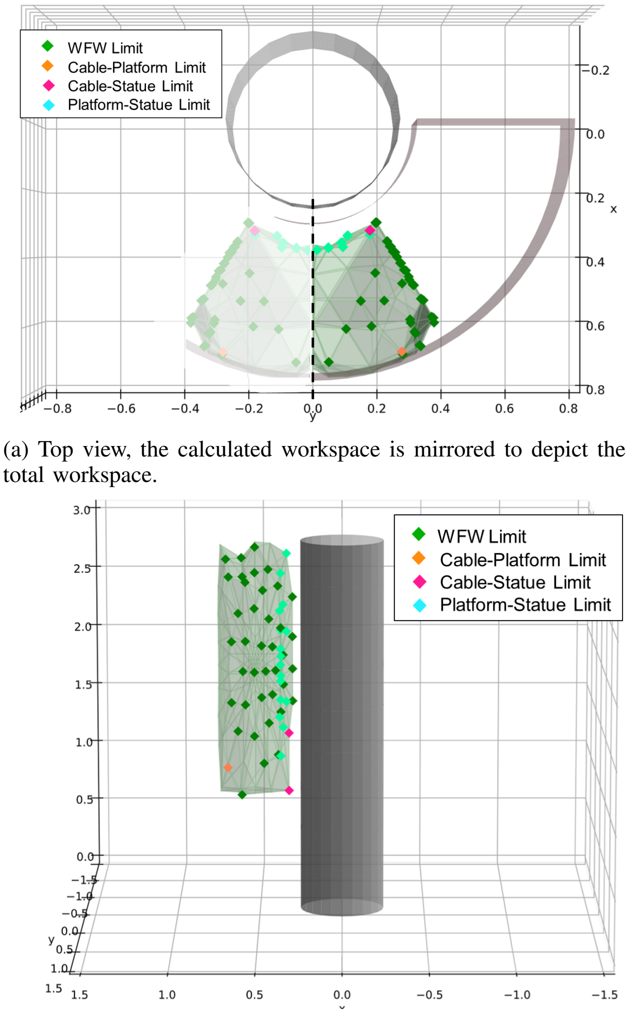
Design		Frame Coordinates (meter)		Platform Coordinates (meter)
C-IPAFalcon	$\mathbf{a_x}$	[-0.089, 0.011, 0.011, -0.089, -0.089, 0.011, 0.011, -0.089]	$\mathbf{b_x}$	[-1.518 0.482 0.482 -1.518 -1.518 0.482 0.482 -1.518]
_Encap	$\mathbf{a_y}$	[-1.127, -1.127, -1.127, -1.127, 1.127 1.127, 1.127, 1.127]	$\mathbf{b_y}$	[-0.161, -0.161, -0.161, -0.161 0.161, 0.161, 0.161, 0.161]
	$\mathbf{a_z}$	[4.000, 4.000, 0.000, 0.000, 4.000, 4.000, 0.000, 0.000]	$\mathbf{b_z}$	[0.010, 0.010, -0.010, -0.010, 0.010, 0.010, -0.010, -0.010]
C-Carisa	$\mathbf{a_x}$	[0.845, 0.945, 0.945, 0.845, 0.845, 0.945, 0.945, 0.845]	$\mathbf{b_x}$	[0.000, 1.999, 1.999, 0.000, 0.000, 1.999, 1.999, 0.000]
	$\mathbf{a_y}$	[-2.638, -2.638, -2.638, -2.638, 2.638, 2.638, 2.638, 2.638]	$\mathbf{b_y}$	[-0.104, -0.104, -0.104, -0.104, 0.104, 0.104, 0.104, 0.104]
	$\mathbf{a_z}$	[4.000, 4.000, 0.000, 0.000, 4.000, 4.000, 0. 000, 0.000]	$\mathbf{b_z}$	[0.010, 0.010, -0.010, -0.010, 0.010, 0.010, -0.010, -0.010]

When calculating the final performance indices, the algorithm did not evaluate for wrench feasibility on the path between a neutral pose (0 in Figure 3) and its tilting and panning poses (1-4 in Figure 3). Chances are, there are missed gaps between these poses. Since the gaps would then still be discovered now and then, but the WFW depicts a regular radial slice, the presence of unfeasible parts between the neutral pose and its panning and tilting poses is not expected.

To produce a higher accuracy, the number of vertices was raised from 42 to 162. The increased detail produces a more representative performance index. For both the optimization and the final results, the hull method's starting positions are the same. When the hull method's starting position would be altered, results would change as well.

VIII. Discussion

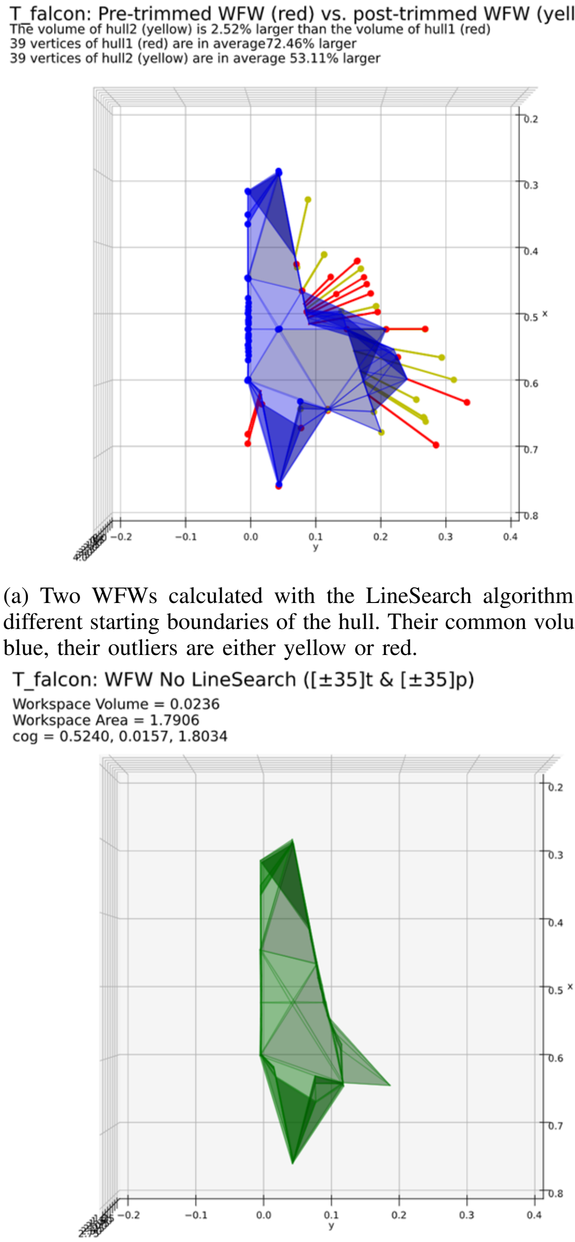
To scan around a statue, operators have to reconfigure the best presented CDPR nine times for ±35° panning and tilting or three times for no panning and tilting. A performance of 100% is not achievable for CDPRs without reconfigurable parts since it implies an impossible yaw rotation of 180° of the platform. The tilting condition adds a different rotation which decreases the WFW and the interference workspace.



(a) Top view, the calculated workspace is mirrored to depict the total workspace.

(b) Side view

Fig. 12: Side view and top view of the total workspace of C-IPAFalcon_Encap


T_falcon: Pre-trimmed WFW (red) vs. post-trimmed WFW (yellow)
The volume of hull2 (yellow) is 2.52% larger than the volume of hull1 (red)
39 vertices of hull1 (red) are in average72.46% larger
39 vertices of hull2 (yellow) are in average 53.11% larger

(a) Two WFWs calculated with the LineSearch algorithm with different starting boundaries of the hull. Their common volume is blue, their outliers are either yellow or red.

T_falcon: WFW No LineSearch ([±35]t & [±35]p)
Workspace Volume = 0.0236
Workspace Area = 1.7906
cog = 0.5240, 0.0157, 1.8034

(b) The benchmark WFW

Fig. 13: LineSearch WFWs and a benchmark WFW for the small-platform T-Falcon.

The required panning, however, is an even more strenuous requirement. It implies a total yaw rotation of 250° for 100% performance. This operation has to be weighed against the difficulty and costs of scanning the statue in a conventional way. The automated solution of the CDPR is beneficial for scanning projects with long scanning times and high labor costs. The achieved workspaces additionally serve as a guideline to the performance of these types of CDPRs for rotating around high 3D objects. The found limits of the designs help to substantiate their applicability to similar problems. For an application, the results present a starting point for a new optimization with varying requirements. Finally, the steps of the design process can act as a source of inspiration for designs in different settings, with different requirements.

For similar applications the architecture of the presented designs will most likely show a similar hierarchy in performance. However, the requirements of this report influenced the performance and the exact geometry of the designs. These will change as soon as requirements on cable force limits, statue, room and platform size, workspace size, panning and tilting or number of cables are altered. The approximation of the statue with a cylinder has made the analysis easier, since irregular statues with protruding parts add interference and require complexly shaped workspaces. The optimization set-up, namely the chosen parameters, their bounds, the constraints and the parameters of the optimization algorithm change the outcomes as well. A trade-off was made between computation time and completeness of the search.

It should be emphasized that the design process in this work is elaborate, but does not cover the entire solution space. Possibly, designs were missed during the brainstorm, honest comparison was not achievable during testing, and false design parameters and algorithm set-up were chosen. However, categorization after brainstorms, comparative testing, parameter optimization and iteration before and during optimization were done. These steps facilitated navigation in an unclear search space with unlimited solutions, in order to discover suitable geometric designs.

As a next step towards building the designs, the stiffness within the workspace and the dynamic workspace for low acceleration will provide insights into feasibility. Furthermore, for a mechanical design, winch placement and the frame and platform structures have to be devised. The model, however, has taken mechanical design into consideration with limited platform shapes, space between endpoints and cable-platform and platform-statue collision detection for the platform's outlines. Research into new, more suitable CDPR designs for 3D object scanning could focus on designs with more cables or layouts with non-colliding crossing cables. For a further increase in performance, the application of alternative CDPR features could be worth looking into. Possibilities are for instance reconfigurability of endpoints, deliberately colliding cables or actuators on the platform.

IX. Conclusion

To rotate around a high 3D object while facing it in every position, different non-reconfigurable, non-suspended CDPR geometry designs were introduced. The CDPR was required to additionally pan and tilt the platform ±35° to register depth. At the same time, requirements were taken into account that represented real-life, such as cable force limits, cable number limits, and frame and platform size. The step-by-step design process of brainstorming, extending, testing, iterating, optimizing and re-iterating was used to converge to suitable designs without using high-dimensional optimization. The intermediate results of the design processes offered insights in important, application-specific geometry features. First of all, endpoints should be combined to avoid cable-cable interference. Furthermore, cable-cable-platform triangles enable large rotations. The dimension of the platform side that belongs to such a triangle should be large for large rotations. Decreasing this platform dimension from 2 m to 1 m is decreasing the performance by more than 50%. For the first time for this application, suitable designs and their performance were introduced: the object-enclosing design C-IPAFalcon_Encap covered 29% of the total 180° workspace and the non-enclosing design C-Carisa covering 16%. Without the required panning and tilting, the performance would be 79% and 66% respectively. The designs were partially optimized to maximize rotation. For 3D CDPRs, this has not been done before. Wrench feasibility, cable-statue, and platform-statue interference limit the workspace of the two designs most significantly. The CDPR in operation covers a 2D upright rectangle of 2.4 m wide for C-IPAFalcon_Encap and 5.3 m for C-Carisa. The C-Carisa CDPR, however, moves on only one side of the statue. Hence, it interferes less with possible protruding parts of a statue. When scanning with panning and tilting, the best performing design has to be repositioned 9 times to scan 360° around the statue.

Acknowledgements

Without Philipp Tempel and Volkert van der Wijk from TU Delft, this work would not have been possible. Thank you for your guidance and advice.

References

[1] Marc Levoy, Szymon Rusinkiewicz, Matt Ginzton, Jeremy Ginsberg, Kari Pulli, David Koller, Sean Anderson, Jonathan Shade, Brian Curless, Lucas Pereira, James Davis, and Duane Fulk. The Digital Michelangelo Project : 3D Scanning of Large Statues. In *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics.*, pages 131–144, 2000.

[2] Lica Pezzati and Raffaella Fontana. 3D Scanning of Artworks. In *Handbook on the Use of Laser in Conservation and Conservation Science*. 2008.

[3] Steven Paquette. 3D scanning in apparel design and human engineering. *IEEE Computer Graphics and Applications*, 16(5):11–15, 1996.

[4] Federico Ponchio and Guido Ranzuglia. 3D Models for Cultural Heritage : Beyond Plain Visualization. *Computer*, 44(7):48 – 55, 2011.

[5] Maria Tsakiri, Charambalos Ioannidis, and Alistair Carty. Laser Scanning Issues for the Geometrical Recording of a Complex Statue. In *Proceedings of Optical 3D Measurement Techniques*, pages 1–8, 2003.

[6] M. Callieri, A. Fasano, G. Impoco, P. Cignoni, R. Scopigno, G. Parrini, and G. Biagini. RoboScan: An automatic system for accurate and unattended 3D scanning. In *Proceedings - 2nd International Symposium on 3D Data Processing, Visualization, and Transmission. 3DPVT 2004*, pages 805–812. IEEE, 2004.

[7] Raffaella Fontana, Marinella Greco, Marzia Materazzi, and Enrico Pampaloni. Three-dimensional modelling of statues : the Minerva of Arezzo. *Journal of Cultural Heritage*, 3(4):325–331, 2002.

[8] P Tempel, M Alfeld, and Volkert van der Wijk. Design and Analysis of Cable-Driven Parallel Robot CaRISA: A Cable Robot for Inspecting and Scanning Artwork Chapter. *ROMANSY 23 - Robot Design, Dynamics and Control*, (September):419–426, 2020.

[9] Andreas Pott. *Cable-driven parallel robots: Theory and application*, volume 120. 2018.

[10] R. R. Thompson and Michael S. Blackstone. Three-dimensional moving camera assembly with an informational cover housing, 2005.

[11] Dinh Quan Nguyen, Marc Gouttefarde, Olivier Company, and Francois Pierrot. On The analysis of large-dimension reconfigurable suspended cable-driven parallel robots. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 5728–5735, 2014.

[12] Marc Gouttefarde and Stéphane Caro. Design of Reconfigurable Cable-Driven Parallel Robots Design of Reconfigurable Cable-Driven Parallel Robots Lorenzo Gagliardini , Marc Gouttefarde , Stéphane Caro HAL Id : hal-01758141. In E Ottaviano, A Pelliccio, and V Gattulli, editors, *Mechatronics for Cultural Heritage and Civil Engineering*, number January, pages 85–113. 2018.

[13] Richard Verhoeven. Analysis of the Workspace of Tendon-Based. *Duisburg Gerhard Mercator University*, 2003.

[14] Dinh Quan Nguyen, Marc Gouttefarde, Olivier Company, and Francois Pierrot. On The analysis of large-dimension reconfigurable suspended cable-driven parallel robots. *Proceedings - IEEE International Conference on Robotics and Automation*, (November 2018):5728–5735, 2014.

[15] Luca Barbazza, Damiano Zanotto, Giulio Rosati, and Sunil K. Agrawal. Design and Optimal Control of an Underactuated Cable-Driven Micro-Macro Robot. *IEEE Robotics and Automation Letters*, 2(2):896–903, 2017.

[16] Marc Gouttefarde, Jean-françois Collard, Nicolas Riehl, Cédric Baradat, Marc Gouttefarde, Jean-françois Collard, Nicolas Riehl, Cédric Baradat, and Geometry Selection. Geometry Selection of a Redundantly Actuated Cable-Suspended Parallel Robot. *IEEE Transactions on Robotics*, 31(2):501–510, 2015.

[17] Mitchell Rushton and Amir Khajepour. Planar Variable Structure Cable-Driven Parallel Robots for Circumventing Obstacles. *Journal of Mechanisms and Robotics*, 13(April):1–24, 2020.

[18] Jeong Hyeon Bak, Sung Wook Hwang, Jonghyun Yoon, Jong Hyeon Park, and Jong Oh Park. Collision-free path planning of cable-driven parallel robots in cluttered environments. *Intelligent Service Robotics*, 12(3):243–253, 2019.

[19] Bin Zi, Jun Lin, and Sen Qian. Localization, obstacle avoidance planning and control of a cooperative cable parallel robot for multiple mobile cranes. *Robotics and Computer-Integrated Manufacturing*, 34:105–123, 2015.

[20] Clément Gosselin and Samuel Bouchard. A Gravity-Powered Mechanism for Extending the Workspace of a Cable-Driven Parallel Mechanism: Application to the Appearance Modelling of Objects. *International Journal of Automation Technology*, 4(4):372–379, 2010.

[21] Michael Anson, Aliakbar Alamdari, and Venkat Krovi. Orientation Workspace and Stiffness Optimization of Cable-Driven Parallel Manipulators With Base Mobility. *Journal of Mechanisms and Robotics*, 9(3), 2017.

[22] Marco Alexander Carpio Alemán, Roque Saltaren, Alejandro Rodriguez, Gerardo Portilla, and Juan Diego Placencia. Rotational workspace expansion of a planar CDPR with a circular end-effector mechanism allowing passive reconfiguration. *Robotics*, 8(3), 2019.

[23] Martin J.D. Otis, Simon Perreault, Thien Ly Nguyen-Dang, Patrice Lambert, Marc Gouttefarde, Denis Laurendeau, and Clément Gosselin. Determination and management of cable interferences between two 6-DOF foot platforms in a cable-driven locomotion interface. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, 39(3):528–544, 2009.

[24] Yonatan Wischnitzer, Nir Shvalb, and Moshe Shoham. Wire-driven parallel robot: Permitting collisions between wires. *International Journal of Robotics Research*, 27(9):1007–1026, 2008.

[25] Saman Lessanibahri, Philippe Cardou, and Stéphane Caro. A Cable-Driven Parallel Robot With An Embedded Tilt-Roll Wrist. *Journal of Mechanisms and Robotics*, 12(2), 2020.

[26] Philipp Miermeister and Andreas Pott. Design of Cable-Driven Parallel Robots with Multiple Platforms and Endless Rotating Axes. *Mechanisms and Machine Science*, 26(Interdisciplinary Applications of Kinematics):21–29, 2015.

[27] Andreas Pott and Philipp Miermeister. Workspace and Interference Analysis of Cable-Driven Parallel Robots with an Unlimited Rotation Axis. pages 341–350, 2018.

[28] T Reichenbach, P Tempel, A W Verl, and A Pott. On Kinetostatics and Workspace Analysis of Multi-platform Cable-Driven Parallel Robots with Unlimited Rotation. In *Robotics and Mechatronics (ISRM 2019): Proceedings of the 6th IFToMM International Symposium*, number June, pages 79–90, 2020.

[29] Sadao Kawamura, Mizuto Ida, Takahiro Wada, and Jing Long Wu. Development of a virtual sports machine using a wire drive system - a trial of virtual tennis. *IEEE International Conference on Intelligent Robots and Systems*, 1:111–116, 1995.

[30] P Lafourcade and R Verhoeven. Une nouvelle architecture, fortement redondante, pour un manipulateur à câbles au volume de travail étendu. *16ème Congrès Français de Mécanique, Nice, France*, 2003.

[31] Satoshi Tadokoro, Yoshio Murao, Manfred Hiller, Rie Murata, Hideaki Kohkawa, and Toshiyuki Matsushima. A motion base with 6-DOF by parallel cable drive architecture. *IEEE/ASME Transactions on Mechatronics*, 7(2):115–123, 2002.

[32] Simon Perreault and Clément M. Gosselin. Cable-driven parallel mechanisms: Application to a locomotion interface. *Journal of Mechanical Design, Transactions of the ASME*, 130(10):1023011–1023018, 2008.

[33] Hussein Hussein, Joao Cavalcanti Santos, and Marc Gouttefarde. Geometric Optimization of a Large Scale CDPR Operating on a Building Facade. *IEEE International Conference on Intelligent Robots and Systems*, (October):5117–5124, 2018.

[34] surfbored. Statues, greek female. http://www.dunjinni.com/forums/printer_friendly_posts.asp?TID=2953, 09 2005. [Online; accessed September 20, 2020].

[35] Stephen Thompson. No title (venus de milo, louvre museum). https://www.ngv.vic.gov.au/explore/collection/work/10713/, 1860. [Online; accessed September 20, 2020].

[36] Andreas Pott. Forward Kinematics and Workspace Determination of a Wire Robot for Industrial Applications. In J. Lenarčič and Philippe Wenger, editors, *Advances in Robot Kinematics: Analysis and Design*, number December, pages 1–472, Dordrecht, 2008. Springer.

[37] Andreas Pott. An improved force distribution algorithm for over-constrained cable-driven parallel robots. *Mechanisms and Machine Science*, 15(August):139–146, 2014.

[38] Dinh Quan Nguyen and Marc Gouttefarde. On the Improvement of Cable Collision Detection Algorithms. *Mechanisms and Machine Science*, 32(March 2016), 2015.

[39] Lorenzo Gagliardini, Stéphane Caro, Marc Gouttefarde, Philippe Wenger, and Alexis Girin. Optimal design of cable-driven parallel robots for large industrial structures. *Proceedings - IEEE International Conference on Robotics and Automation*, (May):5744–5749, 2014.

[40] Dino-Lite. Am4113t. https://www.dino-lite.eu/index.php/nl/component/k2/item/897-am4113t?dt=1631883224251. [Online; accessed April 03, 2021].

[41] Specim. Specim iq technical specifications. https://www.specim.fi/iq/tech-specs/. [Online; accessed April 03, 2021].

[42] Paul Breedveld, Just L Herder, and Tetsuo Tomiyama. Teaching creativity in mechanical design. In *Diversity and Unity. Proceedings of IASDR2011, the 4th World Conference on Design Research*, pages 1–10, 2011.

[43] Pascal Lafourcade, Michel Llibre, and Claude Reboulet. Design of a Parallel Wire-Driven Manipulator for Wind Tunnels. *Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators, October 3–4, 2002, Quebec City, Quebec, Canada*, pages 187–194, 2002.

[44] S. Kawamura, W. Choe, S. Tanaka, and S. R. Pandian. an Ultrahigh Speed Robot FALCON using Wire Drive System. *IEEE international Conference on Robotics and Automation*, pages 215–220, 1995.

[45] Rainer Storn and Kenneth Price. No Title. *Journal of Global Optimization*, 11(4):341–359, 1997.

4 Discussion

First, a literature review was conducted to find the state-of-the-art of workspace improvement of CDPRs through the optimal design of geometrical parameters. Additionally, it focused on suitable methods for obtaining a 3D workspace with large rotations and a 3D obstacle. A narrative review was conducted that addressed CDPRs without reconfigurable endpoints and rigid links, so-called 'regular' CDPRs. The found literature was summarized and organized. It appeared that research on regular CDPRs moving around tall 3D objects with large rotations appeared to be missing. Focusing on regular CDPRs is laying the foundation on which further research on statue scanning CDPRs can be substantiated. Adding additional features such as reconfigurability to a CDPR might not be needed or cumbersome. Likewise, the literature review only focused on the methods and algorithms that were used for optimal design in previous works, because there their functionality is already proven. The most promising methods were combined to a new one. It can be debated that, in order to reach the most suitable solutions, alternative types of CDPRs should have been researched too. These alterations may provide better results,

With the knowledge acquired during the literature review, the goal of the paper was to find good geometric CDPR designs for rotating around 3D objects with large rotation. First, a model was presented to test the performance of a design. Methods and algorithms from different papers were combined for this model. Not all originated from the literature review because some research did not share all of their methods. Nevertheless, all methods were tested before using them in the model. The workspace calculation model was designed for computational speed in order to run many times during an optimization. Yet, a slower more accurate version was used for the final results in the paper.

Subsequently, requirements for a design were set. The panning and tilting, platform weight, cable force limits and scanning depth were based on CDPRs for scanning 2D artwork from [64] and the frame limits and statue size were chosen to mimic a statue in an exhibition hall. The statue was approximated with a cylinder, which made the interference analysis more straightforward. While the protruding parts of irregular statues cause extra interference and complicate the statue surface, a more complex workspace analysis would be required. All these requirements influenced the performance and the exact geometry of the designs. However, specific requirements are still needed since the geometric design of a CDPR is highly application dependent [52]. Nevertheless, the presented designs will most likely show a similar hierarchy in performance for similar applications, while their architectures, or also called cable layouts, are very suitable.

Thereafter, a brainstorm was conducted based on promising architectures from literature. To make the brainstorm more systematic, findings were categorized and extended afterward. The cable layouts with eight cables were subsequently tested with the model. Their performance highly depended on the location of the attachment points, but finding these locations for all layouts was too time consuming. Hence, a good geometry was iteratively found for a design which was then applied to all similar designs for comparison. The geometry optimization was defined by choosing geometry parameters and their bounds, constraints and the parameters of the optimization algorithm. These choices influenced the outcomes and a trade-off was made between computation time and completeness of the search. The design process in this work was elaborate but still did not cover the entire solution space. Possibly, designs were missed during the brainstorm, the design tests did not allow for honest comparison, and false design parameters and algorithm parameters were chosen. However, categorization after brainstorms, comparative testing, parameter optimization and iteration before and during optimization were done. This facilitated navigation in an unclear search space with unlimited solutions, in order to discover good geometric designs.

As a final result, nine designs with optimized geometries were presented. The design C-IPAFalconY_Encap was the best enclosing design and C_Carisa the best non-enclosing design. For these designs, 29% and 16% of the 180° workspace were achieved respectively, which appears to be rather discouraging. A performance of 100%, however, is not achievable in any case since it implies an impossible yaw rotation of 180° of the platform. The tilting condition adds rotations which further decrease the WFW and the interference workspace. The required panning, however, is an even more strenuous requirement, as it implies a total yaw rotation of 270° for 100% performance. After removing the panning and tilting requirements the workspace coverage of 79% and 66% respectively was more promising. Based on the geometries of the nine optimized designs, insights from literature could be confirmed. First of all, the optimization algorithm diminished some dimensions of the platform so distal endpoints would coincide. Shared endpoints were already advocated by Verhoeven [67]. For the large rotations, cables that share a distal endpoint and form a triangle with the platform proved to facilitate large rotations. These triangles were present in many designs such as the Falcon of Kawamura [33], the Nishioka by Tadokoro [62] and in the work of Lafourcade [35]. Furthermore, the height of the room was fully used for every design and in most cases, a long slender platform proved most beneficial.

To scan around a statue, workers have to reconfigure the best CDPR design from this thesis nine times for panning and tilting of ±35° or three times for no panning and tilting. This operation has to be weighed against the difficulty and costs of scanning the statue in a conventional way. The automatic solution of the CDPR is beneficial for scanning projects for which the camera repositioning is time-consuming. In the case of large statues, the CDPR can replace the automatic gantry, while it is more accurate than serial gantries. Although the frame of the CDPR could be large due to the distance between the cable endpoints, tailor-made solutions can be implemented. The frame of the presented geometric designs is represented by four different points at which two cables each are connected to the frame. Instead of connecting all the four attachment points with a stiff structure, these frame connection points could be covered with two vertical pillars or by attaching pulleys to the building's structure. Any small shifts in the arrangement of connection points can be solved with the recalibration of the robot. When the statue has to be scanned from all sides, however, a circle with a radius of 2.3 to 5.3 meters depending on the design, has to be free of obstructions. Besides, when a statue has protruding parts, the cables of a cable-driven parallel robot are likely to interfere. The enclosing C-IPAFalconY_Encap then has to be reshaped. Moreover, the enclosing platform could collide with the statue when it is brought into position. Scanning with the regular CDPR is challenging for statues because a large obstruction-free space around the statues is required and the statues have complex shapes. Yet, they offer the advantages of modularity, automated scanning, scalability and the precision of parallel robots. Without alterations, freestanding obelisks or pillars can already be scanned with the proposed CDPRs.

As a next step towards building the designs, the stiffness within the workspace and the dynamic workspace for low acceleration will provide more insights into feasibility. Furthermore, for a mechanical design, winch placement and the frame and platform structures have to be devised. The model, however, has taken mechanical design into consideration with limited platform shapes, space between endpoints and cable-platform and platform-statue collision detection for the platform's outlines. Research into new, more suitable CDPR designs for 3D Object scanning could focus on designs with more cables or layouts with non-colliding crossing cables. Moreover, the application of alternative CDPR features could enhance performance. Possibilities are reconfiguring endpoints, deliberately letting cables collide or adding actuators to the platform.

5 Conclusion

The goal of this thesis was to provide a suitable geometrical CDPR design for the scanning of 3D objects and to show its potential. For this purpose, a literature review was conducted first to find the state-of-the-art on workspace improvement of CDPRs through the optimal design of geometrical parameters and to explore which methods can be used to obtain a geometrical CDPR design for scanning around a tall 3D object with large rotations. In the second part of this thesis, the geometric CDPR design for this particular application was obtained.

During the literature review, an overview was made of all works on geometric design of non-suspended, non-reconfigurable CDPRs with workspace properties as design goal. It appeared that no geometrical design of CDPRs for rotating around tall 3D objects had been conducted yet. To achieve such a design, methods and designs from the literature review were combined to produce a research proposal. An iterative design process with a brainstorm based on existing designs, testing, and optimization with a genetic algorithm was proposed. The suggested performance index is the percentage of the aspired workspace that can be covered. This workspace has to be wrench feasible and free from any collision.

In the second part of this work, suitable geometric designs were obtained. The platform of the CDPRs needed to be able to rotate around a statue, as well as to pan and tilt ±35° to register depth. At the same time, requirements were taken into account that represent real-life, such as cable force limits, cable number limits, and frame and platform size. The intermediate results of the design processes offered insights into important, application-specific geometry features. First of all, endpoints should be combined to avoid cable-cable interference. Furthermore, cable-cable-platform triangles enable large rotations. The dimension of the platform side that belongs to such a triangle should be large for large rotations. Decreasing this platform dimension from 2 m to 1 m is decreasing the performance by more than 50%. Subsequently, nine suitable designs and their performance were introduced: the best object-enclosing design covers 29% of the total 180° workspace, and the non-enclosing design covers 16%. Without the required panning and tilting, the performance is 79% and 66% respectively. For the two designs, wrench feasibility, cable-statue, and platform-statue interference limit the workspace. The CDPR proximal endpoints cover a 2D upright rectangle for both designs. The non-enclosing CDPR design additionally operates on only one side of the statue. Hence, it interferes less with possible protruding parts of a statue.

For the first time, this thesis presented suitable geometric CDPR designs for rotating around tall 3D objects. Furthermore, the design process was documented and their performance was quantified. The presented designs serve as a guideline to the performance of CDPRs in the new application of rotating around high 3D objects. The found limits of the designs help to substantiate their applicability in a similar setting. For actual application and a tailor-made solution, the results present a starting point for a new optimization with varying requirements. Finally, the steps of the design process can act as a source of inspiration for CDPRs applied in different settings.

Bibliography

[1] Ghasem Abbasnejad, Jungwon Yoon, and Hosu Lee. Optimum kinematic design of a planar cable-driven parallel robot with wrench-closure gait trajectory. *Mechanism and Machine Theory*, 99:1–18, 2016. ISSN 0094114X. doi: 10.1016/j.mechmachtheory.2015.12.009. URL `http://dx.doi.org/10.1016/j.mechmachtheory.2015.12.009`.

[2] Saeed Abdolshah, Damiano Zanotto, Giulio Rosati, and Sunil K. Agrawal. Optimizing stiffness and dexterity of planar adaptive cable-driven parallel robots. *Journal of Mechanisms and Robotics*, 9(3):1–11, 2017. ISSN 19424310. doi: 10.1115/1.4035681.

[3] Michael Anson, Aliakbar Alamdari, and Venkat Krovi. Orientation Workspace and Stiffness Optimization of Cable-Driven Parallel Manipulators With Base Mobility. *Journal of Mechanisms and Robotics*, 9(3), 2017. ISSN 1942-4302. doi: 10.1115/1.4035988.

[4] Mohammad M. Aref, H. D. Taghirad, and S. Barissi. Optimal Design Of Dexterous Cable Driven Parallel Manipulators. *Int. Journal Of Robotics:Theories and Applications*, 1(1):29–47, 2009.

[5] Marc Arsenault. Optimization of the prestress stable wrench closure workspace of planar parallel three-degree-of-freedom cable-driven mechanisms with four cables. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1182–1187, 2010. ISSN 10504729. doi: 10.1109/ROBOT.2010.5509351.

[6] Arian Bahrami and Mansour Nikkhah-Bahrami. Multi-objective design of spatial cable robots. In *Proceedings of the 2nd IASTED International Conference on Robotics, Robo 2011*, number November, pages 345–352, 2011. ISBN 9780889869035. doi: 10.2316/P.2011.752-063.

[7] Fabio Bettio, Ruggero Pintus, Alberto Jaspe Villanueva, Emilio Merella, Fabio Marton, and Enrico Gobbetti. Mont'e scan: Effective shape and color digitization of cluttered 3d artworks. *Journal on Computing and Cultural Heritage (JOCCH)*, 8(1):1–23, 2015.

[8] Javad Bolboli, Mohammad A. Khosravi, and Farzaneh Abdollahi. Stiffness feasible workspace of cable-driven parallel robots with application to optimal design of a planar cable robot. *Robotics and Autonomous Systems*, 114(April 2019):19–28, 2019. ISSN 09218890. doi: 10.1016/j.robot.2019.01.012. URL `https://doi.org/10.1016/j.robot.2019.01.012`.

[9] T Bruckmann and Andreas Pott. IPAnema: A family of Cable-Robots for Industrial Applications. *Cable-Driven Parallel Robots*, 12(1):119–134, 2013. ISSN 10504729. doi: 10.1007/978-3-642-31988-4. URL `http://link.springer.com/10.1007/978-3-642-31988-4`.

[10] Joshua Bryson. *The Optimal Design of Cable-Driven Robots*. Doctoral thesis, University of Delaware, 2016. URL `http://udspace.udel.edu/handle/19716/17757`.

[11] M. Callieri, A. Fasano, G. Impoco, P. Cignoni, R. Scopigno, G. Parrini, and G. Biagini. RoboScan: An automatic system for accurate and unattended 3D scanning. In *Proceedings - 2nd International Symposium on 3D Data Processing, Visualization, and Transmission. 3DPVT 2004*, pages 805–812. IEEE, 2004. ISBN 0769522238. doi: 10.1109/TDPVT.2004.1335398.

[12] Jade Cheng. gadiagram. URL `http://www.jade-cheng.com/au/coalhmm/optimization/gadiagram.svg`. [Online; accessed September 16, 2020].

[13] Sai Deng, Fengshui Jing, Rongzhang Zheng, Zize Liang, and Guodong Yang. Multi-objective pose optimal distribution method for the feed support system of five-hundred-meter aperture spherical radio telescope. *International Journal of Advanced Robotic Systems*, 15:172988141875669, 02 2018. doi: 10.1177/1729881418756695.

[14] Bob Donnan. No title. URL `https://www.sbnation.com/2018/9/6/17824444/nfl-skycam-thursday-night-football-snf-nbc-cameras-on-wires-yo`. [Online; accessed September 20, 2020].

[15] Abbas Fattah and Sunil K. Agrawal. On the design of cable-suspended planar parallel robots. *Journal of Mechanical Design, Transactions of the ASME*, 127(5):1021–1028, 2005. ISSN 10500472. doi: 10.1115/1.1903001.

[16] Raffaella Fontana, Marinella Greco, Marzia Materazzi, and Enrico Pampaloni. Three-dimensional modelling of statues : the Minerva of Arezzo. *Journal of Cultural Heritage*, 3(4):325–331, 2002.

[17] Lorenzo Gagliardini. *Discrete Reconfigurations of Cable-Driven Parallel Robots*. Doctoral thesis, l'Ecole Centrale de Nantes, September 2016.

[18] Lorenzo Gagliardini, Stéphane Caro, Marc Gouttefarde, Philippe Wenger, and Alexis Girin. Optimal design of cable-driven parallel robots for large industrial structures. In *Proceedings - IEEE International Conference on Robotics and Automation*, number May, pages 5744–5749, 2014. doi: 10.1109/ICRA.2014.6907703.

[19] Lorenzo Gagliardini, Marc Gouttefarde, and Stéphane Caro. Design of Reconfigurable Cable-Driven Parallel Robots Design of Reconfigurable Cable-Driven Parallel Robots. In E Ottaviano, A Pelliccio, and V Gattulli, editors, *Mechatronics for Cultural Heritage and Civil Engineering*, number January, pages 85–113. Springer Cham, 2018. ISBN 9783319686462. doi: 10.1007/978-3-319-68646-2{_}4. URL `https://hal.archives-ouvertes.fr/hal-01758141`.

[20] C. M. Gosselin and J. Angeles. A global performance index for the kinematic optimization of robotic manipulators. *Journal of Mechanical Design, Transactions of the ASME*, 113(3):220–226, 1991. ISSN 10500472. doi: 10.1115/1.2912772.

[21] Clément M. Gosselin. *On the Design of Efficient Parallel Mechanisms*, pages 68–96. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998. ISBN 978-3-662-03729-4. doi: 10.1007/978-3-662-03729-4_4. URL `https://doi.org/10.1007/978-3-662-03729-4_4`.

[22] Clément GOSSELIN. Cable-driven parallel mechanisms: state of the art and perspectives. *Mechanical Engineering Reviews*, 1(1):DSM0004–DSM0004, 2014. ISSN 2187-9753. doi: 10.1299/mer.2014dsm0004.

[23] Clément Gosselin and Samuel Bouchard. A Gravity-Powered Mechanism for Extending the Workspace of a Cable-Driven Parallel Mechanism: Application to the Appearance Modelling of Objects. *International Journal of Automation Technology*, 4(4):372–379, 2010. ISSN 1881-7629. doi: 10.20965/ijat.2010.p0372.

[24] Marc Gouttefarde and Clément M. Gosselin. Analysis of the wrench-closure workspace of planar parallel cable-driven mechanisms. *IEEE Transactions on Robotics*, 22(3):434–445, 2006. ISSN 15523098. doi: 10.1109/TRO.2006.870638.

[25] Marc Gouttefarde, Jean-françois Collard, Nicolas Riehl, Cédric Baradat, Marc Gouttefarde, Jean-françois Collard, Nicolas Riehl, Cédric Baradat, and Geometry Selection. Geometry Selection of a Redundantly Actuated Cable-Suspended Parallel Robot. *IEEE Transactions on Robotics*, 31(2):501–510, 2015.

[26] Mahir Hassan and Amir Khajepour. Analysis of a Large-Workspace Cable-Actuated Manipulator for Warehousing Applications. *Proceedings of the ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, pages 1–9, 2009.

[27] A. M. Hay and J. A. Snyman. Optimization of a planar tendon-driven parallel manipulator for a maximal dextrous workspace. *Engineering Optimization*, 37(3):217–236, 2005. ISSN 0305215X. doi: 10.1080/03052150512331328303.

[28] Huajie Hong, Jabran Ali, and Lei Ren. A review on topological architecture and design methods of cable-driven mechanism. *Advances in Mechanical Engineering*, 10(5):1–14, 2018. ISSN 16878140. doi: 10.1177/1687814018774186.

[29] Hussein Hussein, Joao Cavalcanti Santos, and Marc Gouttefarde. Geometric Optimization of a Large Scale CDPR Operating on a Building Facade. *IEEE International Conference on Intelligent Robots and Systems*, (October):5117–5124, 2018. ISSN 21530866. doi: 10.1109/IROS.2018.8593900.

[30] Ch Ioannidis and Maria Tsakiri. Laser scanning and photogrammetry for the documentation of a large statue-experiences in the combined use. In *Proceedings of CIPA XIX International Symposium*, pages 517–523, 2003.

[31] Xuejun Jin, Jinwoo Jung, Seong Ko, Eunpyo Choi, Jong-Oh Park, and Chang-Sei Kim. Geometric Parameter Calibration for a Cable-Driven Parallel Robot Based on a Single One-Dimensional Laser Distance Sensor Measurement and Experimental Modeling. *Sensors*, 18(7):2392, 2018. ISSN 1424-8220. doi: 10.3390/s18072392. URL `http://10.0.13.62/s18072392https://dx.doi.org/10.3390/s18072392`.

[32] Kaveh Kamali, Ahmed Joubair, and Ilian A. Bonev. Optimizing cable arrangement in cable-driven parallel robots to improve the range of available wrenches. *Proceedings of the ASME Design Engineering Technical Conference*, 5B-2018:1–9, 2018. doi: 10.1115/DETC2018-85905.

[33] S. Kawamura, W. Choe, S. Tanaka, and S. R. Pandian. an Ultrahigh Speed Robot FALCON using Wire Drive System. *IEEE international Conference on Robotics and Automation*, pages 215–220, 1995.

[34] József Kövecses and Saeed Ebrahimi. Parameter Analysis and Normalization for the Dynamics and Design of Multibody Systems. *Journal of Computational and Nonlinear Dynamics*, 4(3), 6 2009. ISSN 1555-1415. doi: 10.1115/1.3124785. URL `https://doi.org/10.1115/1.3124785`.

[35] Pascal Lafourcade, Michel Llibre, and Claude Reboulet. Design of a Parallel Wire-Driven Manipulator for Wind Tunnels. *Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators, October 3–4, 2002, Quebec City, Quebec, Canada*, pages 187–194, 2002.

[36] Houssein Lamine, Med Amine Laribi, Sami Bennour, Lotfi Romdhane, and Said Zeghloul. Design Study of a Cable-based Gait Training Machine. *Journal of Bionic Engineering*, 14(2):232–244, 2017. ISSN 16726529. doi: 10.1016/S1672-6529(16)60394-3.

[37] Med Amine Laribi, Giuseppe Carbone, and Saïd Zeghloul. On the Optimal Design of Cable Driven Parallel Robot with a Prescribed Workspace for Upper Limb Rehabilitation Tasks. *Journal of Bionic Engineering*, 16(3):503–513, 2019. ISSN 25432141. doi: 10.1007/s42235-019-0041-4.

[38] Saman Lessanibahri, Philippe Cardou, and Stéphane Caro. Kinetostatic modeling of a cable-driven parallel robot using a tilt-roll wrist. *Mechanisms and Machine Science*, 74:109–120, 2019. ISSN 22110992. doi: 10.1007/978-3-030-20751-9{_}10.

[39] Marc Levoy, Szymon Rusinkiewicz, Matt Ginzton, Jeremy Ginsberg, Kari Pulli, David Koller, Sean Anderson, Jonathan Shade, Brian Curless, Lucas Pereira, James Davis, and Duane Fulk. The Digital Michelangelo Project : 3D Scanning of Large Statues. In *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics.*, pages 131–144, 2000. ISBN 1581132085.

[40] Yangmin Li and Qingsong Xu. GA-based multi-objective optimal design of a planar 3-DOF cable-driven parallel manipulator. *2006 IEEE International Conference on Robotics and Biomimetics, ROBIO 2006*, pages 1360–1365, 2006. doi: 10.1109/ROBIO.2006.340127.

[41] Melanie Mitchell. *An Introduction to genetic algorithms*. The MIT Press Cambridge, Cambridge, 1999. doi: 10.1007/BF02823145.

[42] Ali Nasr and S. Ali A. Moosavian. Multi-criteria design of 6-DoF fully-constrained cable driven redundant parallel manipulator. *International Conference on Robotics and Mechatronics, ICROM 2015*, pages 1–6, 2015. doi: 10.1109/ICRoM.2015.7367591.

[43] Vahid Nazari and Leila Notash. Workspace of wire-actuated parallel manipulators and variations in design parameters. *Transactions of the Canadian Society for Mechanical Engineering*, 37(2):215–229, 2013. ISSN 03158977. doi: 10.1139/tcsme-2013-0013.

[44] Nicoguaro. Coordinate descent, 2020. URL `https://commons.wikimedia.org/wiki/File:Coordinate_descent.svg`. [Online; accessed September 16, 2020].

[45] Leila Notash and Amir Moradi. Optimum layouts for wire-actuated parallel manipulators considering their stiffness characteristics and wire failure. *Transactions of the Canadian Society for Mechanical Engineering*, 36(2):195–206, 2012. ISSN 03158977. doi: 10.1139/tcsme-2012-0014.

[46] Bo Ouyang and Weiwei Shang. Wrench-feasible workspace based optimization of the fixed and moving platforms for cable-driven parallel manipulators. *Robotics and Computer-Integrated Manufacturing*, 30 (6):629–635, 2014. ISSN 07365845. doi: 10.1016/j.rcim.2014.05.001. URL `http://dx.doi.org/10.1016/j.rcim.2014.05.001`.

[47] Simon Perreault and Clément M. Gosselin. Cable-driven parallel mechanisms: Application to a locomotion interface. *Journal of Mechanical Design, Transactions of the ASME*, 130(10):1023011–1023018, 2008. ISSN 10500472. doi: 10.1115/1.2965607.

[48] Cong Bang Pham and Song Huat Yeo. Workspace analysis and optimal design of cable-driven parallel manipulators. *Interval Analysis: Introduction, Methods and Applications*, page 219, 2004.

[49] Cong Bang Pham, Song Huat Yeo, Guilin Yang, and I. Ming Chen. Workspace analysis of fully restrained cable-driven manipulators. *Robotics and Autonomous Systems*, 57(9):901–912, 2009. ISSN 09218890. doi: 10.1016/j.robot.2009.06.004. URL `http://dx.doi.org/10.1016/j.robot.2009.06.004`.

[50] Joshua K. Pickard, Juan A. Carretero, and Virendrakumar C. Bhavsar. Task-optimized cable-actuated planar parallel manipulator architecture and its concurrent implementation. *Proceedings of the 2014 International Conference on High Performance Computing and Simulation, HPCS 2014*, pages 178–185, 2014. doi: 10.1109/HPCSim.2014.6903684.

[51] Federico Ponchio and Guido Ranzuglia. 3D Models for Cultural Heritage : Beyond Plain Visualization. *Computer*, 44(7):48 – 55, 2011. doi: 10.1109/MC.2011.196.

[52] Andreas Pott. *Cable-driven parallel robots: Theory and application*, volume 120. Springer Cham, 2018. ISBN 9783319761374. doi: 10.1007/978-3-319-76138-1.

[53] Jason Pusey, Abbas Fattah, Sunil Agrawal, and Elena Messina. Design and workspace analysis of a 6-6 cable-suspended parallel robot. *Mechanism and Machine Theory*, 39(7):761–778, 2004. ISSN 0094114X. doi: 10.1016/j.mechmachtheory.2004.02.010.

[54] Sen Qian, Bin Zi, Wei Wei Shang, and Qing Song Xu. A review on cable-driven parallel robots. *Chinese Journal of Mechanical Engineering (English Edition)*, 31(4), 2018. ISSN 21928258. doi: 10.1186/s10033-018-0267-9. URL `https://doi.org/10.1186/s10033-018-0267-9`.

[55] Rijksmuseum. De eerste resultaten van operatie nachtwacht. `https://www.rijksmuseum.nl/nl/de-eerste-resultaten-van-operatie-nachtwacht`, 03 2020. [Online; accessed September 20, 2020].

[56] Andrea Shea. A close-up of the chiseled sculpture, "the thinker" by auguste rodin. `https://www.wbur.org/news/2016/05/16/rodin-thinker-peabody-essex`, May 2016. [Online; accessed September 17, 2021].

[57] Andy P. Siddaway, Alex M. Wood, and Larry V. Hedges. How to Do a Systematic Review: A Best Practice Guide for Conducting and Reporting Narrative Reviews, Meta-Analyses, and Meta-Syntheses. *Annual Review of Psychology*, 70(1):747–770, 2019. ISSN 0066-4308. doi: 10.1146/annurev-psych-010418-102803.

[58] Hiroshi Someya and Masayuki Yamamura. A Genetic Algorithm for Function Optimization. *IEEJ Transactions on Electronics, Information and Systems*, 122(3):363–373, 2002. ISSN 0385-4221. doi: 10.1541/ieejeiss1987.122.3{_}363.

[59] Da Song, Lixun Zhang, and Feng Xue. Configuration Optimization and a Tension Distribution Algorithm for Cable-Driven Parallel Robots. *IEEE Access*, 6:33928–33940, 2018. ISSN 21693536. doi: 10.1109/ACCESS.2018.2841988.

[60] Rainer Storn and Kenneth Price. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11(4):341–359, 1997. ISSN 0925-5001. doi: 10.1023/a:1008202821328. URL http://10.0.3.255/a:1008202821328https://dx.doi.org/10.1023/A:1008202821328.

[61] surfbored. Statues, greek female. http://www.dunjinni.com/forums/printer_friendly_posts.asp?TID=2953, 09 2005. [Online; accessed September 20, 2020].

[62] Satoshi Tadokoro, Yoshio Murao, Manfred Hiller, Rie Murata, Hideaki Kohkawa, and Toshiyuki Matsushima. A motion base with 6-DOF by parallel cable drive architecture. *IEEE/ASME Transactions on Mechatronics*, 7(2):115–123, 2002. ISSN 10834435. doi: 10.1109/TMECH.2002.1011248.

[63] Xiaoqiang Tang, Lewei Tang, Jinsong Wang, and Dengfeng Sun. Workspace quality analysis and application for a completely restrained 3-Dof planar cable-driven parallel manipulator. *Journal of Mechanical Science and Technology*, 27(8):2391–2399, 2013. ISSN 1738494X. doi: 10.1007/s12206-013-0624-7.

[64] P Tempel, M Alfeld, and Volkert van der Wijk. Design and Analysis of Cable-Driven Parallel Robot CaRISA: A Cable Robot for Inspecting and Scanning Artwork Chapter. *ROMANSY 23 - Robot Design, Dynamics and Control*, (September):419–426, 2020. doi: 10.1007/978-3-030-58380-4.

[65] Stephen Thompson. No title (venus de milo, louvre museum). https://www.ngv.vic.gov.au/explore/collection/work/10713/, 1860. [Online; accessed September 20, 2020].

[66] Maria Tsakiri, Charambalos Ioannidis, and Alistair Carty. Laser Scanning Issues for the Geometrical Recording of a Complex Statue. In *Proceedings of Optical 3D Measurement Techniques*, pages 1–8, 2003.

[67] Richard Verhoeven. *Analysis of the Workspace of Tendon-Based Stewart Platforms*. PhD thesis, Universität Duisburg-Essen, 2003. URL http://duepublico.uni-duisburg-essen.de/servlets/DerivateServlet/Derivate-5601/verhoevendiss.pdf.

[68] Guilin Yang, Cong Bang Pham, and Song Huat Yeo. Workspace performance optimization of fully restrained cable-driven parallel manipulators. *IEEE International Conference on Intelligent Robots and Systems*, pages 85–90, 2006. doi: 10.1109/IROS.2006.281747.

[69] Rui Yao, Xiaoqiang Tang, Jinsong Wang, and Peng Huang. Dimensional optimization design of the four-cable-driven parallel manipulator in fast. *IEEE/ASME Transactions on Mechatronics*, 15(6):932–941, 2010. ISSN 10834435. doi: 10.1109/TMECH.2009.2035922.

[70] S. H. Yeo, G. Yang, and W. B. Lim. Design and analysis of cable-driven manipulators with variable stiffness. *Mechanism and Machine Theory*, 69:230–244, 2013. ISSN 0094114X. doi: 10.1016/j.mechmachtheory.2013.06.005. URL http://dx.doi.org/10.1016/j.mechmachtheory.2013.06.005.

Appendices

The appendix consists of the following parts:

1. The final literature selection from the literature study
2. The final designs of the CDPRs: coordinates
3. The final designs of the CDPRs: plots
4. Optimization bounds
5. Optimization constraints
6. Brainstorm tables
7. Methods for testing the interference algorithms
8. Most important code

1: The final literature selection from the literature study

Design and workspace analysis of a 6-6 cable-suspended parallel robot	Pusey, J; Fattah, A; Agrawal, S; Messina, E; Jacoff, A
Workspace analysis and optimal design of Cable-Driven Planar Parallel Manipulators	Pham, CB; Yeo, SH; Yang, GL
On the design of cable-suspended planar parallel robots	Fattah, A; Agrawal, SK
Optimization of a planar tendon-driven parallel manipulator for a maximal dextrous workspace	Hay, AM; Snyman, JA
GA-based multi-objective optimal design of a planar 3-DOF cable-driven parallel manipulator	Li, Yangmin; Xu, Qingsong
Cable-driven parallel mechanisms: Application to a locomotion interface	Perreault, Simon; Gosselin, Clement M.
Best kinematic performance analysis of a 6-6 cable-suspended parallel robot	Hadian, H.; Fattah, A.
Dimensional Optimization Design of the Four-Cable-Driven Parallel Manipulator in FAST	Yao, Rui; Tang, Xiaoqiang; Wang, Jinsong; Huang, Peng
Optimization of the prestress stable wrench closure workspace of planar parallel three-degree-of-freedom cable-driven mechanisms with four cables	Arsenault, Marc
ANALYSIS OF A LARGE-WORKSPACE CABLE-ACTUATED MANIPULATOR FOR WAREHOUSING APPLICATIONS	Hassan, Mahir; Khajepour, Amir
Research on Wire-Driven Parallel Manipulators	Wang, Xuanyao; Cao, Yi
OPTIMUM LAYOUTS FOR WIRE-ACTUATED PARALLEL MANIPULATORS CONSIDERING THEIR STIFFNESS CHARACTERISTICS AND WIRE FAILURE	Notash, Leila; Moradi, Amir
Workspace quality analysis and application for a completely restrained 3-Dof planar cable-driven parallel manipulator	Tang, Xiaoqiang; Tang, Lewei; Wang, Jinsong; Sun, Dengfeng

WORKSPACE OF WIRE-ACTUATED PARALLEL MANIPULATORS AND VARIATIONS IN DESIGN PARAMETERS	Nazari, Vahid; Notash, Leila
Workspace Analysis and Performance Evaluation of a 6/6 Cable-Suspended Parallel Robot	Zhou, Hui; Cao, Yi; Yu, Jinghu; Chen, Guilan; Wang, Qiang; Chen, Haiwei
Wrench-feasible workspace based optimization of the fixed and moving platforms for cable-driven parallel manipulators	Ouyang, Bo; Shang, Weiwei
Optimal Design of Cable-Driven Parallel Robots for Large Industrial Structures	Gagliardini, Lorenzo; Caro, Stephane; Gouttefarde, Marc; Wenger, Philippe; Girin, Alexis
Task-Optimized Cable-Actuated Planar Parallel Manipulator Architecture and its Concurrent Implementation	Pickard, Joshua K.; Carretero, Juan. A.; Bhavsar, Virendrakumar C.
Multi-Criteria Design of 6-DoF Fully-Constrained Cable Driven Redundant Parallel Manipulator	Nasr, Ali; Moosavian, S. Ali A.
Design of a Planar Cable Driven Parallel Robot using the concept of Capacity Margin Index	Singh, Ashish; Sahoo, Chiranjibi; Parhi, Dayal R.
Optimum kinematic design of a planar cable-driven parallel robot with wrench-closure gait trajectory	Abbasnejad, Ghasem; Yoon, Jungwon; Lee, Hosu
Anti-pendulation analysis of parallel wave compensation systems	Hu, Yongpan; Tao, Limin; Lv, Wei
Optimizing Stiffness and Dexterity of Planar Adaptive Cable-Driven Parallel Robots	Abdolshah, Saeed; Zanotto, Damiano; Rosati, Giulio; Agrawal, Sunil K.
Orientation Workspace and Stiffness Optimization of Cable-Driven Parallel Manipulators With Base Mobility	Anson, Michael; Alamdari, Aliakbar; Krovi, Venkat
Design Study of a Cable-based Gait Training Machine	Lamine, Houssein; Laribi, Med Amine; Bennour, Sami; Romdhane, Lotfi; Zeghloul, Said
Dimension Synthesis of Suspended Eight Cables-Driven Parallel Robot for Search-and-Rescue Operation	Nurahmi, Latifah; Pramujati, Bambang; Caro, Stephane; Jeffrey
Configuration Optimization and a Tension Distribution Algorithm for Cable-Driven Parallel Robots	Song, Da; Zhang, Lixun; Xue, Feng

Design of Reconfigurable Cable-Driven Parallel Robots Design of Reconfigurable Cable-Driven Parallel Robots	Gagliardini, Lorenzo; Gouttefarde, Marc; Caro, Stephane
OPTIMIZING CABLE ARRANGEMENT IN CABLE-DRIVEN PARALLEL ROBOTS TO IMPROVE THE RANGE OF AVAILABLE WRENCHES	Kamali, Kaveh; Joubair, Ahmed; Bonev, Dian A.
Geometric Optimization of a Large Scale CDPR Operating on a Building Facade	Hussein, Hussein; Santos, Joao Cavalcanti; Gouttefarde, Marc
Structure Optimization of the Cable Driven Legs Trainer	Lamine, Houssein; Laribi, Med Amine; Bennour, Sami; Romdhane, Lotfi; Zeghloul, Said
Stiffness feasible workspace of cable-driven parallel robots with application to optimal design of a planar cable robot	Bolboli, Javad; Khosravi, Mohammad A.; Abdollahi, Farzaneh
Optimal Design and Force Control of a Nine-Cable-Driven Parallel Mechanism for Lunar Takeoff Simulation	Yi, Wangmin; Zheng, Yu; Wang, Weifang; Tang, Xiaoqiang; Liu, Xinjun; Meng, Fanwei
On the Optimal Design of Cable Driven Parallel Robot with a Prescribed Workspace for Upper Limb Rehabilitation Tasks	Amine, Laribi Med; Giuseppe, Carbone; Said, Zeghloul
A Cable-Driven Parallel Robot With An Embedded Tilt-Roll Wrist	Lessanibahri, Saman; Cardou, Philippe; Caro, Stephane
Optimal Design Of Dexterous Cable Driven Parallel Manipulators	Aref, Mohammad M.
Multi-objective design of spatial cable robots	Bahrami, Arian; Nikkhah-Bahrami, Mansour

2: The coordinates of the final CDPR Designs:

The frame coordinates are in the world frame, the platform coordinates in the local platform frame. All coordinates are in meters. The position in the array represents the cable.

Large

T_Nishioka

Cable	Frame coordinates	Platform coordinates
1	[[0.901 -3. 3.793]	[[0.285 -0.01 0.384]
2	[0.972 -3. 3.723]	[1.963 -0.01 0.381]
3	[0.83 -3. 3.723]	[0. -0.01 0.]
4	[0.901 3. 3.793]	[0.285 0.01 0.384]
5	[0.972 3. 3.723]	[1.963 0.01 0.381]
6	[0.83 3. 3.723]	[0. 0.01 0.]
7	[0.9 0. 0.]	[1.963 0. 0.371]
8	[0.8 0. 0.]]	[0. 0. 0.]]

T_falcon

Cable	Frame coordinates	Platform coordinates
1	[0.345 -2.056 4.]	[-1.125 -0.402 0.01]
2	[0.445 -2.056 4.]	[0.87 -0.01 0.01]
3	[0.445 -2.056 0.]	[0.87 -0.01 -0.01]
4	[0.345 -2.056 0.]	[-1.125 -0.402 -0.01]
5	[0.345 2.056 4.]	[-1.125 0.402 0.01]
6	[0.445 2.056 4.]	[0.87 0.01 0.01]
7	[0.445 2.056 0.]	[0.87 0.01 -0.01]
8	[0.345 2.056 0.]]	[-1.125 0.402 -0.01]

T_Lafourcade

Cable	Frame coordinates	Platform coordinates

1	[-1.156 -0.05 3.996]	[-0.557 -0.995 0.01]
2	[-1.156 -0.05 0.]	[-0.557 -0.995 -0.01]
3	[-1.156 0.05 0.]	[-0.557 0.995 -0.01]
4	[-1.156 0.05 3.996]	[-0.557 0.995 0.01]
5	[1.452 -2.994 3.996]	[0.594 -0.01 0.01]
6	[1.452 2.994 3.996]	[0.594 0.01 0.01]
7	[1.452 0. 0.]	[0.594 0. -0.01]
8	[0.932 0. 4.]	[0. 0. 0.]

H_Platform

Cable	Frame coordinates	Platform coordinates
1	[-2.999 -0.05 3.973]	[-0.523 -0.999 0.316]
2	[-2.999 -0.05 0.027]	[-0.523 -0.999 -0.316]
3	[-2.999 0.05 0.027]	[-0.523 0.999 -0.316]
4	[-2.999 0.05 3.973]	[-0.523 0.999 0.316]
5	[2.916 -0.05 3.973]	[-0.503 -0.999 0.316]
6	[2.916 -0.05 0.027]	[-0.503 -0.999 -0.316]
7	[2.916 0.05 0.027]	[-0.503 0.999 -0.316]
8	[2.916 0.05 3.973]	[-0.503 0.999 0.316]

C-IPAFalconX

Cable	Frame coordinates	Platform coordinates
1	[-1.001 -0.05 4.]	[-0.639 -0.999 0.01]
2	[-1.001 -0.05 0.]	[-0.639 -0.999 -0.01]
3	[-1.001 0.05 0.]	[-0.639 0.999 -0.01]
4	[-1.001 0.05 4.]	[-0.639 0.999 0.01]
5	[1.724 -0.05 4.]	[0.001 -0.999 0.01]
6	[1.724 0.05 4.]	[0.001 0.999 0.01]
7	[1.724 -0.05 0.]	[0.001 -0.999 -0.01]

8	[1.724 0.05 0.]	[0.001 0.999 -0.01]

C-IPAFalconY

Cable	Frame coordinates	Platform coordinates
1	[0.207 -2.027 4.]	[-1.082 -0.401 0.017]
2	[0.307 -2.027 4.]	[0.915 -0.401 0.017]
3	[0.307 -2.027 0.]	[0.915 -0.401 -0.017]
4	[0.207 -2.027 0.]	[-1.082 -0.401 -0.017]
5	[0.207 2.027 4.]	[-1.082 0.401 0.017]
6	[0.307 2.027 4.]	[0.915 0.401 0.017]
7	[0.307 2.027 0.]	[0.915 0.401 -0.017]
8	[0.207 2.027 0.]	[-1.082 0.401 -0.017]

C-Carisa

Cable	Frame coordinates	Platform coordinates
1	[[0.845 -2.638 4.]	[[0. -0.104 0.01]
2	[0.945 -2.638 4.]	[1.999 -0.104 0.01]
3	[0.945 -2.638 0.]	[1.999 -0.104 -0.01]
4	[0.845 -2.638 0.]	[0. -0.104 -0.01]
5	[0.845 2.638 4.]	[0. 0.104 0.01]
6	[0.945 2.638 4.]	[1.999 0.104 0.01]
7	[0.945 2.638 0.]	[1.999 0.104 -0.01]
8	[0.845 2.638 0.]]	[0. 0.104 -0.01]]

T-Carisa

Cable	Frame coordinates	Platform coordinates
1	[0.874 -2.892 4.]	[0. -0.335 0.01]
2	[0.974 -2.892 4.]	[1.999 -0.01 0.01]
3	[0.974 -2.892 0.]	[1.999 -0.01 -0.01]

4	[0.874 -2.892 0.]	[0. -0.335 -0.01]
5	[0.874 2.892 4.]	[0. 0.335 0.01]
6	[0.974 2.892 4.]	[1.999 0.01 0.01]
7	[0.974 2.892 0.]	[1.999 0.01 -0.01]
8	[0.874 2.892 0.]	[0. 0.335 -0.01]

C-IPAFalcon_Encap

Cable	Frame coordinates	Platform coordinates
1	[-0.089 -1.127 4.]	[-1.518 -0.161 0.01]
2	[0.011 -1.127 4.]	[0.482 -0.161 0.01]
3	[0.011 -1.127 0.]	[0.482 -0.161 -0.01]
4	[-0.089 -1.127 0.]	[-1.518 -0.161 -0.01]
5	[-0.089 1.127 4.]	[-1.518 0.161 0.01]
6	[0.011 1.127 4.]	[0.482 0.161 0.01]
7	[0.011 1.127 0.]	[0.482 0.161 -0.01]
8	[-0.089 1.127 0.]	[-1.518 0.161 -0.01]

Small

T_Nishioka

Cable	Frame coordinates	Platform coordinates
1	[1.059 -3. 3.973]	[0.987 -0.01 -0.101]
2	[1.13 -3. 3.902]	[0.986 -0.01 -0.168]
3	[0.988 -3. 3.902]	[0. -0.01 0.]
4	[1.059 3. 3.973]	[0.987 0.01 -0.101]
5	[1.13 3. 3.902]	[0.986 0.01 -0.168]
6	[0.988 3. 3.902]	[0. 0.01 0.]
7	[0.452 0. 0.]	[0.986 0. -0.178]
8	[0.352 0. 0.]	[0. 0. 0.]

T_falcon

Cable	Frame coordinates	Platform coordinates
1	[0.671 -2.208 3.914]	[-0.169 -0.301 0.01]
2	[0.771 -2.208 3.914]	[0.831 -0.01 0.01]
3	[0.771 -2.208 0.086]	[0.831 -0.01 -0.01]
4	[0.671 -2.208 0.086]	[-0.169 -0.301 -0.01]
5	[0.671 2.208 3.914]	[-0.169 0.301 0.01]
6	[0.771 2.208 3.914]	[0.831 0.01 0.01]
7	[0.771 2.208 0.086]	[0.831 0.01 -0.01]
8	[0.671 2.208 0.086]	[-0.169 0.301 -0.01]

T_Lafourcade

Cable	Frame coordinates	Platform coordinates
1	[-0.753 -0.05 4.]	[-0.657 -0.5 0.01]
2	[-0.753 -0.05 0.]	[-0.657 -0.5 -0.01]
3	[-0.753 0.05 0.]	[-0.657 0.5 -0.01]
4	[-0.753 0.05 4.]	[-0.657 0.5 0.01]
5	[1.102 -2.903 4.]	[0.215 -0.01 0.01]
6	[1.102 2.903 4.]	[0.215 0.01 0.01]
7	[1.102 0. 0.]	[0.215 0. -0.01]
8	[0.588 0. 4.]	[0. 0. 0.]

H_Platform
N.A.
C-IPAFalconX
N.A.
C-IPAFalconY

Cable	Frame coordinates	Platform coordinates
1	[0.735 -1.781 4.]	[-0.083 -0.105 0.01]
2	[0.835 -1.781 4.]	[0.917 -0.105 0.01]

3	[0.835 -1.781 0.]	[0.917 -0.105 -0.01]
4	[0.735 -1.781 0.]	[-0.083 -0.105 -0.01]
5	[0.735 1.781 4.]	[-0.083 0.105 0.01]
6	[0.835 1.781 4.]	[0.917 0.105 0.01]
7	[0.835 1.781 0.]	[0.917 0.105 -0.01]
8	[0.735 1.781 0.]	[-0.083 0.105 -0.01]

C-Carisa

Cable	Frame coordinates	Platform coordinates
1	[0.8 -1.681 4.]	[0. -0.104 0.01]
2	[0.9 -1.681 4.]	[1. -0.104 0.01]
3	[0.9 -1.681 0.]	[1. -0.104 -0.01]
4	[0.8 -1.681 0.]	[0. -0.104 -0.01]
5	[0.8 1.681 4.]	[0. 0.104 0.01]
6	[0.9 1.681 4.]	[1. 0.104 0.01]
7	[0.9 1.681 0.]	[1. 0.104 -0.01]
8	[0.8 1.681 0.]	[0. 0.104 -0.01]

T-Carisa

Cable	Frame coordinates	Platform coordinates
1	[0.802 -1.798 3.99]	[0. -0.22 0.01]
2	[0.902 -1.798 3.99]	[1. -0.01 0.01]
3	[0.902 -1.798 0.01]	[1. -0.01 -0.01]
4	[0.802 -1.798 0.01]	[0. -0.22 -0.01]
5	[0.802 1.798 3.99]	[0. 0.22 0.01]
6	[0.902 1.798 3.99]	[1. 0.01 0.01]
7	[0.902 1.798 0.01]	[1. 0.01 -0.01]
8	[0.802 1.798 0.01]	[0. 0.22 -0.01]

C-IPAFalcon_Encap

Cable	Frame coordinates	Platform coordinates
1	[-0.087 -1.637 4.]	[-0.999 -0.145 0.01]
2	[0.013 -1.637 4.]	[0.001 -0.145 0.01]
3	[0.013 -1.637 0.]	[0.001 -0.145 -0.01]
4	[-0.087 -1.637 0.]	[-0.999 -0.145 -0.01]
5	[-0.087 1.637 4.]	[-0.999 0.145 0.01]
6	[0.013 1.637 4.]	[0.001 0.145 0.01]
7	[0.013 1.637 0.]	[0.001 0.145 -0.01]
8	[-0.087 1.637 0.]	[-0.999 0.145 -0.01]

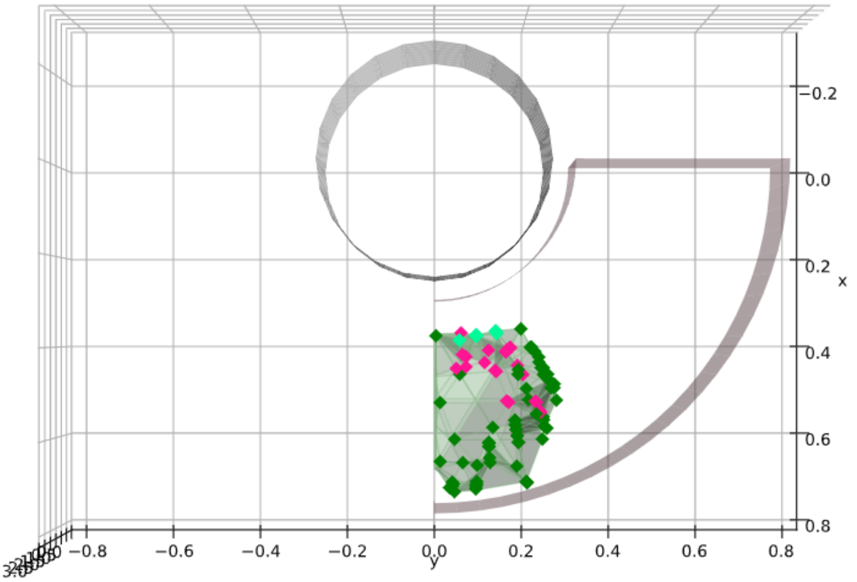
3: The final designs of the CDPRs: plots

Large

T-Nishioka

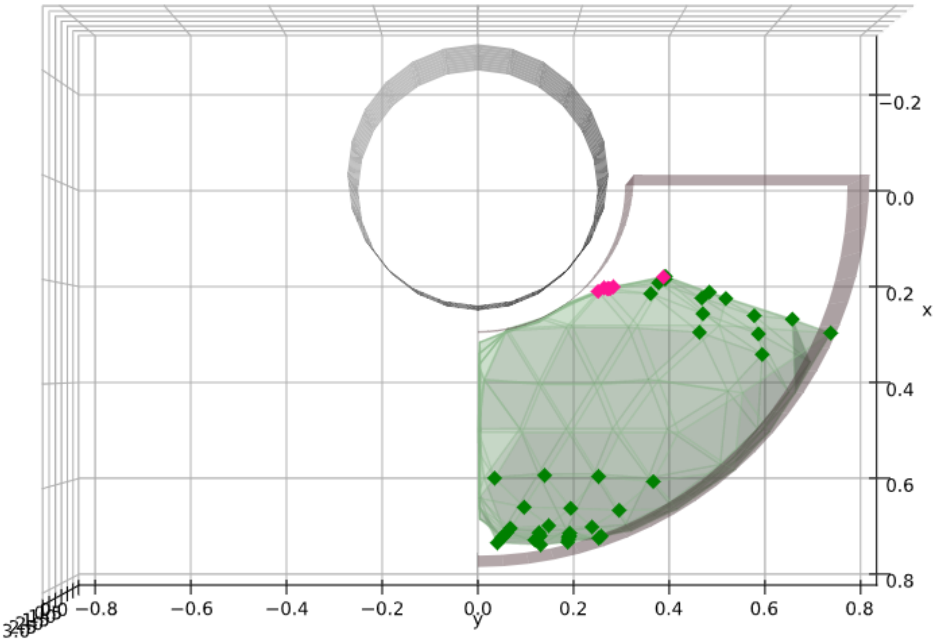
T_8_3_332_Nishioka_Rotated_Vertical3_V2: Total Hull ([±35]t

Workspace Volume = 0.0938



T_8_3_332_Nishioka_Rotated_Vertical3_V2: Total Hull ([±0]t & [±0

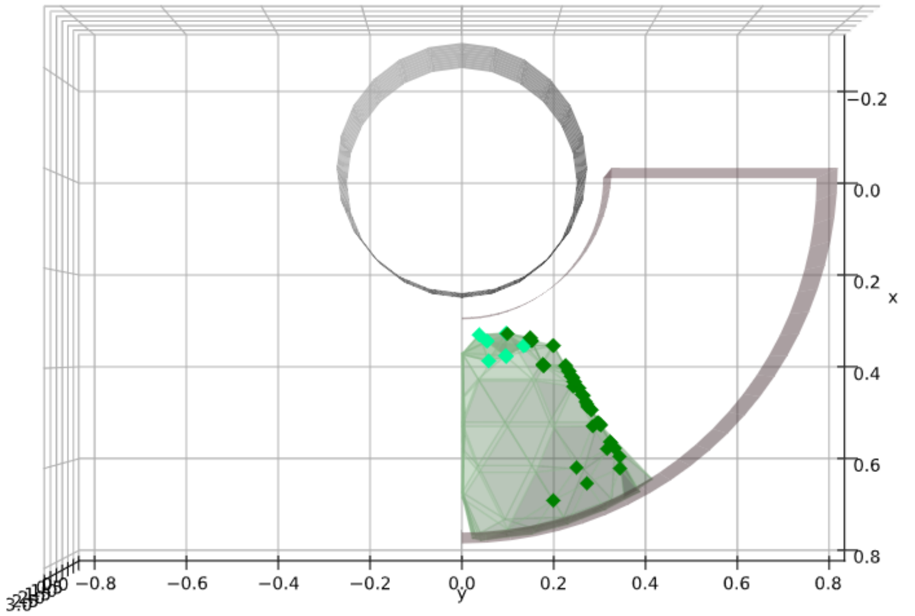
Workspace Volume = 0.4448



T-Falcon

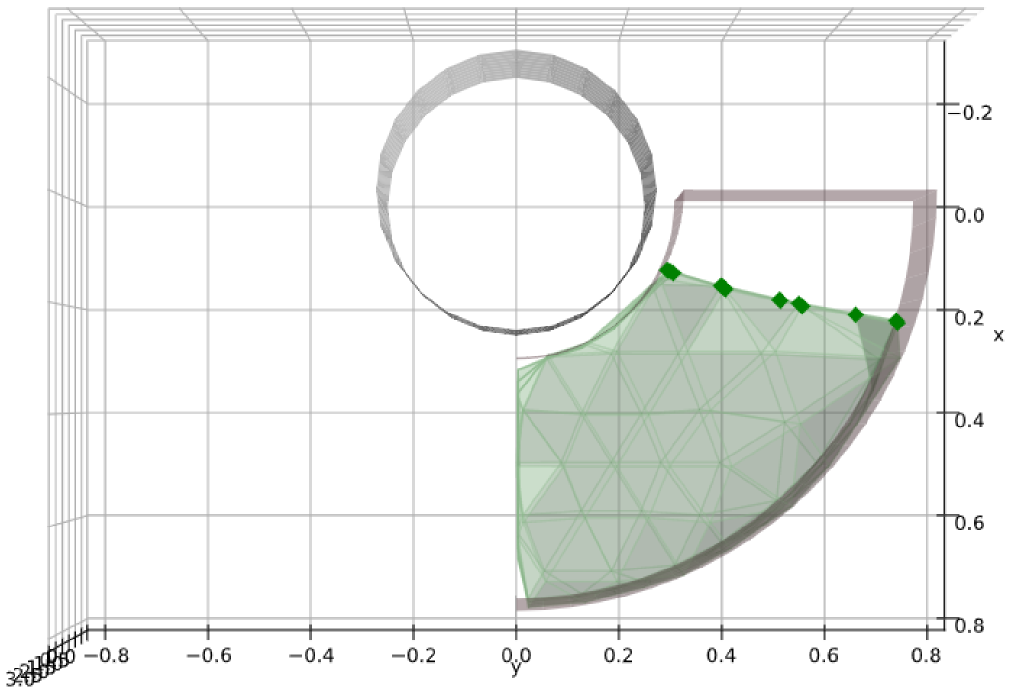
T_falcon: Total Hull ([±35]t & [±35]p) Top View

Workspace Volume = 0.1845

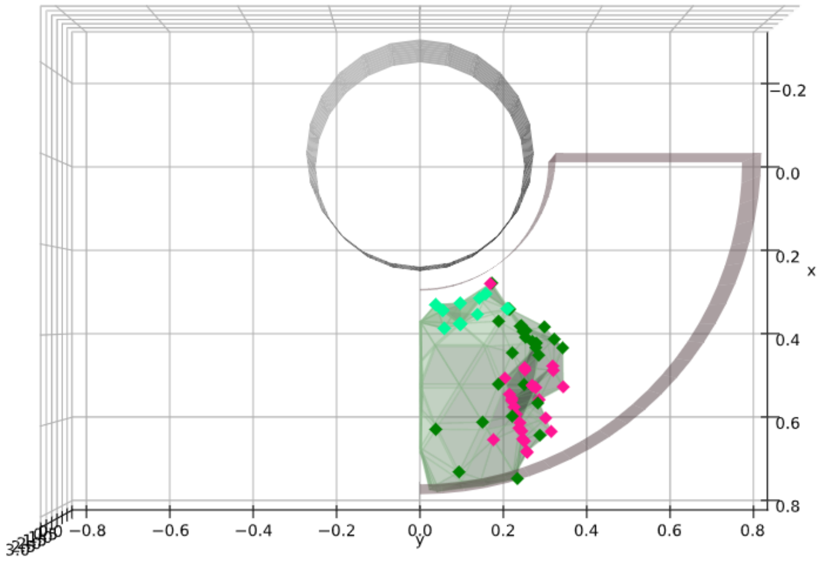


T_falcon: Total Hull ([±0]t & [±0]p) Top View

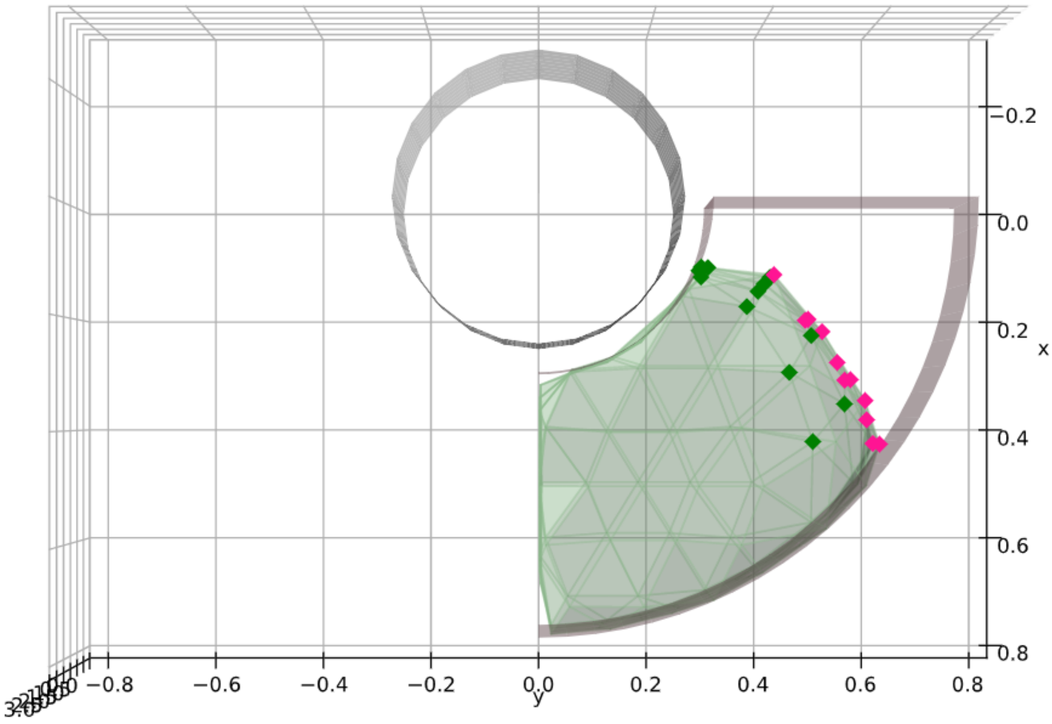
Workspace Volume = 0.5331



T-Lafourcade



T_lafourcade: Total Hull ([±35]t & [±35]p) Top View
Workspace Volume = 0.1665

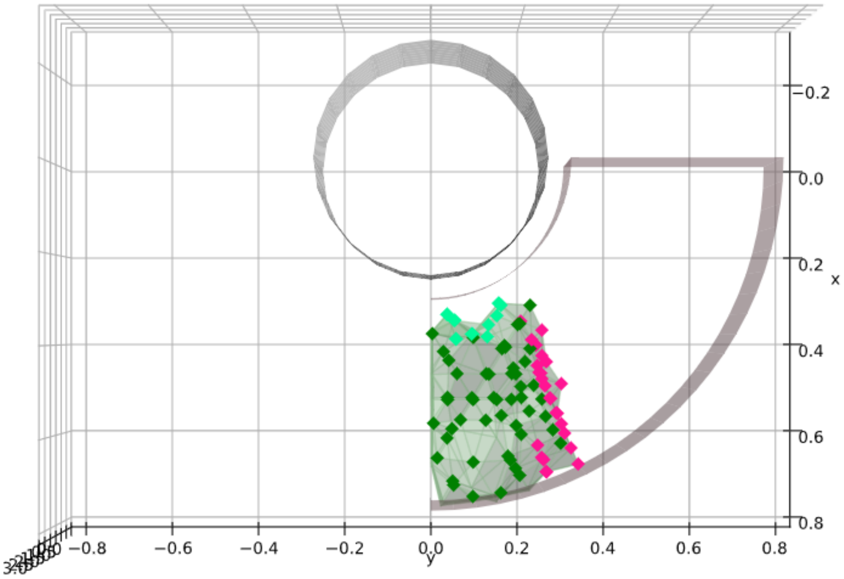


[_lafourcade: Total Hull ([±0]t & [±0]p) Top View
Vorkspace Volume = 0.4861

H-Platform

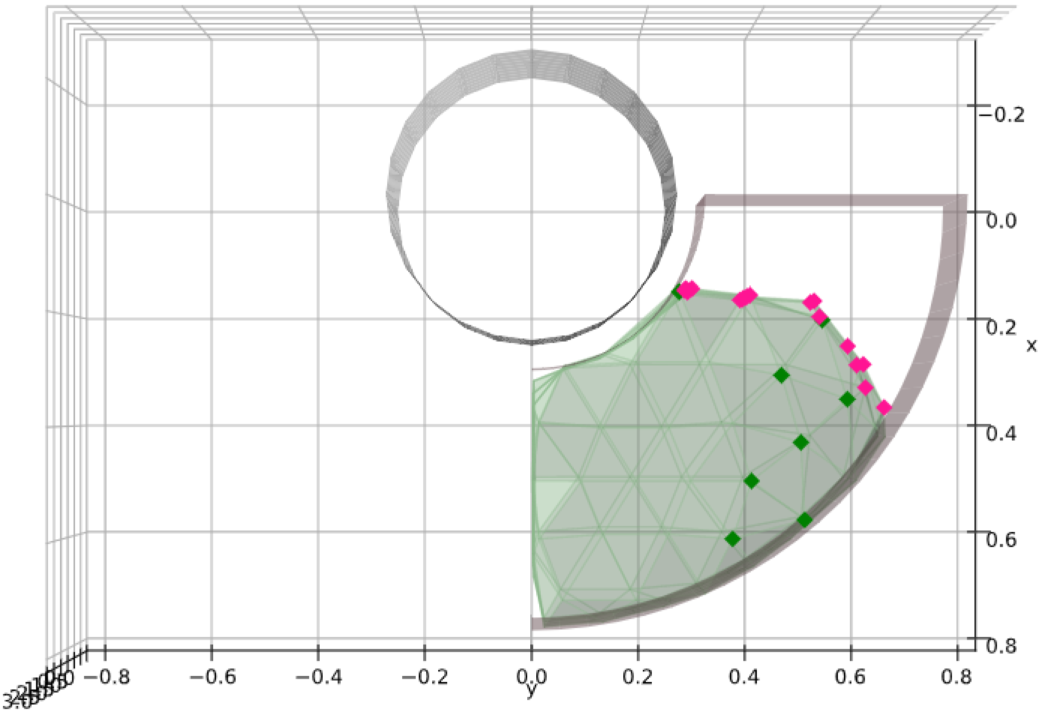
H_platform: Total Hull ([±35]t & [±35]p) Top View

Workspace Volume = 0.1015

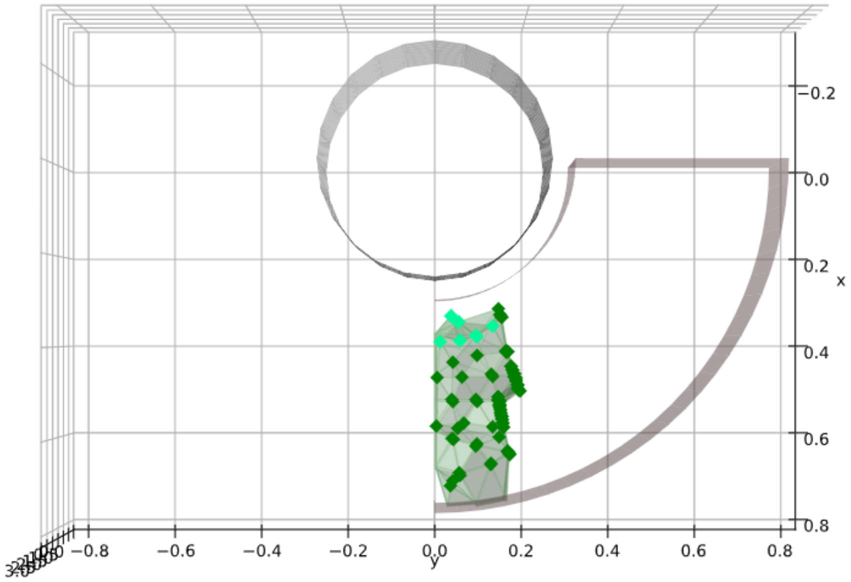


H_platform: Total Hull ([±0]t & [±0]p) Top View

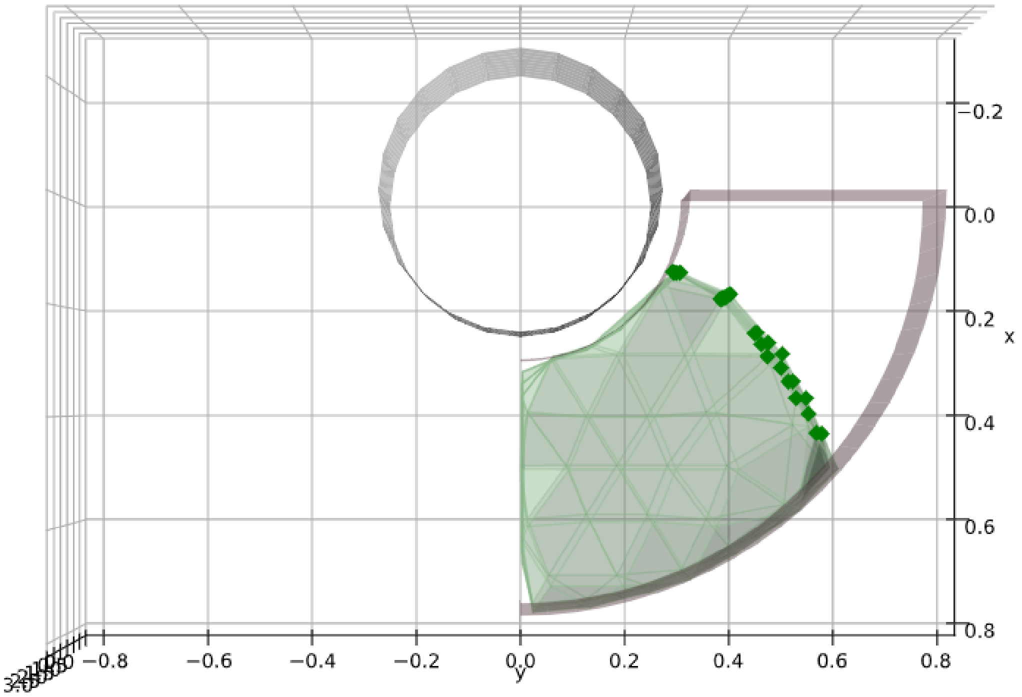
Workspace Volume = 0.4884



C-IPAFalconX



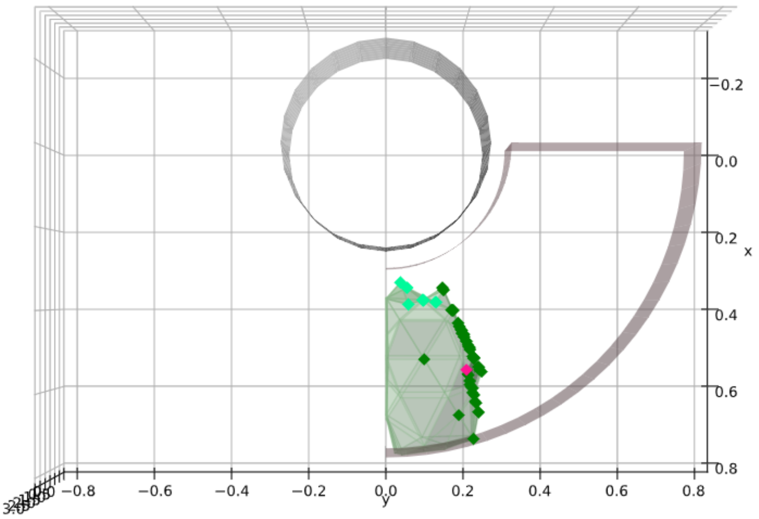
IPAFalconX: Total Hull ([±35]t & [±35]p) Top View
Workspace Volume = 0.0612



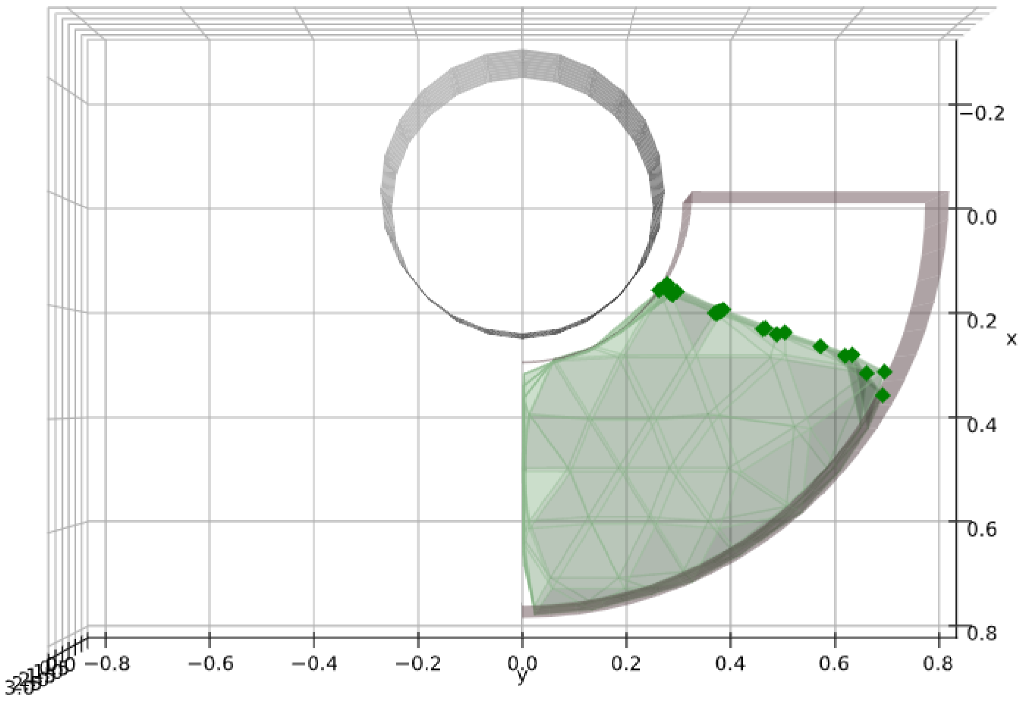
IPAFalconX: Total Hull ([±0]t & [±0]p) Top View
Workspace Volume = 0.4480

C-IPAFalconY

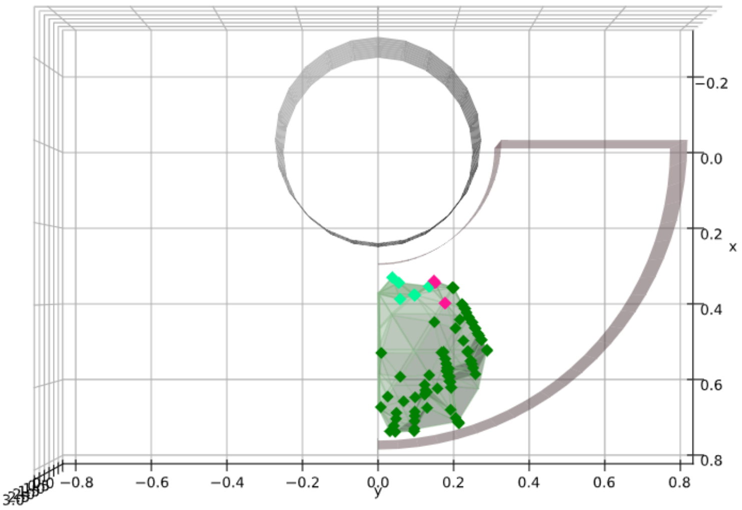
IPAFalconY: Total Hull ([±35]t & [±35]p) Top View
Workspace Volume = 0.1328



IPAFalconY: Total Hull ([±0]t & [±0]p) Top View
Workspace Volume = 0.4803

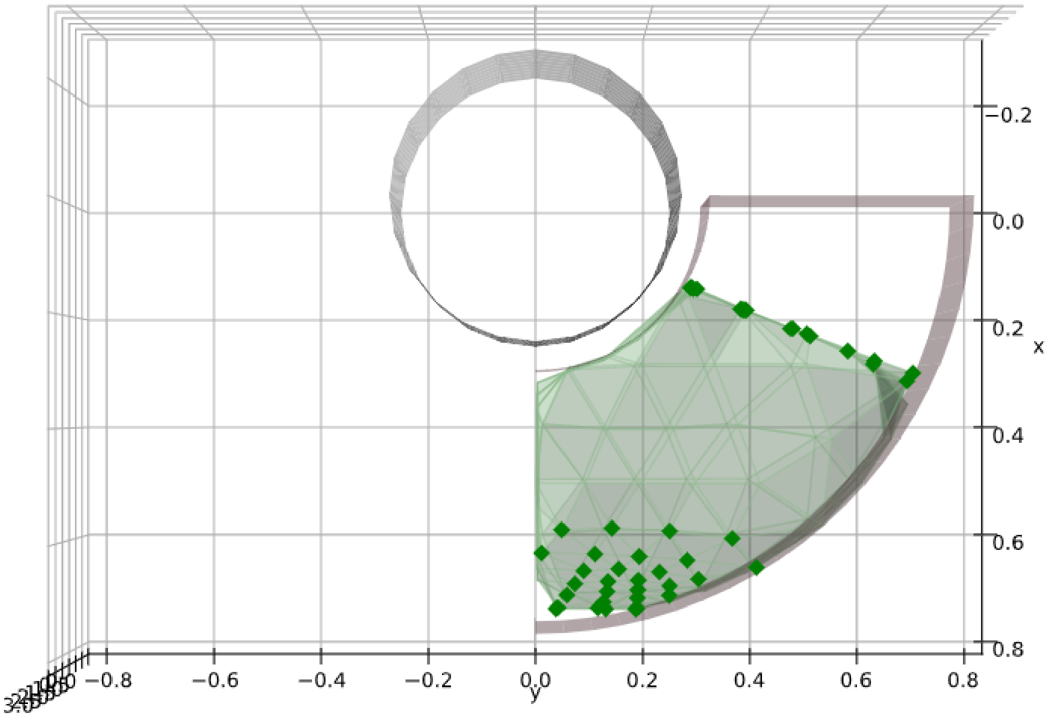


C-Carisa



CarisaLowDim: Total Hull ([±35]t & [±35]p) Top View

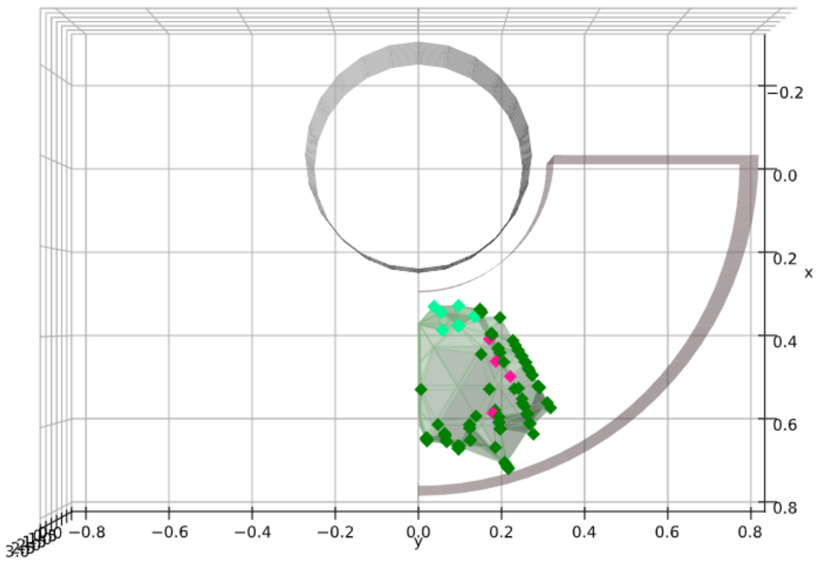
Workspace Volume = 0.1082



CarisaLowDim: Total Hull ([±0]t & [±0]p) Top View

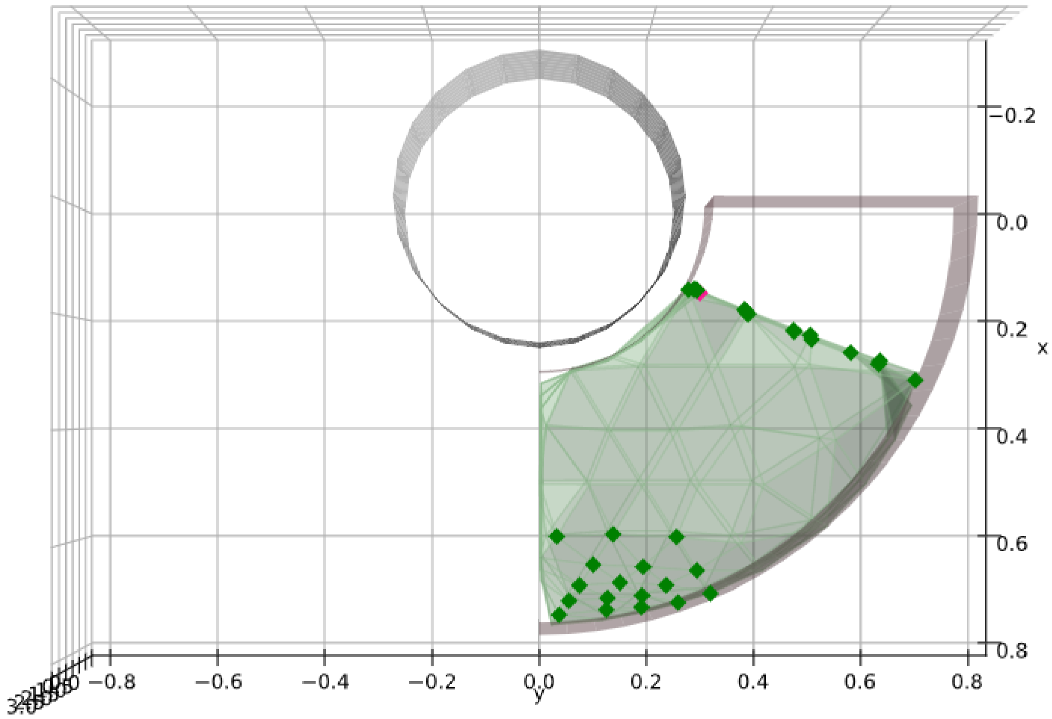
Workspace Volume = 0.4606

T-Carisa



T_Carisa: Total Hull ([±35]t & [±35]p) Top View

Workspace Volume = 0.1049



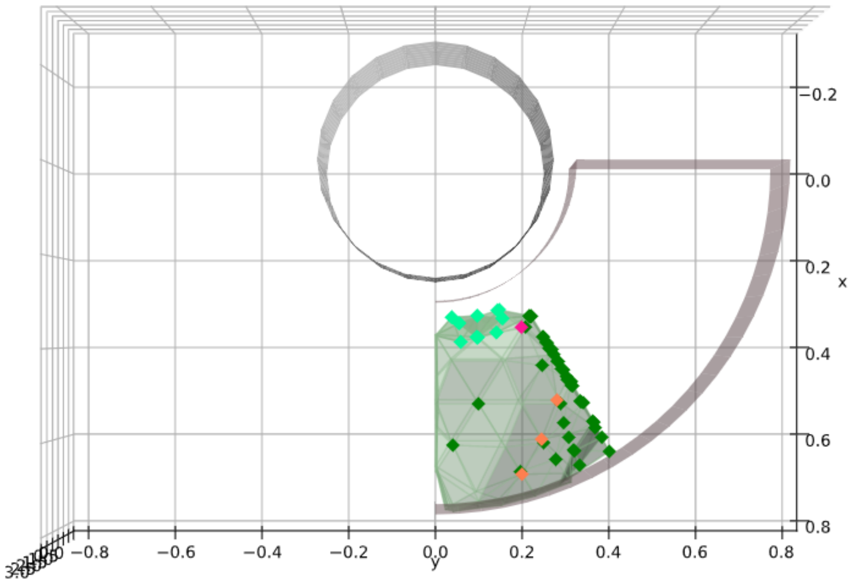
T_Carisa: Total Hull ([±0]t & [±0]p) Top View

Workspace Volume = 0.4734

C-IPAFalcon_Encap

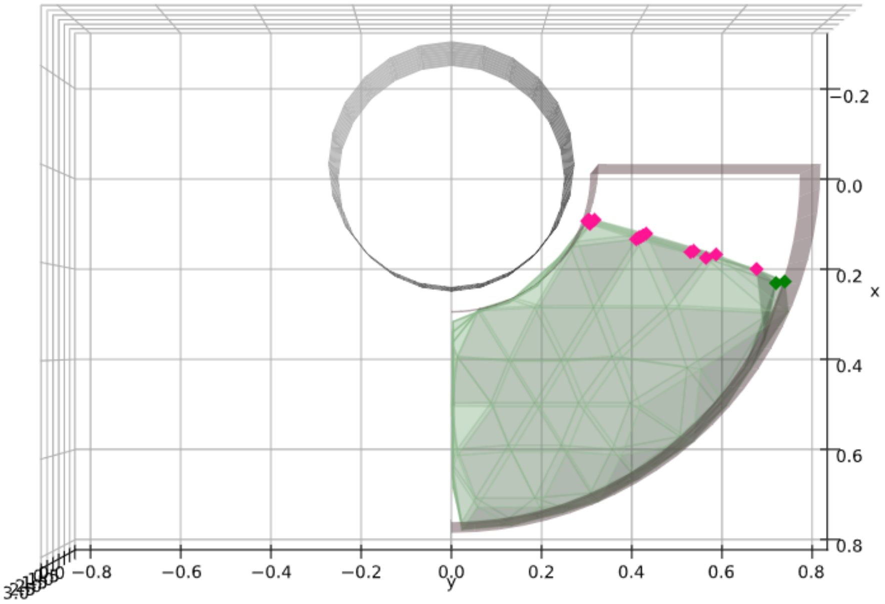
IPAFalcon_Encap: Total Hull ([±35]t & [±35]p) Top View

Workspace Volume = 0.1926



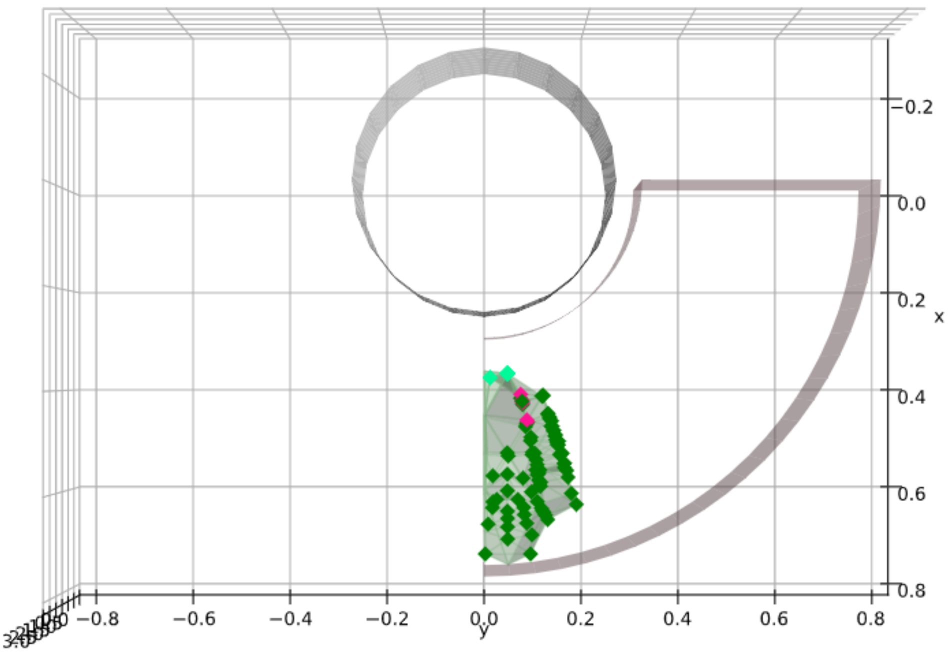
IPAFalcon_Encap: Total Hull ([±0]t & [±0]p) Top View

Workspace Volume = 0.5517



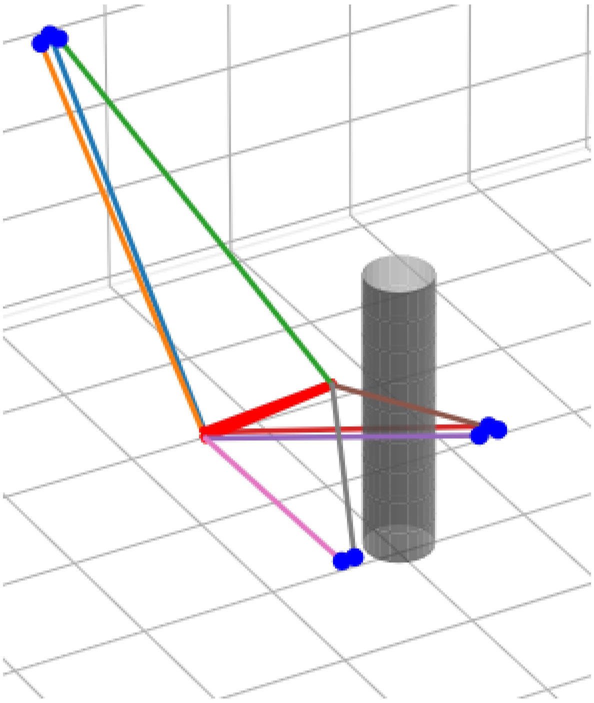
Small

T-Nishioka

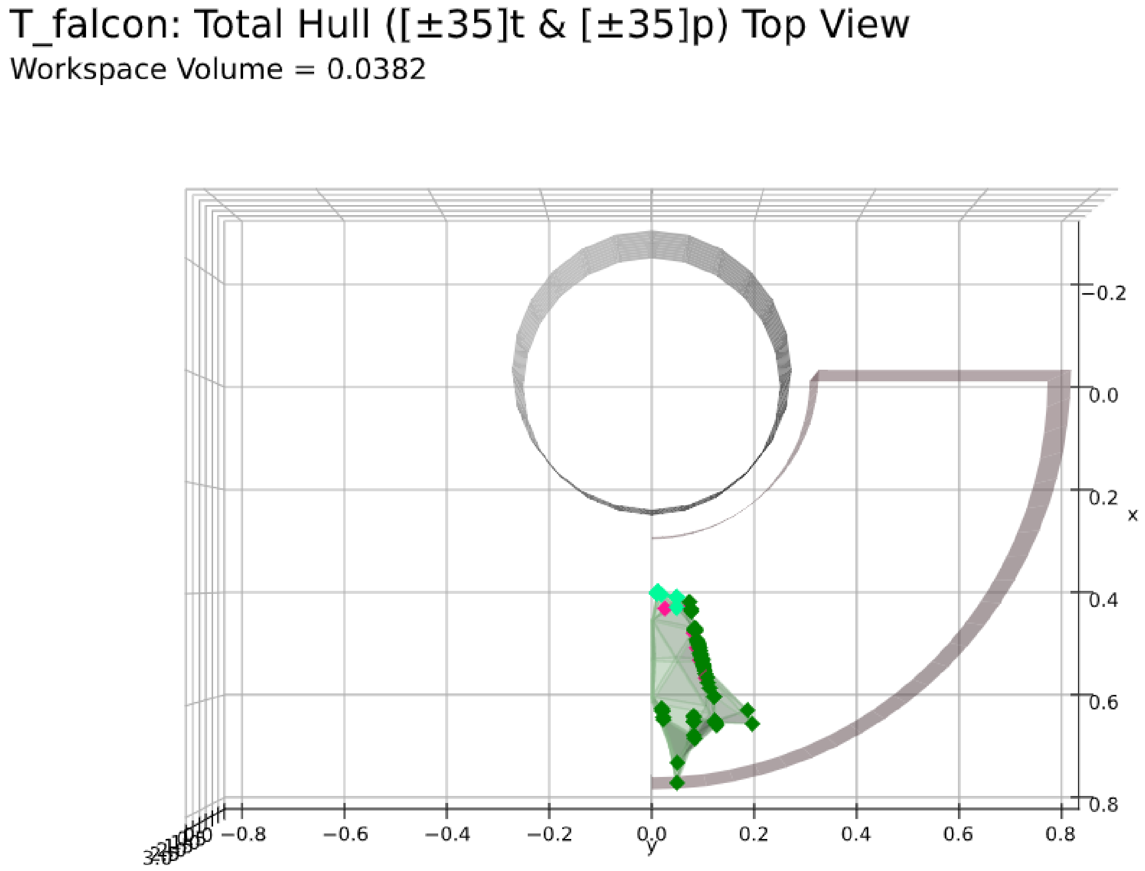


T_8_3_332_Nishioka_Rotated_Vertical3_V2: Total Hull ([±35]t & [±

Workspace Volume = 0.0411

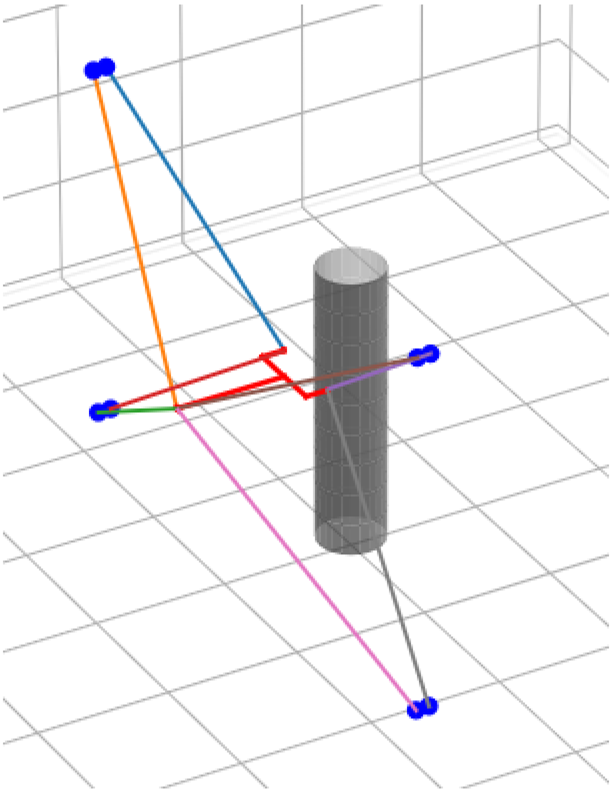


T-Falcon

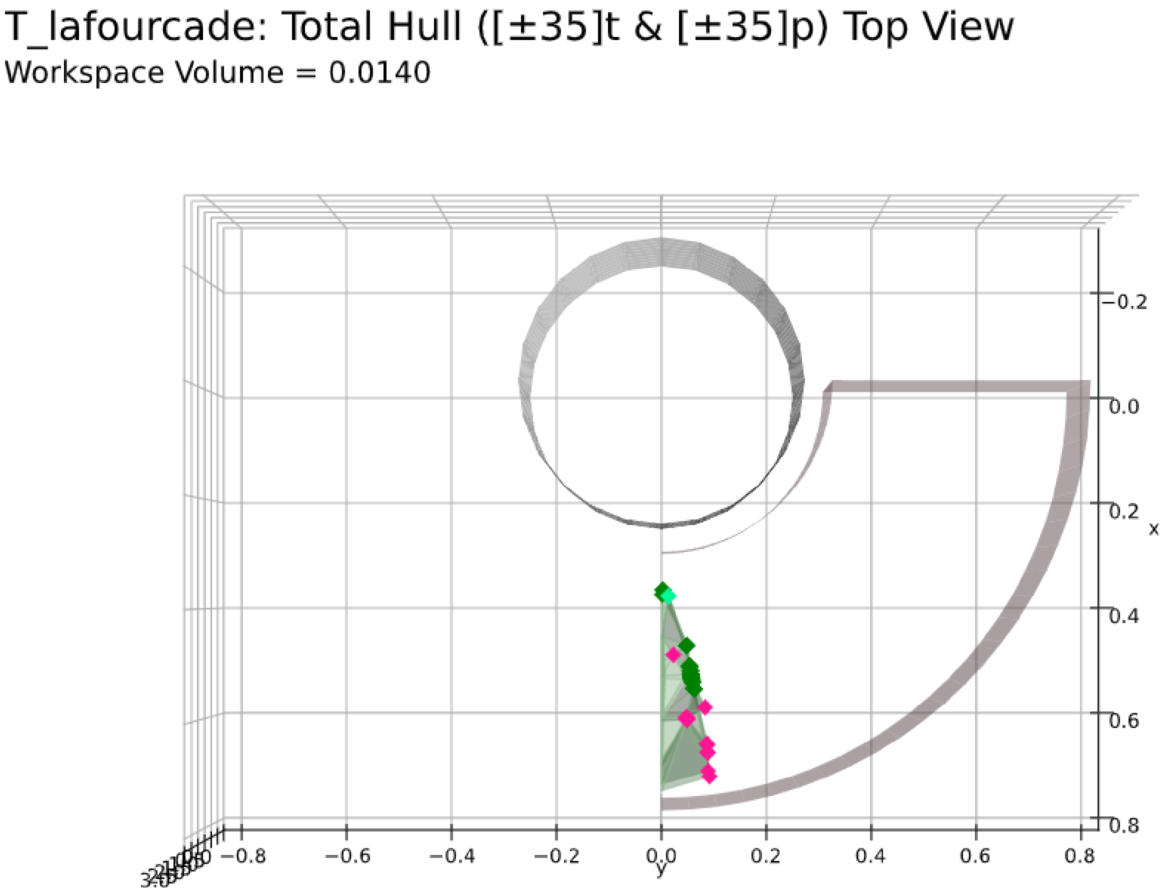


T_falcon: Total Hull ([±35]t & [±35]p) Top View

Workspace Volume = 0.0382

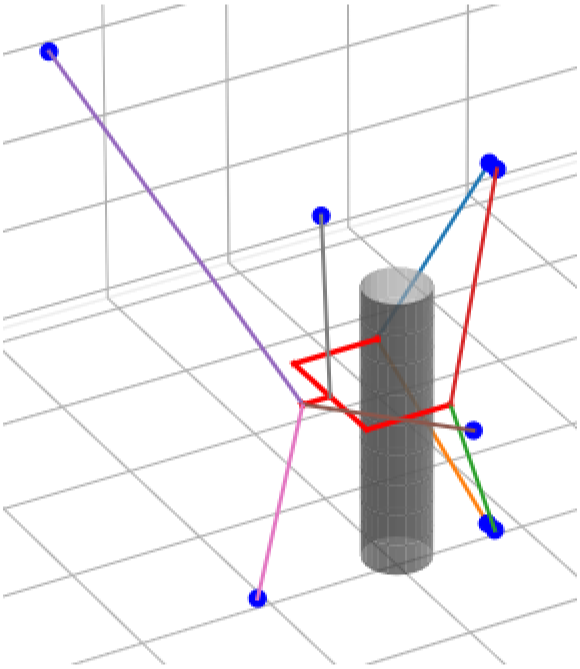


T-Lafourcade



T_lafourcade: Total Hull ([±35]t & [±35]p) Top View

Workspace Volume = 0.0140



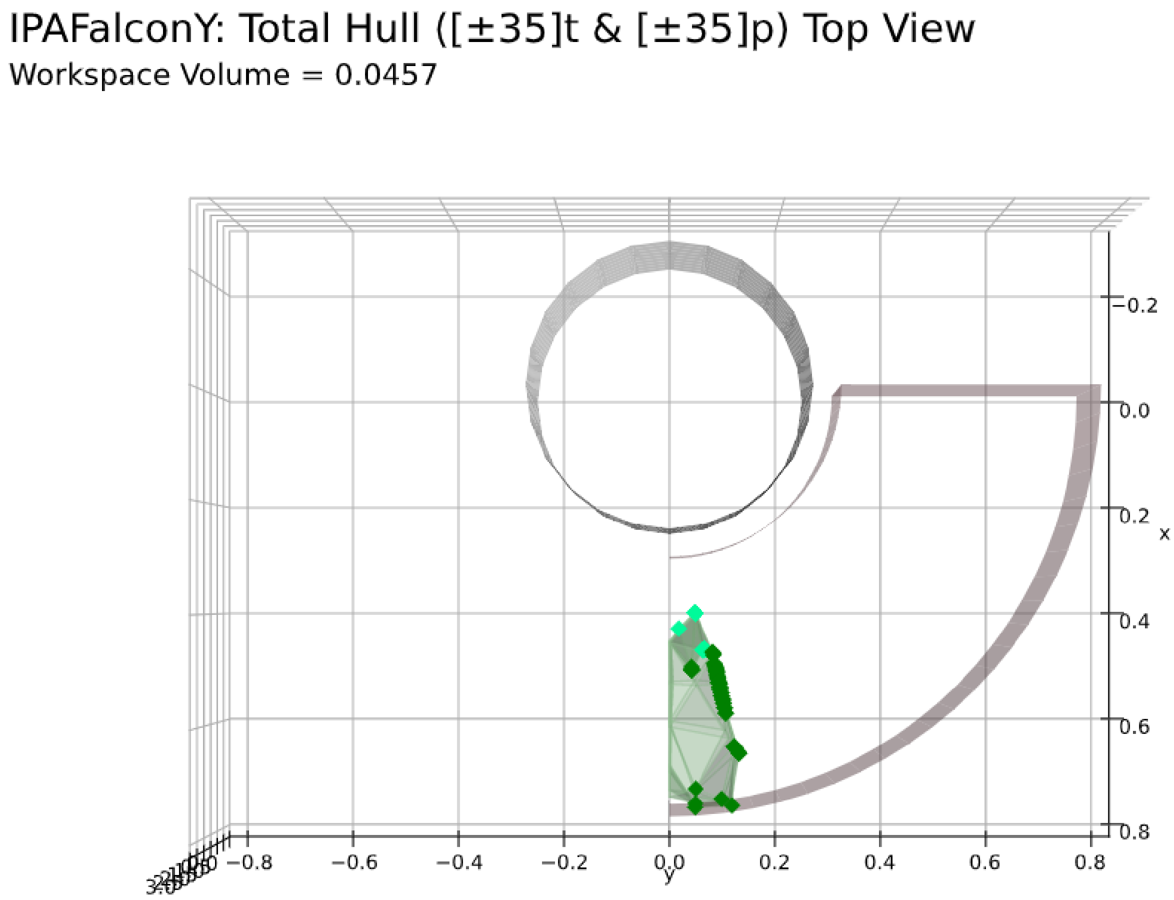
H-Platform

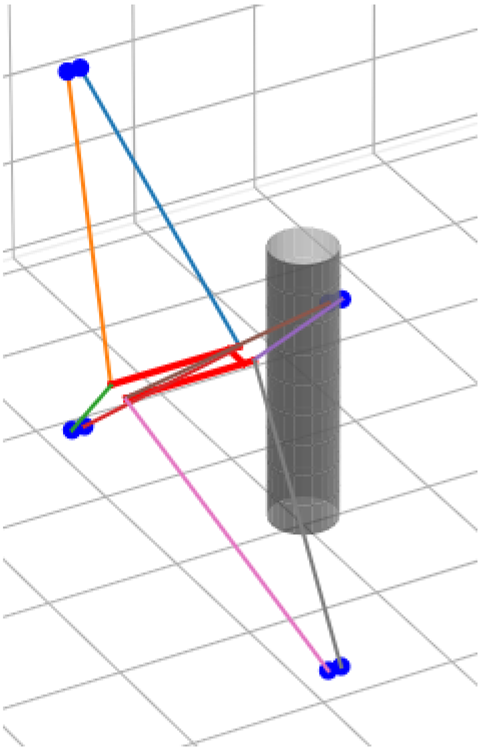
N.a.

C-IPAFalconX

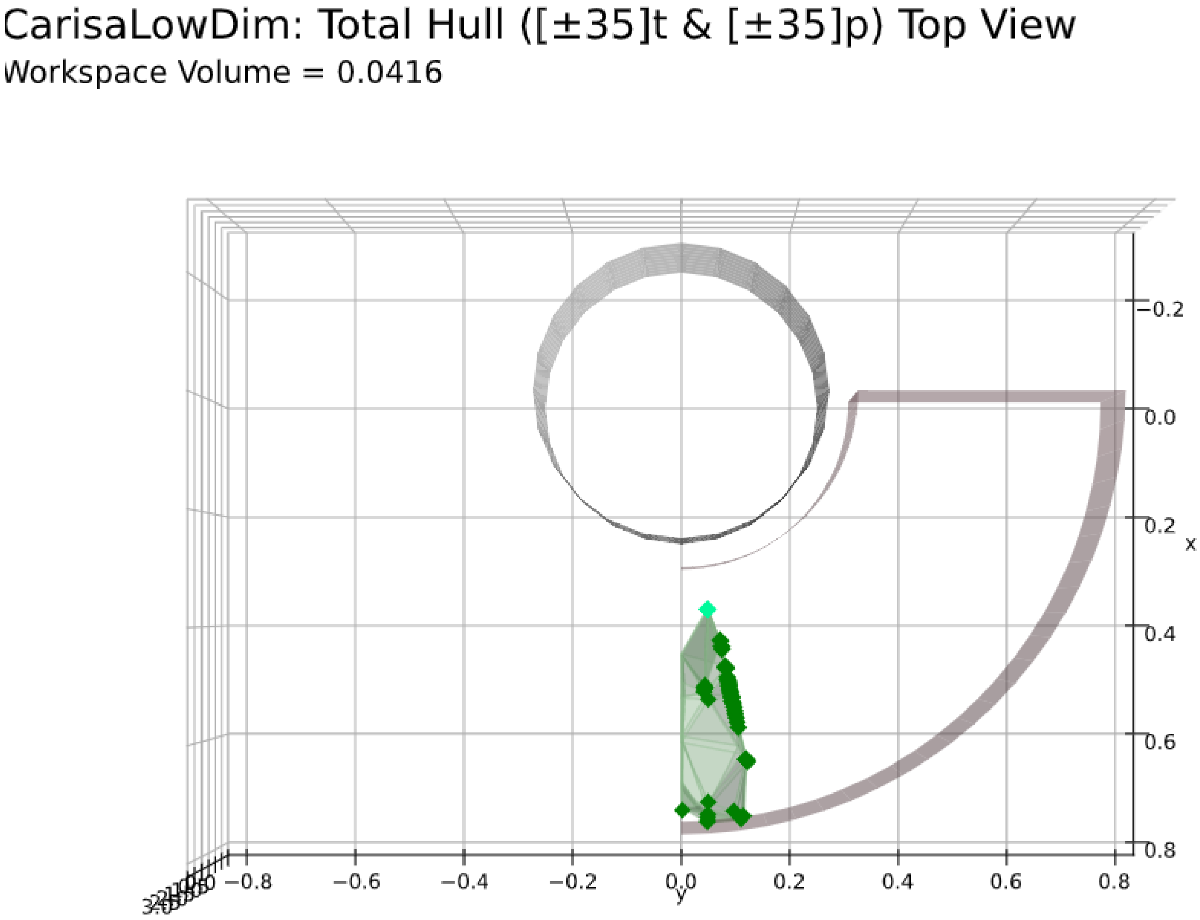
N.a.

C-IPAFalconY



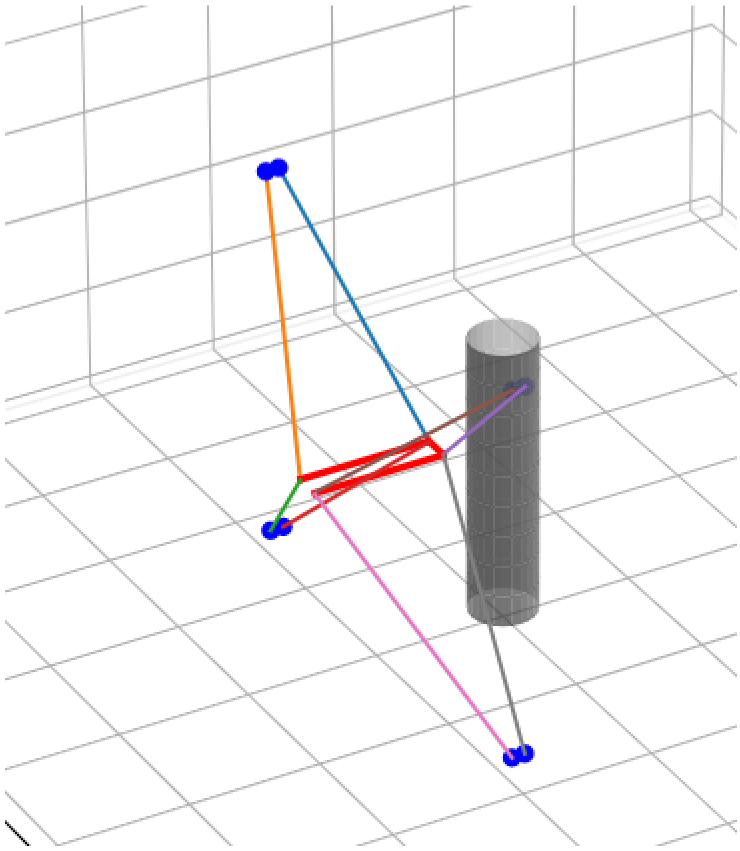


C-Carisa

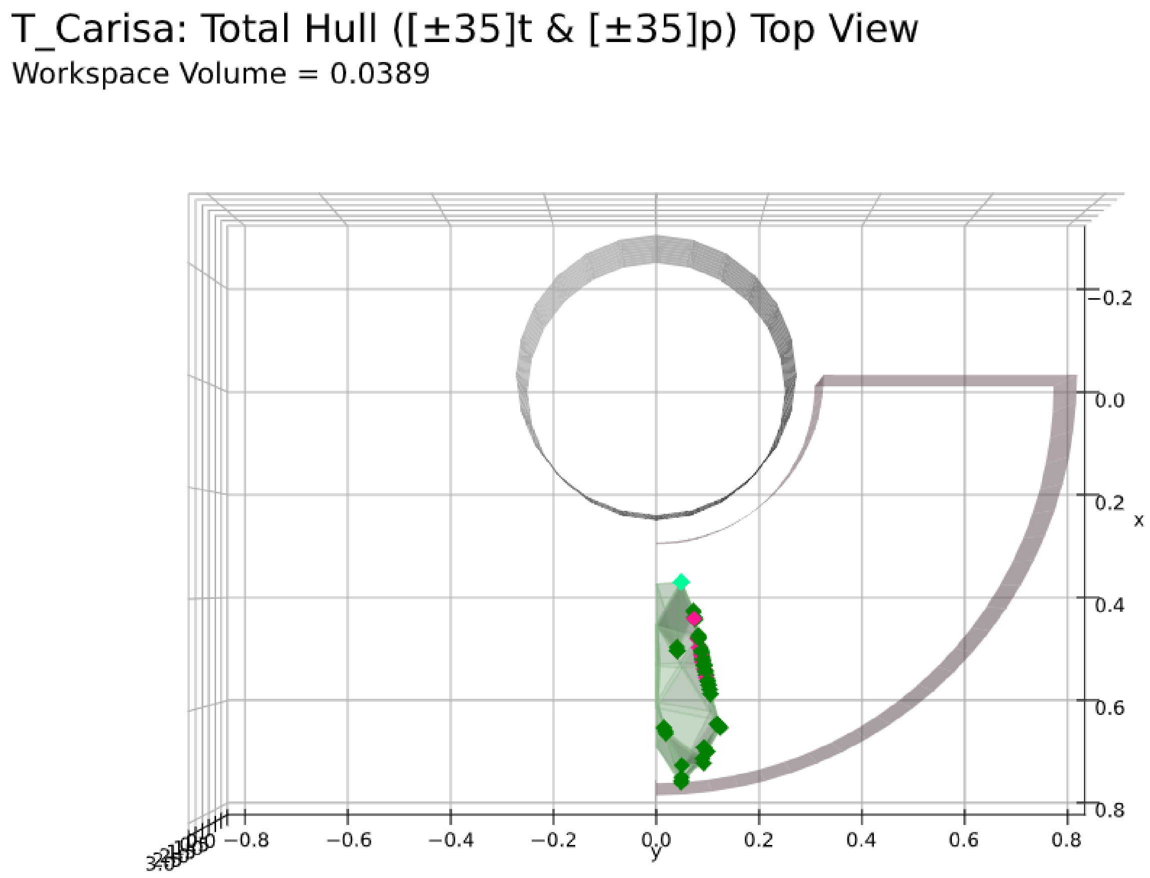


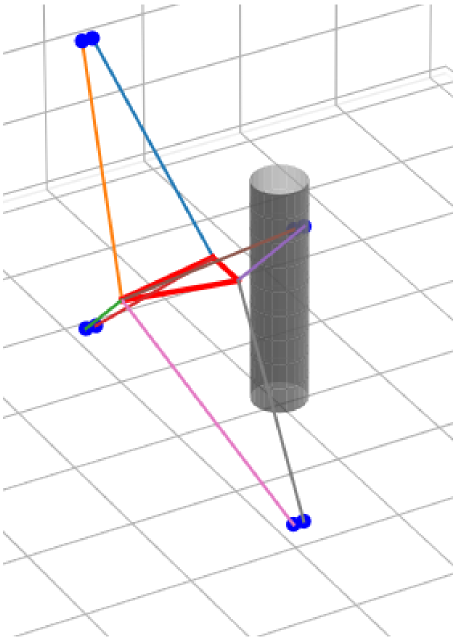
CarisaLowDim: Total Hull ([±35]t & [±35]p) Top View

Workspace Volume = 0.0416



T-Carisa

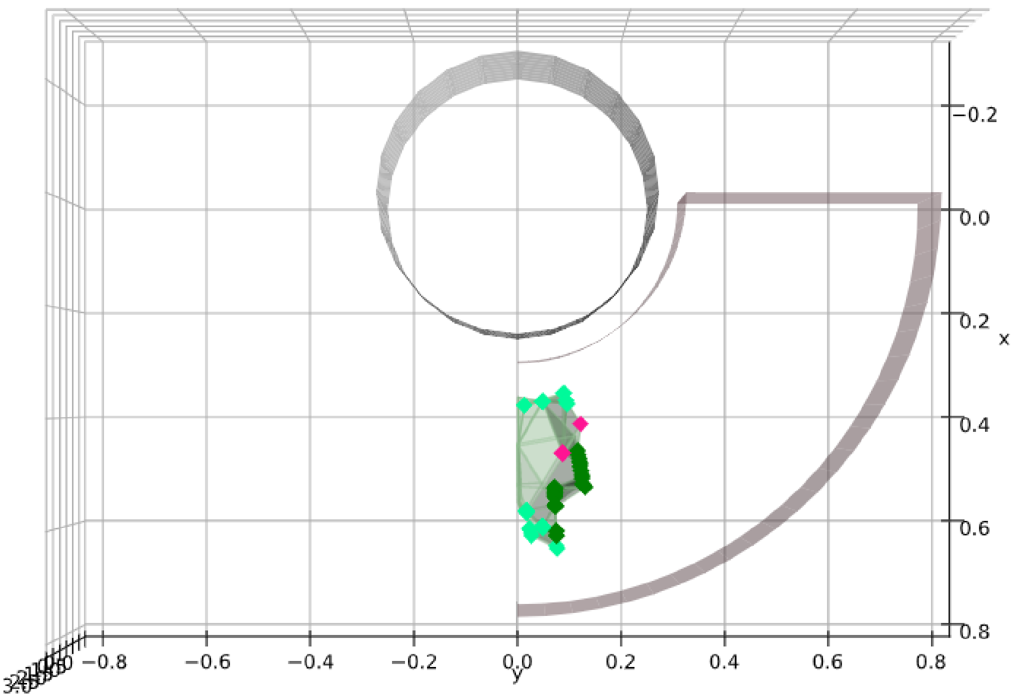


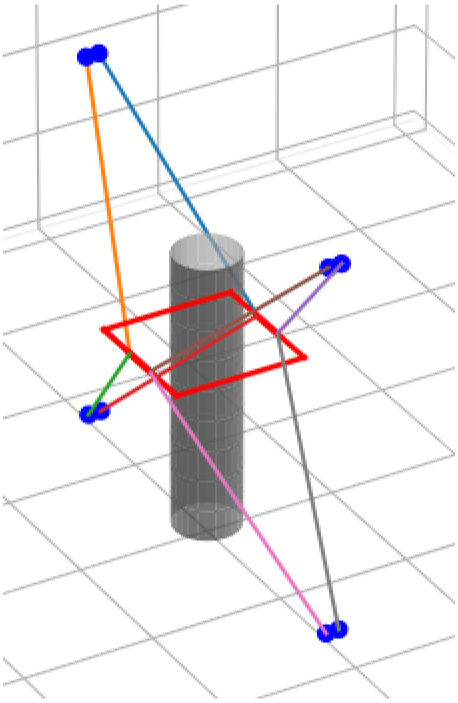


C-IPAFalcon_Encap

IPAFalcon_Encap: Total Hull ([±35]t & [±35]p) Top View

Workspace Volume = 0.0301





4: Optimization bounds

Large

T-Nishioka

Bounds:
W1: (0, 2)
H1: (-1, 1)
W2: (0, 2)
H2: (-1, 1)
X_19: (0, 3)
Z_19: (3, 4)
X_14: (0.4, 1.5)

T-Falcon

```
bounds=[(0, 2), #lp
          (0, 2), #lp_b
          (0.5, 2), # wp
          (-0.5, 1.5), # x_0
          (-1, -3), # y_0
```

T-Lafourcade

```
bounds=
[(0, 2),# lp
(0, 2), # lp_b
(1, 2), # wp
(-0.75, -3), # x_0
(0.3, 1.9), # x_17
(1, 3), # x_11
(-1, -3)], # y_11
```

H-Platform

```
bounds=[(0, 1),  # lp
          (0.207, 2), # wp
          (0, 1), # hp
          (-1, -3),  # x_0
          (3.5, 4), # z_0
          (1, 3),], # x_1
```

C-IPAFalconX

```
bounds=
[(0, 2), #lp
 (0, 2), #lp_b
 (0.7, 2), #wp
 (0.02, 1), #hp
 (-0.7, -3), #x0
 (0.75, 3)], #x2
```

C-IPAFalconY

Bounds:

lp: (0, 2)
Lp_b: (0, 2)

W_p: (0.207, 1)
Hp: (0.02, 1)
x_0: (-1, 2)
y_0: (-1, -3)

C-Carisa

Lp_b: (0, 2)
wp: (0.207, 2)
X_0: (0.5, 2.5)
y_0: (-1, -3)

T-Carisa

Lp_b: (0, 2)
wp: (0.207, 2)
X_0: (0.5, 2.5)
y_0: (-1, -3)

C-IPAFalcon_Encap

```
bounds=
np.array([[0.77, 2], #lp
          [0.02, 0.5], #wp
          [-0.5, 0.5], #x0
          [-1, -3]]), #y0
```

Small

T-Nishioka

Bounds:
W1: (0, 1.5)
H1: (-1, 1)
W2: (0, 1)
H2: (-1, 1)
X_19: (0, 1.5)
Z_19: (3, 4)
X_14: (0.4, 1.5)

T-Falcon

```
(0, 1), #lp
(0.6, 1), # wp
(-0.5, 1.5), # x_0
(-1, -3), # y_0
(3.5, 4)], # z_0
```

T-Lafourcade

```
bounds=
[(0, 1),# lp
(0.125, 1), # lp_b
(-0.75, -3), # x_0
(0.3, 1.9), # x_17
(1, 3), # x_11
(-1, -3)], # y_11
```

H-Platform

C-IPAFalconX

```
bounds=[(0, 1), #lp
        (0, 1), #lp_b
        (-0.7, -3), #x0
        (0.75, 3)], #x2
```

C-IPAFalconY

Bounds:
lp: (0, 1)
W_p: (0.7, 1) x_0: (-.5, 1.5)
y_0: (-1, -3)
Z_0: (3.5, 4)

C-Carisa

Bounds:
wp: (0.207, 1)
X_0: (0.524, 1.524)
y_0: (-1, -3)
z_0: (3, 4)

T-Carisa

Bounds:
wp: (0.207, 1)
X_0: (0.524, 1.524)
y_0: (-1, -3)
z_0: (3, 4)

C-IPAFalcon_Encap

```
([[0.77, 1], #lp
     [0.02, 0.5], #wp
       [-0.5, 0.5], #x0
      [-1, -3]]), #y0
```

5: Optimization constraints

T-Nishioka

- Multiple constraints to not let the dimensions of the platform triangle exceed 2m.
- Constraint to let point 1 of the platform not interfere with the statue while tilting.
- Constraints to prevent x_19 from surpassing the x values of the platform extremes while moving in the workspace.

T-Falcon

- Constraint on the total length of lp and lp_b
- Constraints to prevent x_0 from surpassing the x values of the platform extremes while moving in the workspace.

T-Lafourcade

- No constraints

H-Platform

- Constraints to make sure the distal endpoints x -locations are to some extent mirrored in the zy axis. When they are not close to being mirrored, no feasible designs are found.

C-IPAFalconX

- Constraint on the total length of lp and lp_b

C-IPAFalconY

- Constraint on the total length of lp and lp_b
- Constraints to prevent x_0 from surpassing the x values of the platform extremes while moving in the workspace.

C-Carisa

- Constraint on the total length of lp and lp_b
- Constraints to prevent x_0 from surpassing the x values of the platform extremes while moving in the workspace.

T-Carisa

- Constraint on the total length of lp and lp_b
- Constraints to prevent x_0 from surpassing the x values of the platform extremes while moving in the workspace.

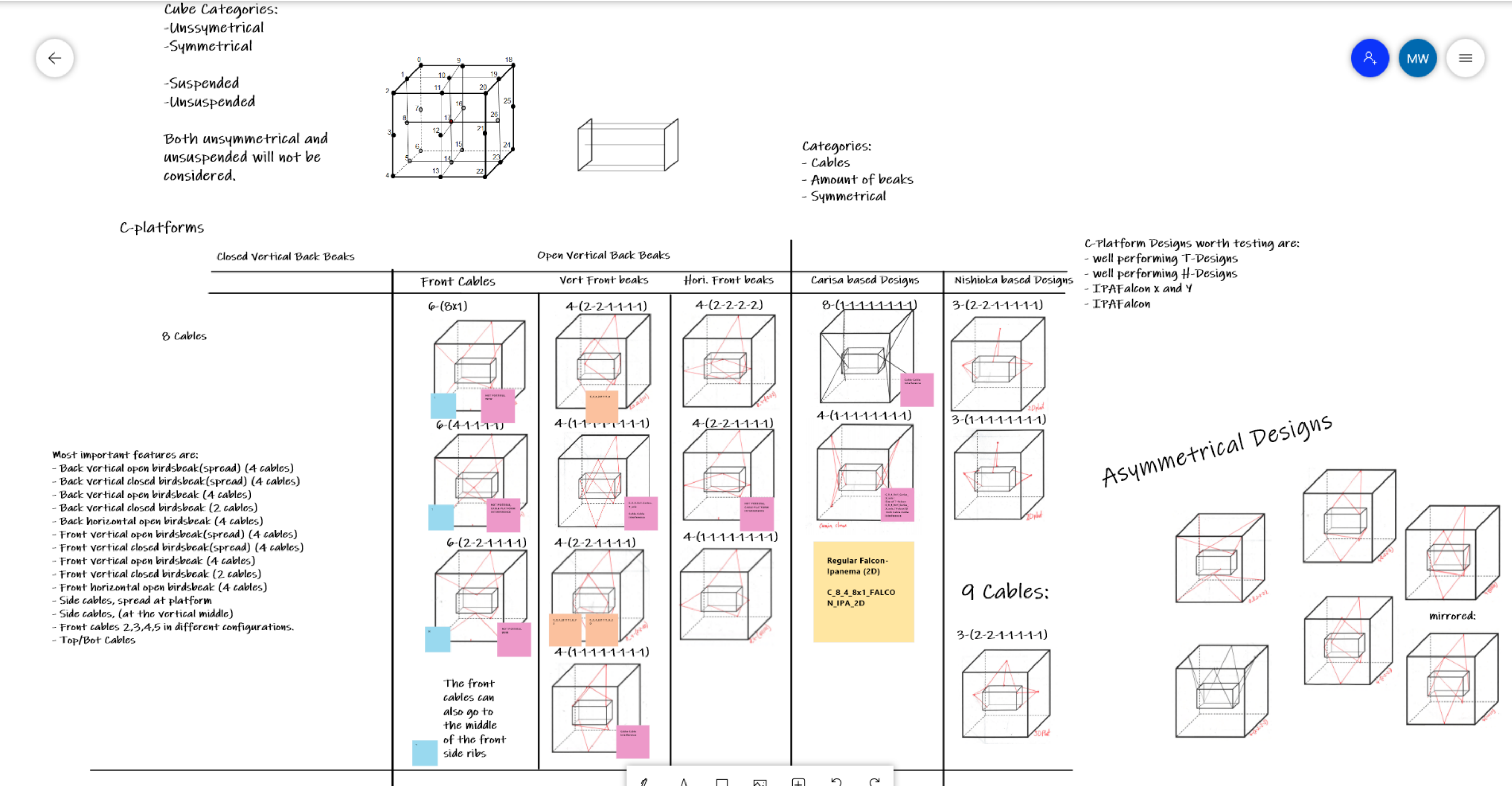
C-IPAFalcon_Encap

No constraints.

6: Brainstorm tables

C-Platforms:

- Next page: upperhalf
- Page thereafter: lower half



Cube Categories:
-Unssymetrical
-Symmetrical

-Suspended
-Unsuspended

Both unsymmetrical and unsuspended will not be considered.

Categories:
- Cables
- Amount of beaks
- Symmetrical

C-platforms

Closed Vertical Back Beaks

Open Vertical Back Beaks

Front Cables

Vert Front beaks

Hori. Front beaks

Carisa based Designs

Nishioka based Designs

8 Cables

6-(8x1)

6-(4-1-1-1-1)

6-(2-2-1-1-1-1)

The front cables can also go to the middle of the front side ribs

4-(2-2-1-1-1-1)

4-(1-1-1-1-1-1-1-1)

4-(2-2-1-1-1-1)

4-(1-1-1-1-1-1-1-1)

4-(2-2-2-2)

4-(2-2-1-1-1-1)

4-(1-1-1-1-1-1-1-1)

8-(1-1-1-1-1-1-1-1)

4-(1-1-1-1-1-1-1-1)

Regular Falcon-Ipanema (2D)

C_8_4_8x1_FALCON_IPA_2D

3-(2-2-1-1-1-1)

3-(1-1-1-1-1-1-1-1)

9 Cables:

3-(2-2-1-1-1-1-1)

C-Platform Designs worth testing are:
- well performing T-Designs
- well performing H-Designs
- IPAFalcon x and Y
- IPAFalcon

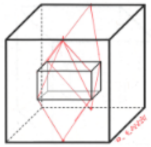
Asymmetrical Designs

mirrored:

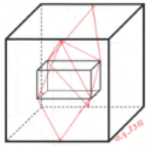
Most important features are:
- Back vertical open birdsbeak(spread) (4 cables)
- Back vertical closed birdsbeak(spread) (4 cables)
- Back vertical open birdsbeak (4 cables)
- Back vertical closed birdsbeak (2 cables)
- Back horizontal open birdsbeak (4 cables)
- Front vertical open birdsbeak(spread) (4 cables)
- Front vertical closed birdsbeak(spread) (4 cables)
- Front vertical open birdsbeak (4 cables)
- Front vertical closed birdsbeak (2 cables)
- Front horizontal open birdsbeak (4 cables)
- Side cables, spread at platform
- Side cables, (at the vertical middle)
- Front cables 2,3,4,5 in different configurations.
- Top/Bot Cables

10 Cables

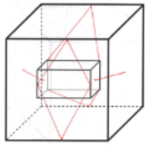
4-(2-2-2-2-1-1)



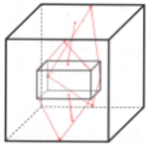
4-(3-3-2-2)



6-(2-2-1-1-1-1-1-1)

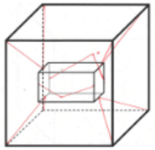


6-(2-2-1-1-1-1-1-1)

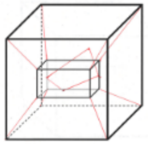


12 Cables

10-(2-2-2-2-1-1-1-1)

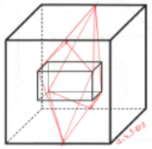


10-(1-1-1-1-1-1-1-1-1-1-1-1-1)

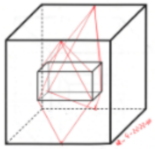


Instead of opening to the frame, let the cables close to the frame.

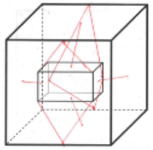
4-(3-3-3-3)



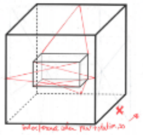
4-(2-2-2-2-1-1-1-1)

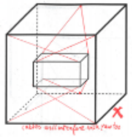


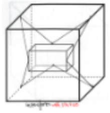
8-(2-2-1-1-1-1-1-1-1-1)



Interfering Designs

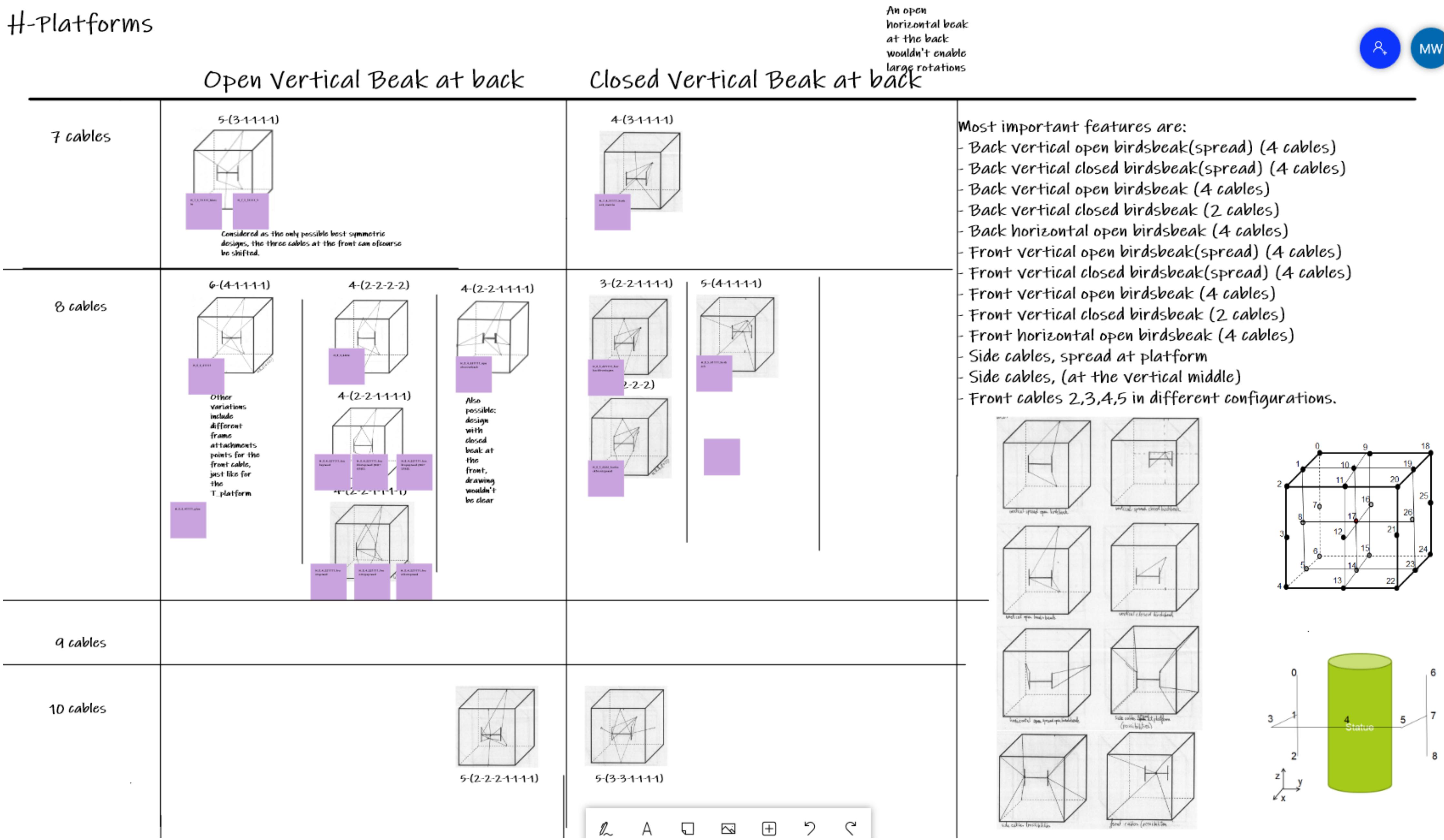






H-Platforms:

- On next page



H-Platforms

An open horizontal beak at the back wouldn't enable large rotations

	Open Vertical Beak at back	Closed Vertical Beak at back
7 cables	5-(3-1-1-1-1) Considered as the only possible best symmetric designs, the three cables at the front can ofcourse be shifted.	4-(3-1-1-1-1)

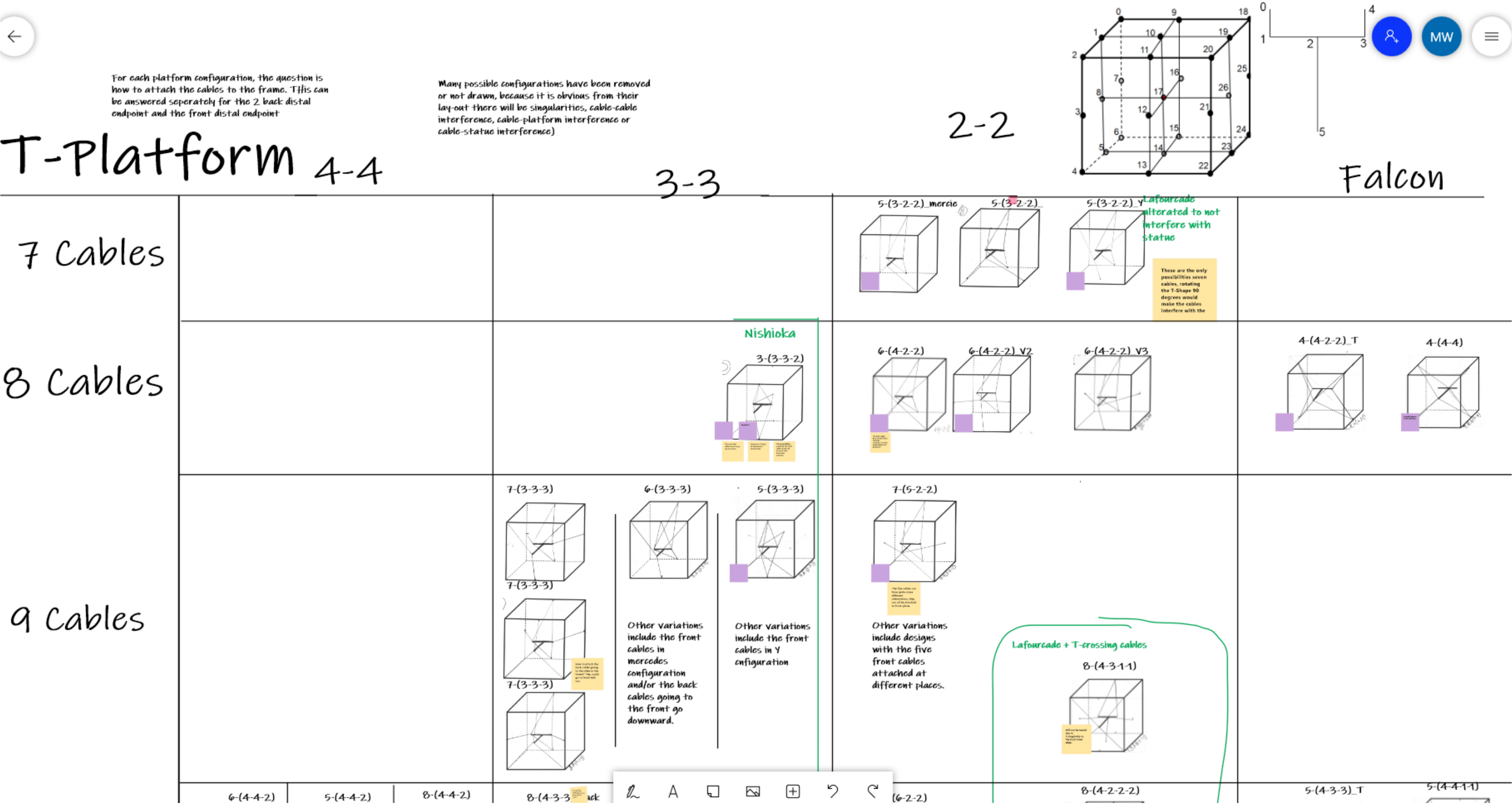
8 cables	6-(4-1-1-1-1)
	Other variations include different frame attachments points for the front cable, just like for the T_platform
	4-(2-2-2-2)
	4-(2-2-1-1-1-1)
	4-(2-2-1-1-1-1)
	4-(2-2-1-1-1-1)
	Also possible: design with closed beak at the front, drawing wouldn't be clear
		3-(2-2-1-1-1-1)
		2-2-2)
		5-(4-1-1-1-1)

9 cables		
10 cables	5-(2-2-2-1-1-1-1)	5-(3-3-1-1-1-1)

Most important features are:
- Back vertical open birdsbeak(spread) (4 cables)
- Back vertical closed birdsbeak(spread) (4 cables)
- Back vertical open birdsbeak (4 cables)
- Back vertical closed birdsbeak (2 cables)
- Back horizontal open birdsbeak (4 cables)
- Front vertical open birdsbeak(spread) (4 cables)
- Front vertical closed birdsbeak(spread) (4 cables)
- Front vertical open birdsbeak (4 cables)
- Front vertical closed birdsbeak (2 cables)
- Front horizontal open birdsbeak (4 cables)
- Side cables, spread at platform
- Side cables, (at the vertical middle)
- Front cables 2,3,4,5 in different configurations.

T-Platforms:

- On next page: upper half
- Page thereafter: lower half



For each platform configuration, the question is how to attach the cables to the frame. This can be answered seperately for the 2 back distal endpoint and the front distal endpoint

Many possible configurations have been removed or not drawn, because it is obvious from their lay-out there will be singularities, cable-cable interference, cable-platform interference or cable-statue interference)

T-Platform

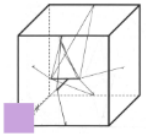
	4-4	3-3	2-2	Falcon
7 Cables			5-(3-2-2)_mercie 5-(3-2-2) 5-(3-2-2)_Y Lafourcade alterated to not interfere with statue These are the only possibilities seven cables, rotating the T-Shape 90 degrees would make the cables interfere with the	
8 Cables		Nishioka 3-(3-3-2)	6-(4-2-2) 6-(4-2-2)_V2 6-(4-2-2)_V3	4-(4-2-2)_T 4-(4-4)

9 Cables		7-(3-3-3)
		7-(3-3-3)
		7-(3-3-3)
		6-(3-3-3)
		Other variations include the front cables in mercedes configuration and/or the back cables going to the front go downward.
		5-(3-3-3)
		Other variations include the front cables in Y configuration
			7-(5-2-2)
			Other variations include designs with the five front cables attached at different places.
			Lafourcade + T-crossing cables
			8-(4-3-1-1)	

	6-(4-4-2)
	5-(4-4-2)
	8-(4-4-2)
		8-(4-3-3 ack	(6-2-2)
			8-(4-2-2-2)
				5-(4-3-3)_T
				5-(4-4-1-1)

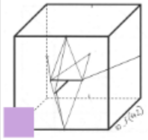
10 Cables

6-(4-4-2)



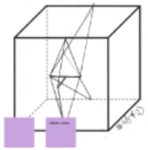
Other variations include the two side cables attached at different points and/or the front beak going downward.

5-(4-4-2)

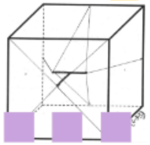


Other variations include the two side cables attached at different points

8-(4-4-2)



8-(4-3-3 ack



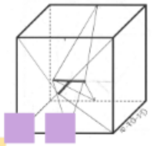
Could be a 9 cable variety with 4 instead of 3 cables up front

The four front cables can be attached differently as discussed for less cables.

The side back cables can go to mid/front as well, these will be the varieties.

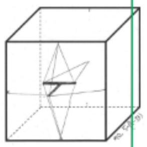
Other variations include the four front cables at different frame points, the two side cables attached at different frame points.

7-(4-3-3)



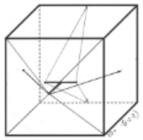
Other variations include the front bird beak going upwards and/or the four front cables attached at other frame points

5-(4-3-3)



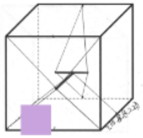
Other variations include the front bird beak going upwards and/or the four front cables attached at other frame points

8-(6-2-2)

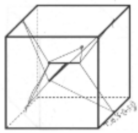


Other variations include designs with the six front cables attached at different places.

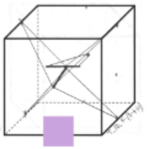
8-(4-2-2-2)



5-(4-3-3)_T

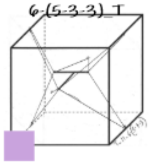


5-(4-4-1-1)

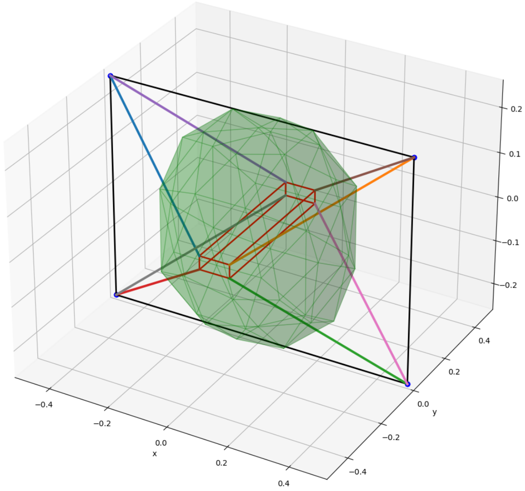


11 Cables

6-(5-3-3)_T



7: Interference Reports (Methods)



Cable-Driven Parallel Robots for 3D Art Object: Methods (Unoffical document, this was used for my own documentation) Reasoning for cable interference is found in this document.

September 17, 2021

TU Delft
Delft, Netherlands
Mees Waal

Contents

1 Creating the CDPR Model

To perform the optimization a model of the CDPR was created in Python. This model consists of several different aspects:

1.1 Creating an architecture

The architecture comprises of the frame coordinates, the platform coordinates and the cable lay-out.

1.1.1 Platform and frame coordinates

The platform and frame coordinates are described with a function called cdpr_dimensioning. Its inputs are the width, height and length of both the frame and platform and six other parameters d_lf, d_hf, top_off_x, top_off_y, top_off_z and bot_off_Z. d_lf, d_hf give the distal endpoints unique locations when the frame width is zero. This would be the case for a FALCON robot. top_off_x, top_off_y, top_off_z are used to slightly offset the top of the frame. bot_off_z can be used to place all the bottom distal endpoints to the top. The output of the cdpr_dimensioning function are the frame coordinates expressed in the world frame and the platform coordinates expressed in the local frame.

1.1.2 Changing the Cable Lay-out

The cable Lay-out is changed with the function cable_shuffle. It restructures the matrix with the distal endpoints. It is used within the function describing the Cable Kinematics. Its inputs are the matrix describing the distal endpoints and a 'ShuffleKey'. This is a vector with a length of eight, the position of every entry describes the frame coordinate and its value represents the platform coordinate it is connected to. Figure 1 and figure 2 show the cable-layout for the Shufflekey [32107654] and [01234567] respectively. The frame and platform coordinates numbers are shown in figure 3.

1.2 Kinematic equations

The available wrench set and the force distributions are not the same[23]. The available wrench set is calculated for a given pose, it describes all the wrenches that the platform can withstand for certain cable limits.

The available poses are calculated for a given wrench set, it determines whether a pose is possible for a certain wrench set. This is done by checking at a pose whether there is a certain force distribution in the cable that can withstand the wrench at that pose.

Pott[23] and Perreault[22] use the latter method to determine wrench feasibility. Gouttefarde[8], Hussein[?] , Kamali[11] and Gagliardini[7] all use the method introduced by Bouchard[6] to find the available wrench set. Yi[31] relies on a method based on the method of[15] which tests whether a desired cable tension at a pose is admissible.[2] uses this method as well, it is the same method as the closed-form method proposed in[23]. There are combinations of cable forces that do not contribute to the exerted wrench, but only to the internal forces. The force vector of the internal force lies within the nullspace of the jacobian[5].[5] cites[13] to state that the cable forces have effect on decreasing vibrations and increasing stiffness.[16] and[24] use the capacity margin index, which relies on calculating the wrench space.

In section 5.3.7 of[23] multiple methods to compute the workspace are discussed. The available wrench set method suffers from poor numerical performance. The closed-form method is

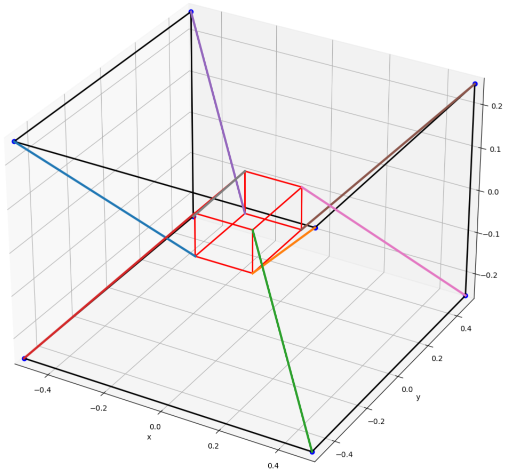


Figure 1: A CDPR with the [32107654] configuration

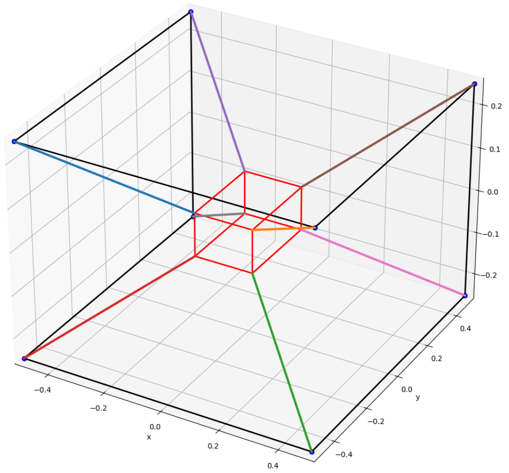


Figure 2: A CDPR with the [01234567] configuration

workable, when applied to fully constraint robots, but only finds a subset of the workspace for suspended cable robots. Linear programming can be used when it is not important to find continuous solutions. Considering computation time, solving linear systems takes the least amount of computation time to solve for feasible poses[23]. The Scipy package in Python is offering the function linprog to solve linear equations. The interior point algorithm is applied by this linprog function.

The function to solve the kinematics of the CDPR is called cdpr_kinematics. It has many inputs:

1. The frame coordinates expressed in the world frame and the platform coordinates expressed in the local frame from the function CDPRDimensioning.
2. The platform position in world coordinates
3. The rotation of the platform in world coordinates
4. The degrees of freedom
5. The amount of cables
6. The applied wrench
7. The cable force limits
8. The ShuffleKey

In the function the distal endpoints are first expressed in world coordinates. Then the structure matrix is defined. Subsequently, a set of linear equations is defined according to the equations 3.40 - 3.46 in[23]. The function both returns a boolean that states whether a solution has been found and provides the found cable force distribution.

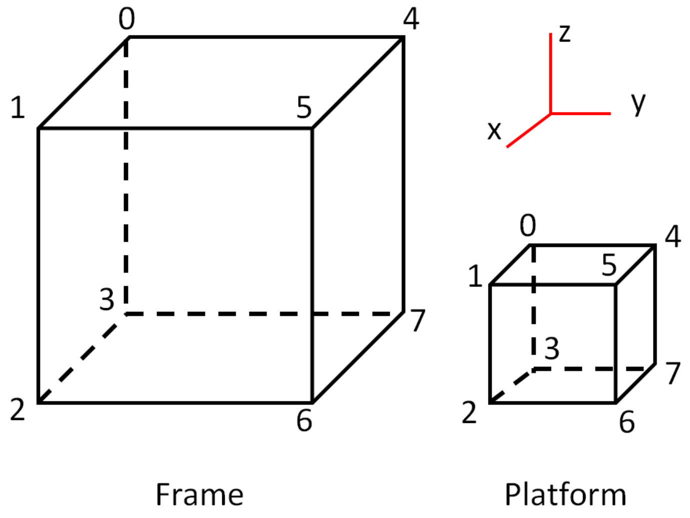


Figure 3: The numeration of endpoints used in the scripts.

1.3 Workspace Computation

Some numerical methods cannot work with intervals or are slow when using intervals. When using boundary methods such as the Hull Method, these numerical approaches can be used. With the Hull method the boundary of the workspace is approximated through triangulation. Its advantages are its speed and precision and straightforward and precise determination of the volume of the workspace. Disadvantages over a grid method are that they are more elaborate to implement and they offer little information about the points within the workspace[23]. The method is initialized by defining an octahedron or other triangular shape in the expected middle of the workspace. Especially the icosahedron is interesting since these solids lead to a perfectly regular structure for the hull, so all vertices share the same number of triangles[23]. Each triangle of the shape could then be subdivided into more triangles to increase the accuracy. Subsequently the vertices of the triangles are projected onto the boundary of the workspace with the line search algorithm from page 194 in[23]. To create an icosahedron, programming code found online was used[29].

To create a workspace hull, the function hull_method has been used. It uses the cdpr_kinematics function as workspace evaluation inside so the function hull_method has all the inputs this function has. Furthermore, it has the inputs lambda_minInit and lambda_maxInit which are the initial values for the range of the line search algorithm, the input mid, which is the starting value from which the line search algorithm is started, the input eps, which defines the convergence condition of the line search and the input recur_lvl, which is the recursion level of the icosahedron. The inputs vmatrix and index_matrix contain the parameters to create an icosahedron with an origin at $[0, 0, 0]$ and the distance of the vertices of the icosahedron to the origin of 1. In[23] a method with 8192 triangles is used and eps $= 0.0001$. The line search method is used as a for loop with 1000 iterations. When the maximum amount of iterations is reached, a warning is printed with the statement that the line method did not converge. Every vertex of the icosahedron is evaluated with the line search method. Furthermore the area, volume and centre of gravity of the workspace are computed. The formula's from page 196 of[23] are used for this purpose. The formulas in the script for area and volume were checked by applying them to an icosahedron with recursion level 0 and a distance to the origin of every vertex of zero. The results were compared with the formulas from[12] for the area and volume of such a general icosahedron.

1.4 Visualization

The function cdpr_plot has as inputs the proximal endpoints in world coordinates, the distal endpoints in world coordinates and the distal endpoints in world coordinates shuffled to draw the cables.

The function hull_plot has as inputs the location of the vertices of the hull in world coordinates and a matrix that describes which vertices form triangles together. Furthermore, it prints the volume, area and centre of gravity of the workspace, so it takes these values as input as well.

1.5 Performance Indices

In Guay[10] the manipulability and dexterity are described as performance indices that describe the ellipsoid from mapping the unit sphere of the jacobian. It is stated that the dexterity has drawback of merely giving the size ratio of the ellipsoid but not its size and the dexterity index is not representative when the jacobian matrix is not homogenous in all dimensions[10].

1.6 Notes to self

1. In case of the standard simple version of the cable-driven parallel robot, i.e. a cube shaped frame and platform, there are many singularities because the structure matrix is not full rank. This can be changed for example by manipulating the platform to give it a shape that has only rectangles as sides instead of cubes.

2. Should the required wrench set $[w]_r$ just be 5N in negative vertical direction or should some torsion also be added? Answer: Yes, some torsion should be added.

2 Adding Interference

The interferences need to be considered to provide model that has the same limits in movements as the real-life model. Interferences can be avoided with trajectory planning, but trajectory planning will not be considered in this research, because a functioning CDPR without extra trajectory planning would be a more simple solution. Ideally interference detection is integrated into the model for determining the workspace with feasible poses. During this research, interference detection is implemented in multiple ways and subsequently tested. The findings are presented below for each method.

2.1 Methods from literature

2.1.1 Perreaults Analytical Method

This method is an analytical method presented by Perreault[21]. For each pair of cables and a fixed orientation of the end-effector, a certain plane is calculated. When the origin of the end-effector coincides with the plane, cable interference between the cables take place. The benefits of this method are that it provides an exact solution of all the positions for which interference happens without the solution being trajectory dependent. Also, the planes can help to create an understanding of where the regions of interference lie. Finally, it provides a graphical interpretation of the areas of interference as seen in figure 4. The downside of this method is that all the planes of the different cable pairs have to be combined and interpreted to form a workspace. This workspace is then only valid for one orientation of the platform.

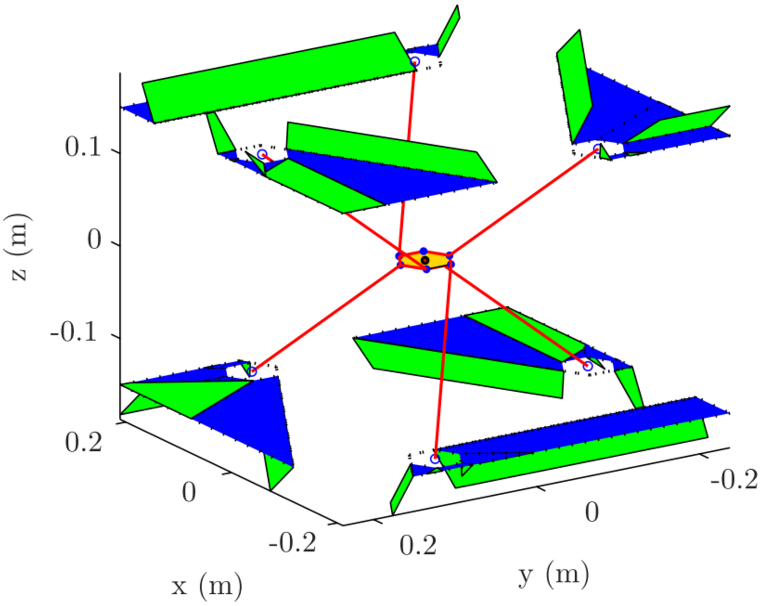


Figure 4: Planes representing cable- cable and cable-platform interference, taken from[21].

2.1.2 Perreaults Numerical Method

This method is a numerical method and is introduced by Perreault as well[22]. The method checks whether the cables are colliding at a pose or have collided while moving to the pose. The latter is determined by checking whether the cables have crossed between two poses. The catch is that cables can appear to have crossed each other but have not actually collided. A cable crossing has taken place when a sign change of the distance between the cables is detected. The formula for the sign of the distance between two cables is as follows[19]:

$$s_{ij} = \text{sign}\left(\left(\overrightarrow{A_iB_i} \times \overrightarrow{A_iB_j}\right)^T \cdot \overrightarrow{A_iA_j}\right) \tag{1}$$

A crossing is visualised in figure 7. The cables collide when all of the following three conditions from[22] relating to figure 5 are satisfied:

- $0 \leq n_1 \leq$ Length Cable 1
- $0 \leq n_2 \leq$ Length Cable 2
- $|n_r| \leq \epsilon$ or n_r changes sign while moving between two poses.

The values for n_r, n_1 and n_2 multiplied with the corresponding unit vectors form the vectors $\mathbf{n_r}$, $\mathbf{n_1}$ and $\mathbf{n_2}$. Their formula are as follows[22]:

$$\mathbf{n}_r = n_r \frac{(\mathbf{n}_1 \times \mathbf{n}_2)}{|\mathbf{n}_1 \times \mathbf{n}_2|} = n_r \frac{(\mathbf{u}_1 \times \mathbf{u}_2)}{|\mathbf{u}_1 \times \mathbf{u}_2|} \tag{2}$$

$$\mathbf{n}_1 = -n_1 \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|} \tag{3}$$

$$\mathbf{n}_2 = -n_2 \frac{\mathbf{u}_2}{\|\mathbf{u}_2\|} \tag{4}$$

The norms are then calculated by solving the following system of equations[22]:

$$\mathbf{n}_1 + \mathbf{n}_r - \mathbf{n}_2 - \mathbf{r} = 0 \tag{5}$$

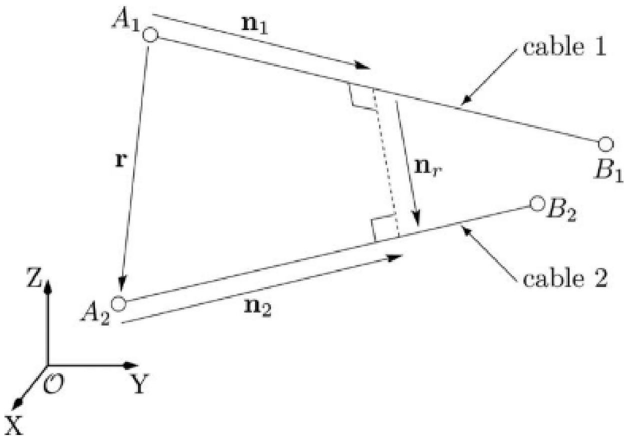


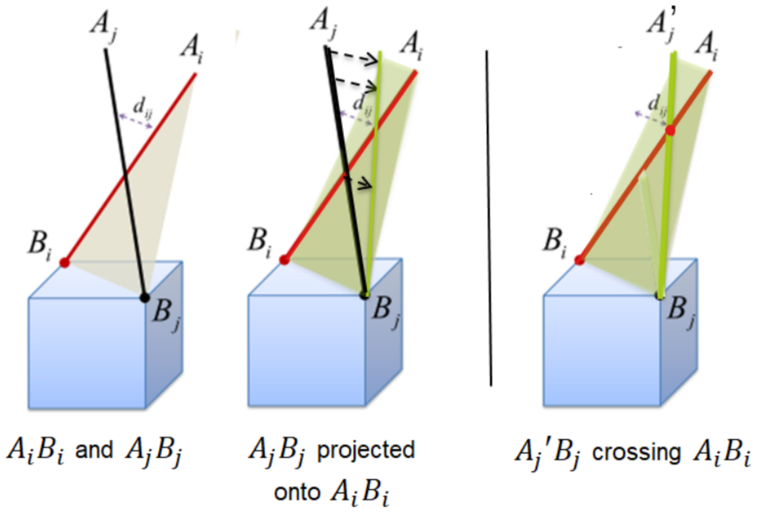
Figure 5: Analyzing the relative position of two cables, taken from[22].

2.1.3 Nguyen's Numerical Method

This method is a numerical method and is taken from[19]. It is an altered version of Perreaults Numerical Method. Firstly it omits checking whether the cables collide while being at a pose, but only checks whether the cables have crossed while moving to the pose. Secondly, when a sign change is detected, it is evaluated whether the cables have crossed inside or outside the range of their own lengths, so whether n_1 and n_2 lie between 0 and cable length l. This evaluation is done by projecting one cable to the plane of the other and then checking whether they collide within their cable length. Figure 6 helps to visualize that, a projection A'_jB_j of the cable A_jB_j is formed on the plane defined by $A_iB_iB_j$. The downside of the projection method is that the distance between two poses can be so large that the projection method is not reliable enough to determine whether a collision has happened in between. Either a 'missed detection' happens, so a collision has taken place but the cable projections only collide at the new position outside of their lengths, or a 'false alarm' happens, so the a collision hasn't really taken place, but the projections of the cable do collide with each other within their lengths. When not a single pair of cables has collided when moving to a new pose, the new pose is considered to be a part of the workspace.

2.2 Combined Methods

During the analysis of the methods, more methods were introduced to check whether they provide better results.



A_iB_i and A_jB_j

A_jB_j projected onto A_iB_i

$A_j'B_j$ crossing A_iB_i

Figure 6: Visualizing the projection of cables, partly taken from[19].

2.2.1 Perreaults Method without Sign Check

This method is almost the same method as Perreaults Numerical method but without the condition of the sign change. So it has the following conditions:

- $0 \leq n_1 \leq$ Length Cable 1
- $0 \leq n_2 \leq$ Length Cable 2
- $|n_r| \leq \epsilon$.

It only checks whether the cables are colliding at a single pose. These conditions were used to calculate a hull in two different ways. Either the hull was calculated with the LineSearch method, so a varying stepsize depending on the proximity to the solution, or the steps for evaluation of the poses was a fixed small incremental step.

2.2.2 Nguyens Method with Cable Distance Check

This method is a combination of Perreaults Numerical method and Nguyens Numerical Method. The Nguyen conditions need to be satisfied, as well as the conditions of the Perreaults Method without signcheck. So at a pose it is first checked whether the cables are with less than a cable diameter close to each other and whether this point lies within their cable length. When this is not the case it is checked whether there is a sign change and the projection of the cables on top of each other collide.

2.3 Testing the methods

To evaluate the cable detection of the different algorithms a few test cases were designed. Their exact parameters are found in Appendix 10. The lay-outs are shown in figures 8, 9, 10 and 11 and can be described as follows:

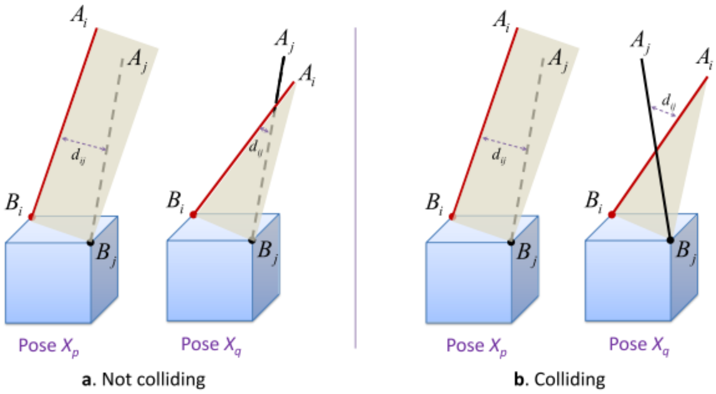


Figure 7: To the left two cables that haven't crossed, to the right two cables that did, taken from[19].

1. A rod shaped platform with an IPAnema cable configuration and a 15° rotation around the z-axis.
2. A pizza box-shaped platform rotated with $-45°$ around the z-axis and $10°$ around the y-axis with a cable configuration taken from[22].
3. A large rectangular platform with a cable-configuration taken from[9]. The platform is rotated 15° around the z-axis.
4. The same platform and cable configuration as in three, but now with a different rotation: $-5°$ around the z-axis, 10 degrees around the y-axis and an 5 degrees around the x-axis.

Perreaults numerical method without sign check and without line search can be assumed to be the benchmark algorithm, because it samples space thoroughly. Checking the iteration step for test case 3 it appeared that using the cable width as iteration step produced the same result as a step 5 times as small. A step two times the size of the cable width misses out on a collision for two vertices however. From these findings it is argued that using the cable width as an iteration step is reliable. This method cannot be used later on during optimization however, because it is too computationally expensive. The other algorithms are compared with Perreaults numerical method without sign check and line search for every test case.

Several conclusions are drawn from those test cases. The observations are first described for each test case and are summarized at the end to form a conclusion.

2.4 Test case 1

For the lay-out in figure 8 the different algorithms are compared with the benchmark in figure 12, 13, 14 and 15.

2.4.1 Observation 1

For some cable lay-outs some cables share distal or proximal endpoints. When this is case, the cables cannot collide and should be filtered out before evaluation. Furthermore, the system of equations to calculate the parameters of the closest distance between cables n_r, n_1 and n_2 is not

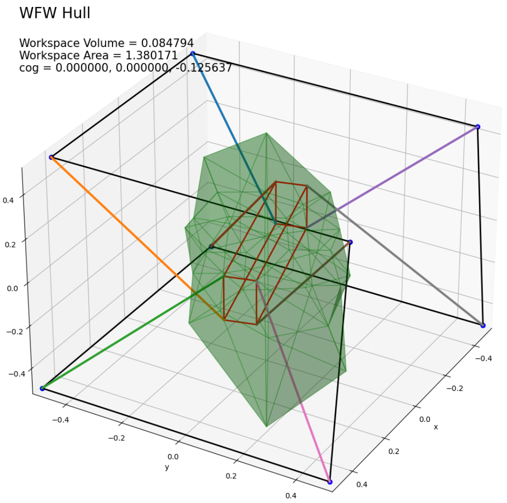

WFW Hull
Workspace Volume = 0.084794
Workspace Area = 1.380171
cog = 0.000000, 0.000000, -0.125637
x
y

Figure 8: Test case 1

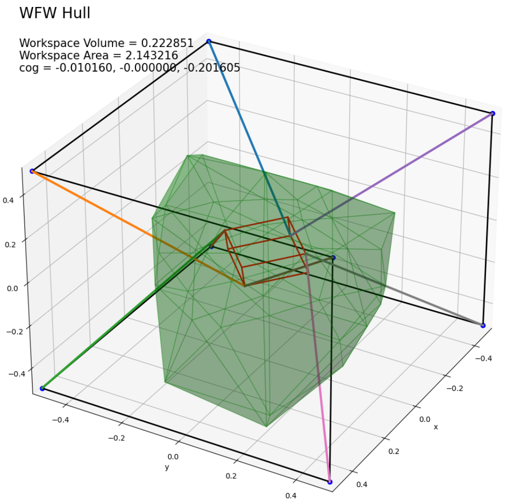

WFW Hull
Workspace Volume = 0.222851
Workspace Area = 2.143216
cog = -0.010160, -0.000000, -0.201605
x
y

Figure 9: Test case 2

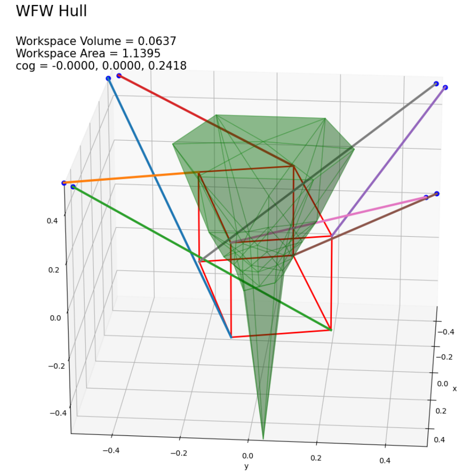


Figure 10: Test case 3

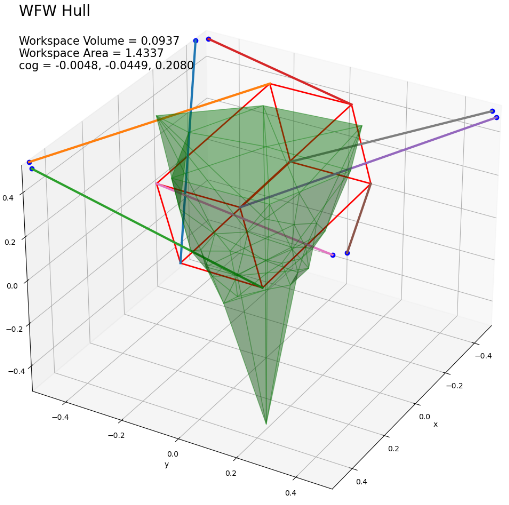


Figure 11: Test case 4

solvable for cables that are parallel or very close to being parallel. All algorithms were changed in such a way that cables that are very close to being parallel are not considered to be colliding.

2.4.2 Observation 2

In figure 12 it can be seen that Nguyens method is substantially larger than the benchmark for 26 vertices. Inspecting the cable-cable collisions along the path to one of these vertices, it appears the method of Nguyen detects a collision further away, because it only detects a collision at sign change. The cables with a physical width implemented in the model collide earlier for this configuration. This difference causes a significant change in size of the hull for this test case.

2.4.3 Observation 3

Nguyens Method with Cable Distance Check is presenting a hull that is almost equal to the benchmark hull. So it is confirmed that Nguyens numerical method is not taking the cable width into the equation and for that reason presents a larger hull. Perreaults method without sign check performs well too.

2.4.4 Observation 4

Perreaults method with sign check comes close to the benchmark as well, but shows some large dents in the hull for 2 vertices. These are caused by a false collision detection. A cable pair changes sign and has its closest distance n_r within its cable lengths, but the cables are still far apart from each other.

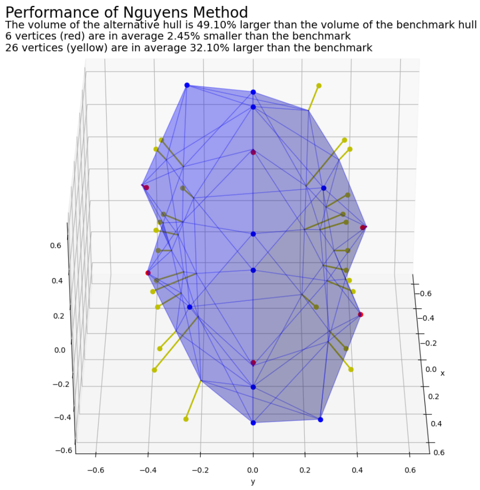


Figure 12: The performance of Nguyens method with respect to the benchmark for test case 1

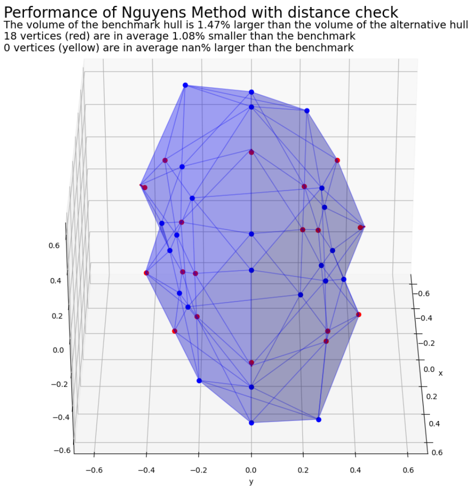


Figure 13: The performance of Nguyens method in combination with a distance check with respect to the benchmark for test case 1

2.5 Test case 2

For the lay-out in figure 9 the different algorithms are compared with the benchmark in figure 16, 17, 18 and 19.

2.5.1 Observation 1

The observed test case doesn't have any cable-cable collisions in the searched sphere around the middle point. All four methods either do not detect collisions or detect a few on the outsides of the searched sphere, which only decreases the workspace by a maximum of 1.5%. From this test case no further conclusions can be drawn.

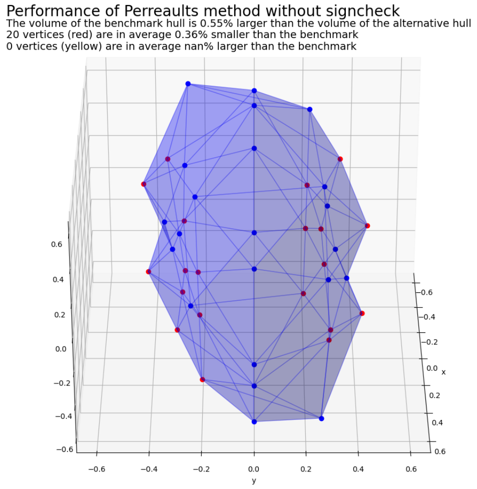


Figure 14: The performance of Perreaults Method without sign check with respect to the benchmark for test case 1

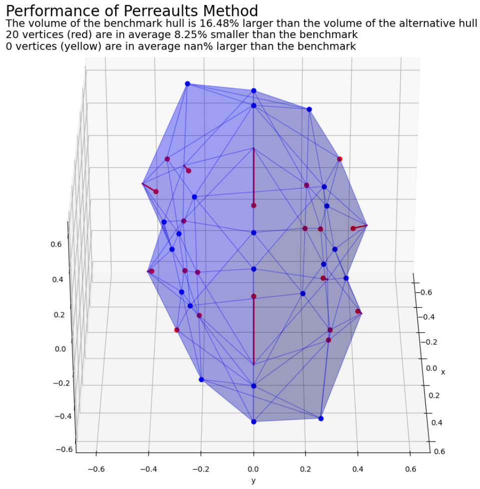


Figure 15: The performance of Perreaults Method with respect to the benchmark for test case 1

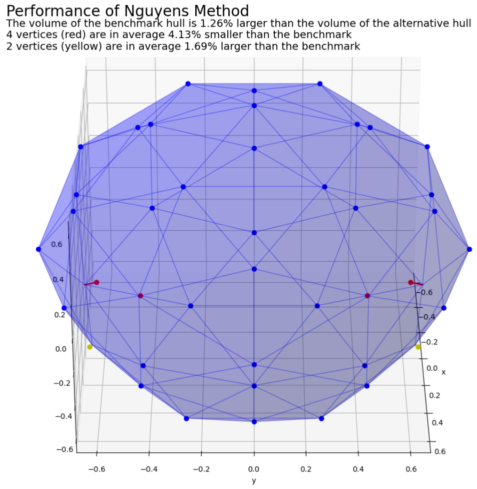


Figure 16: The performance of Nguyens method with respect to the benchmark for test case 2

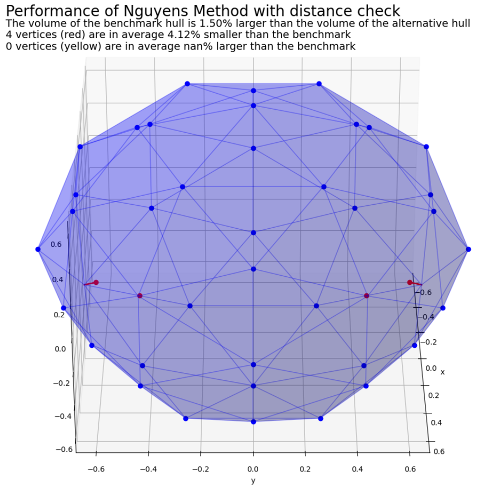


Figure 17: The performance of Nguyens method in combination with a distance check with respect to the benchmark for test case 2

2.6 Test case 3

For the lay-out in figure 10 the different algorithms are compared with the benchmark in figure 20, 21, 22 and 23.

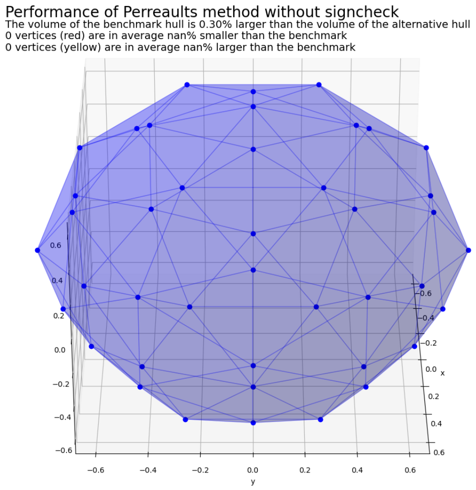


Figure 18: The performance of Perreaults Method without sign check with respect to the benchmark for test case 2

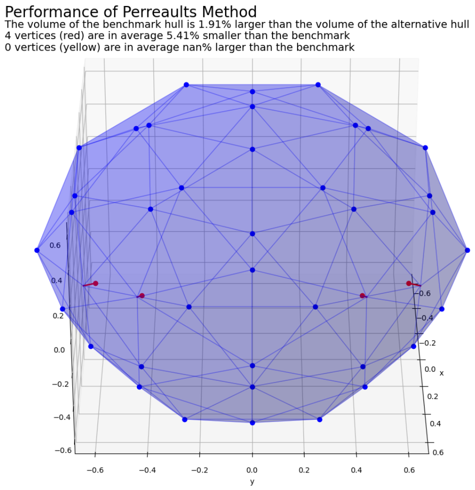


Figure 19: The performance of Perreaults Method with respect to the benchmark for test case 2

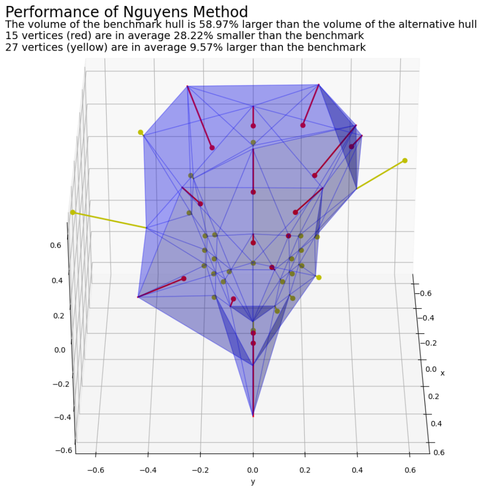


Figure 20: The performance of Nguyens method with respect to the benchmark for test case 3

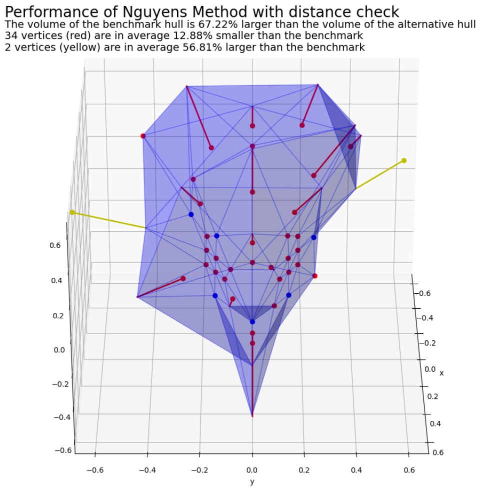


Figure 21: The performance of Nguyens method in combination with a distance check with respect to the benchmark for test case 3

2.6.1 Observation 1

As Perreaults Numerical method without sign check and with the line search algorithm was used, it appeared that for many vertices, there were no cable-collisions detected whatsoever, because the hull is 280 % larger than the benchmark hull. When the line search method is used, points in space are sampled along a line from the center outwards. The distance between the sampled

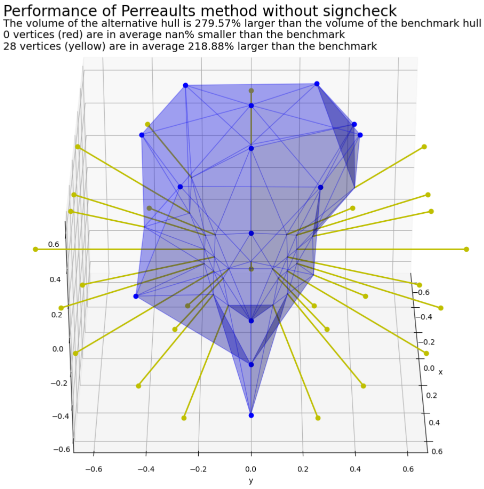


Figure 22: The performance of Perreaults Method without sign check with respect to the benchmark for test case 3

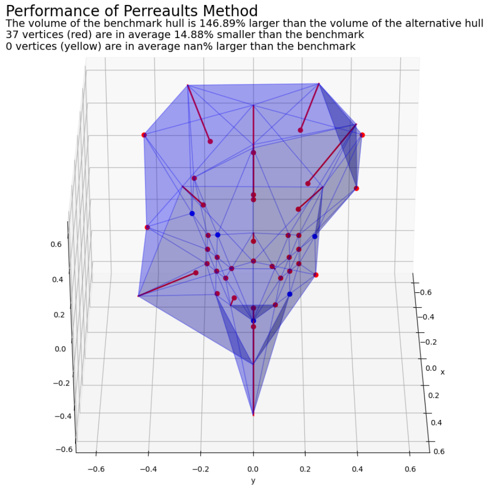


Figure 23: The performance of Perreaults Method with respect to the benchmark for test case 3

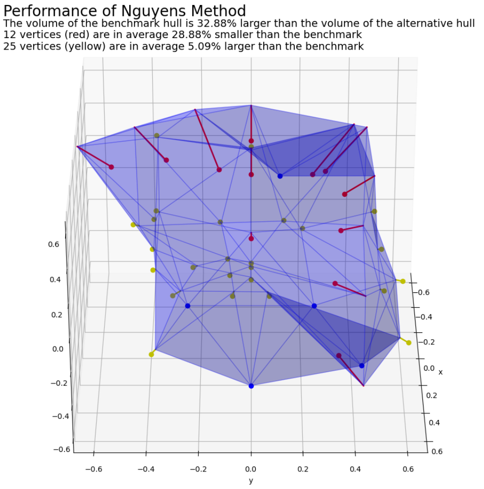


Figure 24: The performance of Nguyens method with respect to the benchmark for test case 4

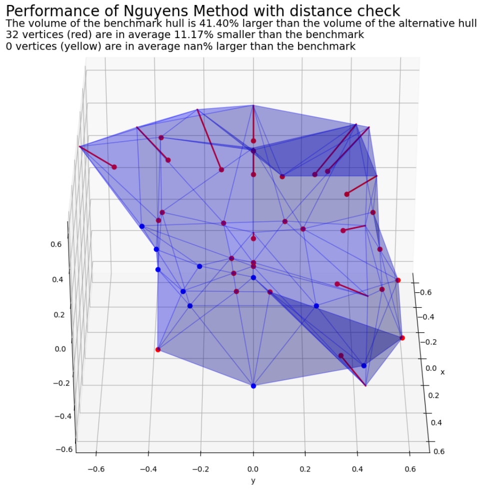


Figure 25: The performance of Nguyens method in combination with a distance check with respect to the benchmark for test case 4

points is large and in this case only at the points itself it is checked whether there is a collision. So all the collisions that would happen when the platform would move outward along that line are missed. Only by luck the line search algorithm evaluates a point at which a collision happens. So in order to have a functioning cable-cable collision detection, there are two options: either sample points in space regularly and finely, or use a method that can reason whether a collision

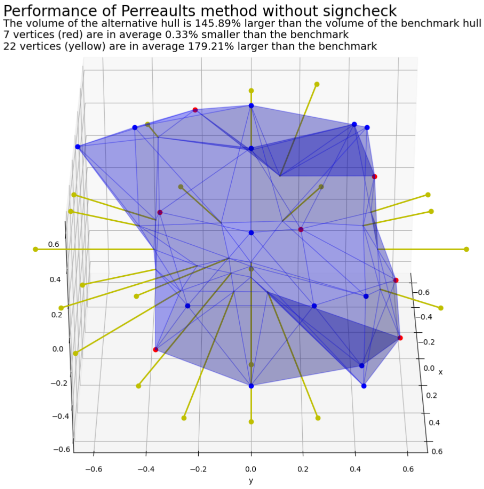


Figure 26: The performance of Perreaults Method without sign check with respect to the benchmark for test case 4

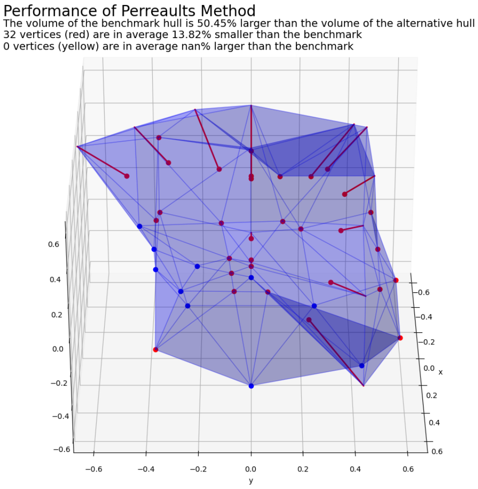


Figure 27: The performance of Perreaults Method with respect to the benchmark for test case 4

has happened *between* points.

2.6.2 Observation 2

Both Nguyen methods in figure 20 and 21 are substantially smaller for 15 vertices. Inspection of the model shows that no real collision takes place, but the sign between the cables does change. The orientation of the cables is unfortunate, because in case of sign changes, it seems like the cables collide after projecting them onto each other, even though they don't.

The hull of both Nguyen methods is also a lot larger at two vertices. For this specific cable-layout, the cables touch at one position along the vertex path without changing their orientation with respect to each other. So a sign change does not occur. This event happened at the proximal endpoint of one of the cables.

2.6.3 Observation 3

When comparing figure 20 and 21 it can be seen another time that the cable distance check decreases the hull of Nguyens method at 25 vertices with a small amount. So a collision is detected a bit earlier when the physical width of the cable is taken into account.

2.7 Test case 4

For the lay-out in figure 11 the different algorithms are compared with the benchmark in figure 24, 25, 26 and 15.

2.7.1 Observation 1

As Perreaults Numerical method without sign check and with the line search algorithm was used, it appeared that for many vertices, there were no cable-collisions whatsoever, because the hull is 146 % larger than the benchmark hull. The same behavior was reported for test case 3, observation 1 (2.6.1).

2.7.2 Observation 2

When using the distance check for Nguyens method, the benchmark is 41% larger instead of just 33% for Nguyens method without distance check. The advantages of Nguyens method with distance check are that there are no detections anymore that are too late and the method is more conservative.

2.7.3 Observation 3

Perreaults Method is even more conservative than Nguyens method with distance check. For the same vertices for which there existed early detections already for both Nguyen methods, Perreaults Method shows an even earlier detection.

2.8 Pros and Cons of the Different Methods

Perreaults Method

+ Quick method, because it can be used in combination with the line search method. Less points in space need to be sampled.

+ Takes into account the width of the cables.

– In some cases this method is detecting non-existing collisions so has a smaller hull. This is caused by the fact that the projection method is not flawlessly revealing whether between two sample points the cables have collided within their lengths. This method discovers non-existing collisions at the same vertices as Nguyens methods, but then even sooner. For some vertices this method detected a non-existing collision when Nguyens method wasn't, such event is described in test case 1 observation 4.

Perreaults Numerical Method without sign check without line search

+ Straightforward to implement. At every position just the distance between the cables n_r and its location along the cables n_1 and n_2 have to be calculated.

+ Very accurate, every point is evaluated, the algorithm can't misinterpret the result.

– Slow method. This algorithm only works when the samples lie very close to each other, so a lot of samples have to be taken.

Perreaults Numerical Method without sign check with line search

+ Straightforward to implement. At every position just the distance between the cables n_r and its location along the cables n_1 and n_2 have to be calculated.

+ Quick, only the distance between the cables n_r and its location along the cables n_1 and n_2 have to be calculated at the points in space determined by the line search method.

– Highly inaccurate, because this method samples at less points and does not take into consideration that a collision could take place between these points. A lot of collisions are missed for this reason.

Nguyens Numerical Method

+ Quick method, because it can be used in combination with the line search method. Less points in space need to be sampled.

– In some cases this method is detecting non-existing collisions so has a smaller hull. This is caused by the fact that the projection method is not flawlessly revealing whether between two sample points the cables have collided within their lengths .

– In some rare cases this method doesn't detect cases for which the cables already interfere but a sign change hasn't taken place yet. This is caused by the fact that this method doesn't take into account the width of the physical cables.

– In some more rare cases it doesn't detect a collision, because after the collision the cables move back and do not move through each other. This happened for two vertices in test case 3.

Nguyens Numerical Method with distance check

+ Quick method, because it can be used in combination with the line search method. Less points in space need to be sampled.

+ Takes into account the width of the cables.

– In some cases this method is detecting non-existing collisions so has a smaller hull. This is caused by the fact that the projection method is not flawlessly revealing whether between two sample points the cables have collided within their lengths.

– In some more rare cases this method doesn't detect a collision, because afterwards the cables move back and do not move through each other. This happened for two vertices in test case 3.

2.9 General Recommendations

- The starting point for the Hull method has to be inside the collision-free workspace, otherwise the signs that serve as a benchmark are not correct.

- Perreaults method without Line Search and without sign check can be used as a very accurate benchmark.

- Nguyens method with distance check is the recommended method for calculating the cable-cable collision hull. It is quick and it represents the benchmark hull more accurately than Nguyens method by being more conservative at vertices where the width of the physical cable plays a role. Perreaults method is a good second option, but is more conservative in cases where it shouldn't.

- The benchmark method doesn't work anymore when angles are incorporated. There are giant leaps in rotation steps and the benchmark method cannot know what happens in between two rotation positions.

2.10 ALternative Ideas

2.11 Releasing Cables

This could be an extra strategy to research. It is proposed in[20].

3 Cable-Platform Collision

3.1 Method selection

Three promising methods from literature were found for cable-object collision:

1. The first method is introduced by Nguyen[19]. The method approximates the mobile platform with triangles. For every cable it is tested whether there exists a collision with any triangle, except for the triangles that are attached to the distal endpoint of that particular cable. This method is:

   + easy to implement.

   − quite a computationally expensive method when the amount of cables and triangles is high.

2. The second method is introduced by Nguyen as well[19]. To explain this method figure 28 is referenced to. For each distal endpoint B_i the closest vertices of the platform are selected, which are D_{ik}, $D_{i(k+1)}$ and $D_{iN_{Bi}}$ in figure 28. Together with the closest vertices a distal endpoint forms a convex cone in which the platform lies. Of a point M inside the cone, all its signs with respect to the faces of the cone are evaluated with the following formula:

   $$S_{Bi}(k) = \text{sign}\left(\left(\overrightarrow{B_iD}_{ik} \times \overrightarrow{B_iD}_{i(k+1)}\right)^T \cdot \overrightarrow{D_{ik}M}\right). \tag{6}$$

   When all the signs of the proximal endpoint of a cable with respect to the sides of the cone match with the signs of the point inside the cone, the cable collides. When one of the signs is zero, the cable collides as well. This method is:

   + computationally efficient because of initialization with the cone. A approximation is presented in Nguyen's paper as well.

   − only working when the platform has a convex shape.

   − difficult to implement, because points for the convex shape still have to be manually selected.

3. The third method is used by Perreault[22]. It uses the cable-cable collision algorithm, while treating the edges of the platform as cables. This method is:

   + easy to implement.

   − quite a computationally expensive method when the amount of cables and edges is high.

3.2 Implementation

The method that is implemented is the second method of Nguyen. For the research purposes of this report the platform is simplified into a convex cuboid, so the closest vertices can be selected manually. The vertices keep their nearest neighbouring vertices when the dimensions of the cuboid are altered during an optimization. In python first a function is written that defines the signs of all three sides around a distal endpoint with respect to a point within the platform. In a second function this list of signs is then compared with the signs of the proximal endpoints with

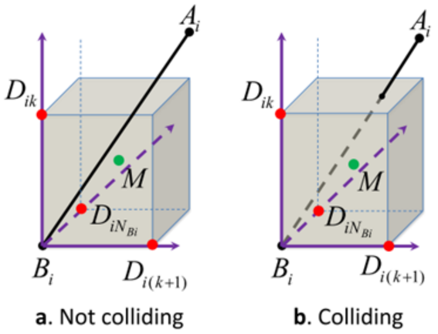


Figure 28: Visualisation of the detection of a cable-collision, taken from[19].

respect to all the sides. When the signs of a proximal endpoint with respect to the three sides around its distal endpoint do not exactly match with the signs for a point within the platform with respect to the three sides, a cable is considered to not collide with the platform. When no cable collides with the platform, a pose is considered to be part of the workspace.

3.3 Verification of the Method

To check whether the cable-platform collision detection algorithm is able to assist in creating cable-platform collision free workspaces is verified with two tests. In the first test, the workspace of a trivial CDPR lay-out will be created. This test will reveal whether the method works correctly. In the second test the hull method was used with either a line search method or a regular grid of samples to check whether the line-search method misses certain cable-platform collisions.

3.3.1 Trivial lay-out test

For this test a lay-out was used for which it is easy to determine the cable-platform collision free workspace without the use of a computer. The results of the cable- platform collision detection algorithm are compared with the findings from common sense. The lay-out used is shown in figure 29. The platform in this fixed orientation should be able to move freely up and down without collisions as long as the cables do not run above or below the platform. This is not the case within the frame. When moving along the x and y direction, the platform is colliding with a cable as soon as the side of the platform touches the side of the frame. So the workspace should be a rectangle, unbounded in z direction and in x and y direction bounded by the distance of the side of the platform to the side of the frame. This was confirmed by the algorithm and the result is shown in figure 30 and 31.

3.4 Benchmark test

To test whether the line search algorithm might oversee any collisions it was applied to the different test cases introduced in section 2.3. As a benchmark the algorithm was applied to poses sampled with a fine stepsie of 0.05, significantly smaller than the smallest platform dimension 0.1 belonging to test case 2. For none of the test cases the cable-platform detection algorithm showed any deviation from the benchmark. However, at some points the line search algorithm performed a little bit better than the benchmark. This is caused by the fact that the line search

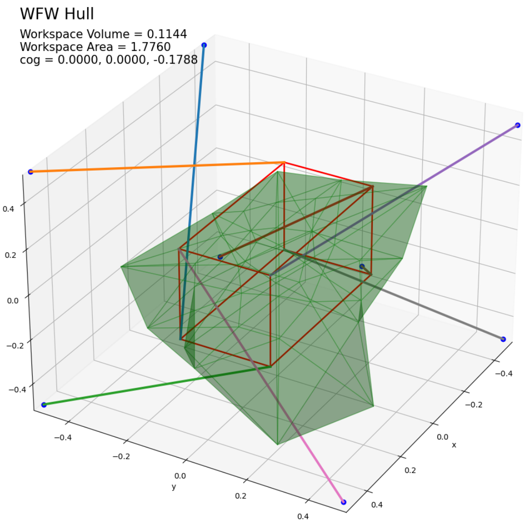


Figure 29: A trivial CDPR lay-out and its WFW to test the cable-platform collision detection performance.

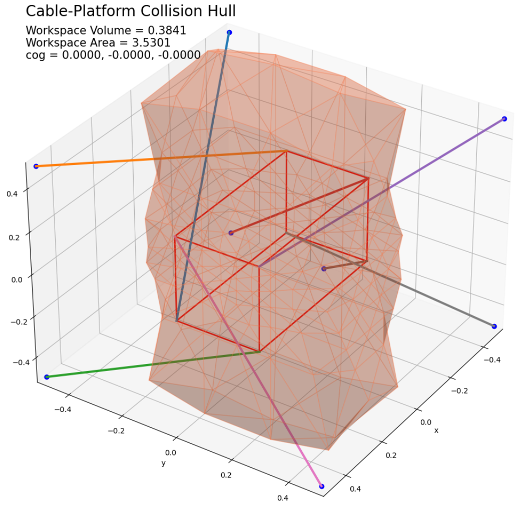


Figure 30: Trivial test case to test the cable-platform collision free workspace

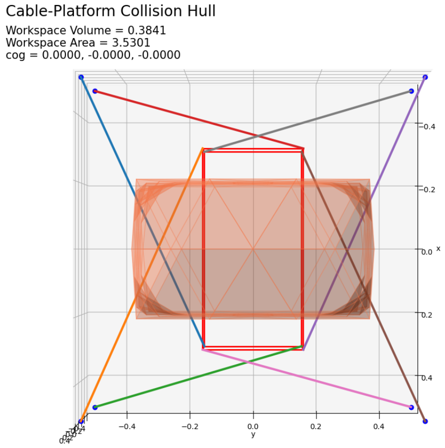


Figure 31: Trivial test case to test the cable-platform collision free workspace - top view

algorithm, when converging to the right position, is converging with a very small error. So this difference is caused by the resolution of the sample step size and the allowed convergence error of the line search algorithm. The different cable-platform collision free work spaces and their comparison with the benchmarks are shown in figure 32 to 35.

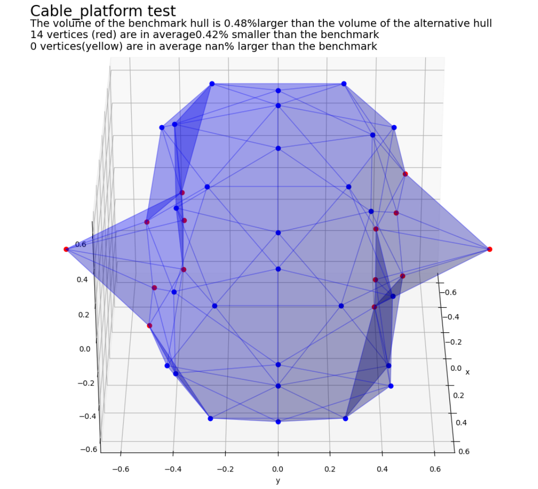


Figure 32: The cable-platform collision free workspace of test case 1

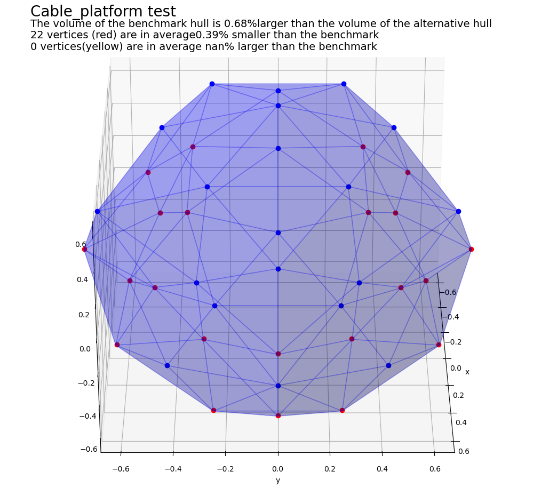


Figure 33: The cable-platform collision free workspace of test case 2

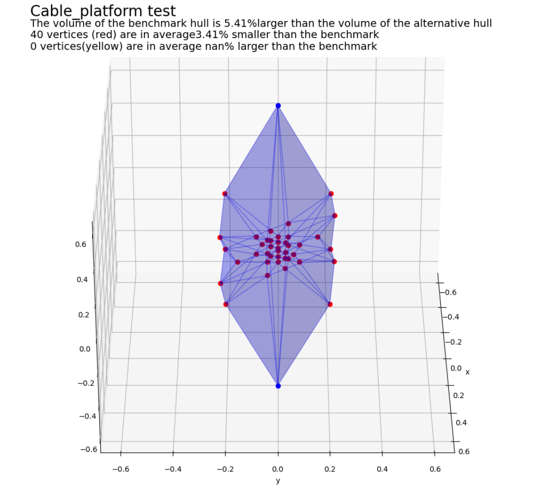


Figure 34: The cable-platform collision free workspace of test case 3

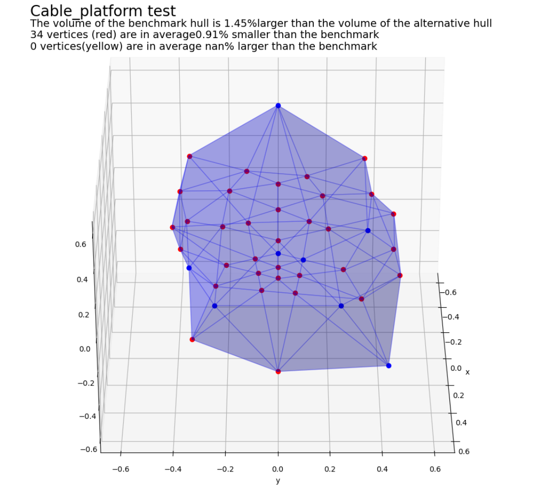


Figure 35: The cable-platform collision free workspace of test case 4

3.5 Pitfalls

When using the Hull method for determining the cable-platform collision free workspace, no assumptions can be made about the space between two evaluations along the line of the midpoint to a vertex. When the steps of the Line Search algorithm are too large, a collision could be missed.

4 Cable-Statue Interference

An algorithm based on the algorithm for cable-cable collision detection will be used for cable-statue collision, since the statue can assumed to be a cylinder and a cylinder can be seen as a thick cable. The same method is applied in[7]. The cable-statue collision detection method has another feature added to check whether a cable crosses the bottom or top of the cylinder. This feature is added because the cylinder of the statue is too thick to be modelled merely as a cable. There are some events possible for which a cable and the statue have their closest points closer to each other than the minimum distance, although this point lies outside of their lengths. In this case, the cable could still cross the statue wall through the top or bottom.

Another condition is added to the code that can be used later on for platform-statue collision as well: for a parallel cable (or edge of the platform) it is only evaluated whether the cable isn't closer to the statue than the minimum distance and not where within their lengths the closest point with respect to each other is. The parallel cable/edge is only allowed within the minimum distance of the statue when it is fully above or below the statue. The cable-cable algorithm originally always interprets parallel cables as not touching.

An alternative method dealing with the a cable-cylinder collision is the method presented by Martin[17]. In his work, Martin provides a method of calculating the volume of all the platform positions for which a cable interferes with the statue. Martin provides an analytical method to find all the positions for collision for a certain orientation of the platform. The advantage is that this only has to be done once for a platform orientation. Then the cable-statue collision workspace is every position outside of this region. This method is especially well suited for a cylinder unable to move, because then the region of interference only has to be calculated once. The method presented in this report is numerically evaluating whether a collision occurs for every position and orientation. The method used in this report is sufficient when using the hull method to calculate the workspace. When combining the hull method and Martins collision volume, it would have to be evaluated whether every position/ orientation belongs to one of the interference regions.

An alternative idea that hasn't been used is introduced by Blanchet **??**. A cable and its shifted version form a plane and it is checked whether this plane intersects with the statue.

4.1 Testing the performance

The performance was tested in two ways: first the method with sign check and line search was evaluated to check whether the hull is calculated correctly and is limited by collisions. The second test is done to evaluate what the performance of the cable-statue collision algorithms with respect to the benchmark algorithm is. The two tested algorithms both use the line search algorithm, but one works with a signcheck and one without. They were tested to find out whether the sign check is of added value. The method without sign check is quicker but it might miss some collisions as cables can move entirely through the statue when the steps of the line search are large enough. The functionality entirely depends on the inital step size of the line search and the ratio of CDPR platform, CDPR frame and statue. For the examples in this section a statue is modelled as a cylinder with a height of 1, a radius of 0.1 and the center of the bottom circle at $[-0.5, 0, -0.5]$.

4.1.1 Verification of the method

To investigate whether the hull is correctly limited by cable-statue collisions, the cable-statue collision hull was calculated for the simpler case introduced in section 3.3.1. The resulting hull with the platform in the middle of the hull is shown in figure 36. The platform at one of the vertices of the hull where a cable-statue collision takes place, is shown in figure 37. The shape of the hull is as expected: no collisions take place when the platform is moving in the direction opposite of the statue. There is little room for the platform to move in the direction of the statue and when the platform moves to far to the sides, a cables is colliding with the statue too. This evaluation of the hull shows that the workspace hull is correctly limtited by cable-statue collisions when using the detection algorithm.

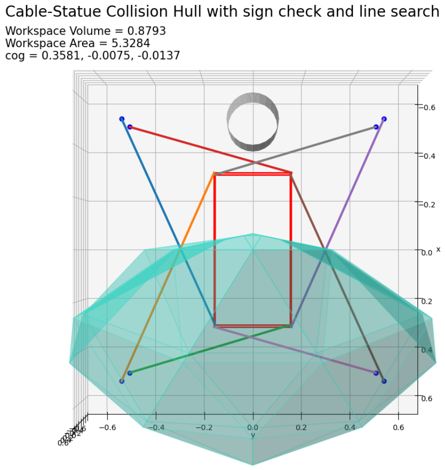


Figure 36: Trivial test case to verify the cable-statue collision free workspace with the platform in the starting position of the hull.

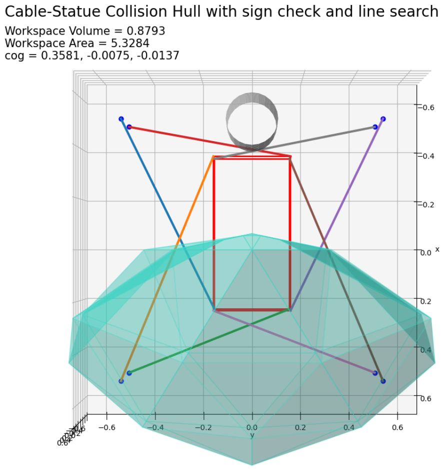


Figure 37: Trivial test case to test the cable-statue collision free workspace with the platform on a colliding position

4.1.2 Sign check and line search test

For this test the cable-statue collision hulls of the four test cases from section 2.3 are calculated. This is done for three algorithms: an algorithm with sign check and line search and an algorithm with the line search but without sign check are compared with the benchmark. The benchmark method does not use the line search method but small incremental steps and only uses an algorithm that detects whether cable and statue are too close at a position instead of an additional sign change detection. The comparisons with the benchmarks are shown in figure 38 to 45. The following conclusions can be drawn from the comparisons:

- The methods are not showing any false detections. The red vertices are caused by the fact that the line search method can be more accurate than fine regular sampling.
- In figure 39, 43 and 45 it can be seen that some cable-statue collisions remain undetected when using the algorithm without sign check. The algorithm with sign check only misses a collision for test case 4. So the sign check algorithm is more accurate.
- The missed collision in test case 4 for the method with sign check, shown in figure 44, is caused by a cable that interferes, but it's closest point to the statue is outside the cable length. The distal endpoint of the cable is still within the statue and will be filtered out using the platform-statue interference.

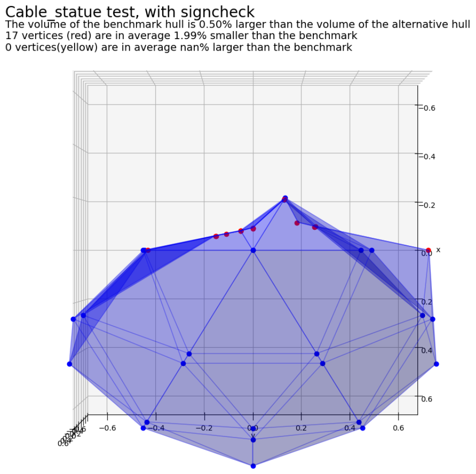


Figure 38: Cable-Statue collision free workspace for test case 1: a hull computed with signcheck is compared with the benchmark hull

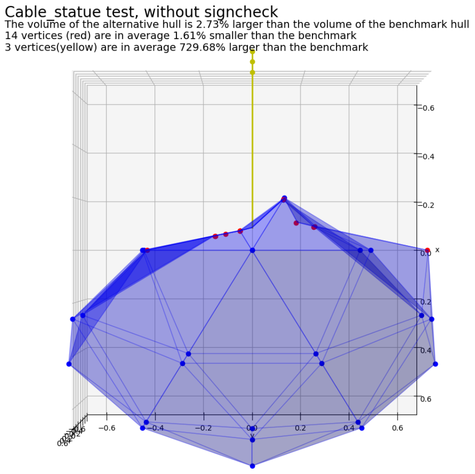


Figure 39: Cable-Statue collision free workspace for test case 1: a hull computed without signcheck is compared with the benchmark hull

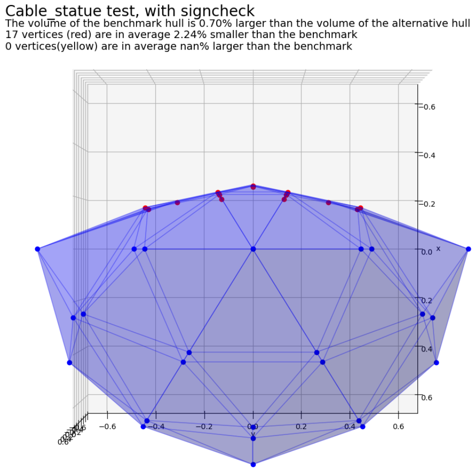


Figure 40: Cable-Statue collision free workspace for test case 2: a hull computed with signcheck is compared with the benchmark hull

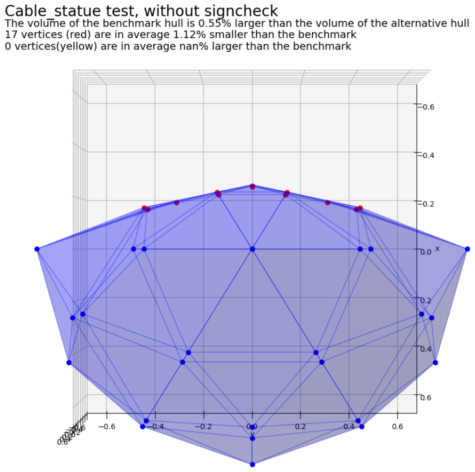
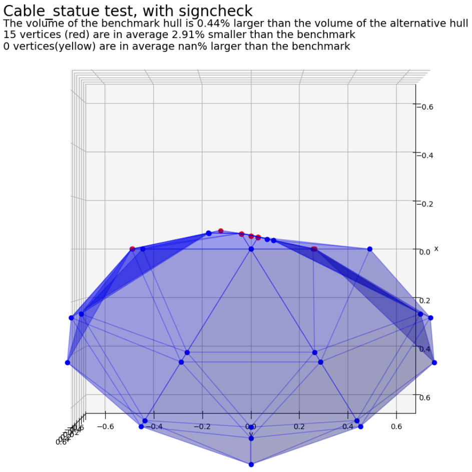


Figure 41: Cable-Statue collision free workspace for test case 2: a hull computed without signcheck is compared with the benchmark hull

4.2 Pitfalls

The closest point between the cable axis and the statue axis can be within the length of the statue but outside of the cable length. The cable can then still collide with the statue, because



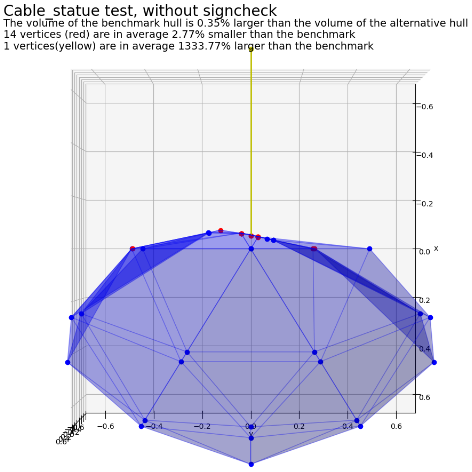
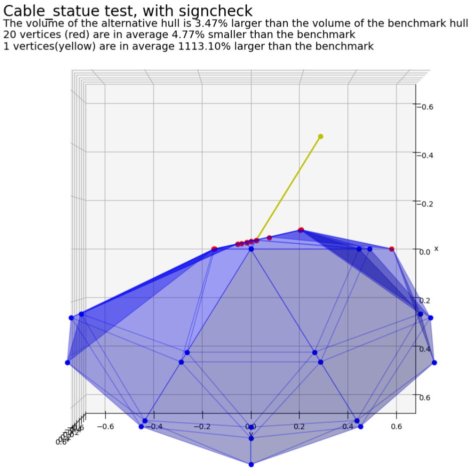


Figure 42: Cable-Statue collision free workspace for test case 3: a hull computed with signcheck is compared with the benchmark hull

Figure 43: Cable-Statue collision free workspace for test case 3: a hull computed without signcheck is compared with the benchmark hull



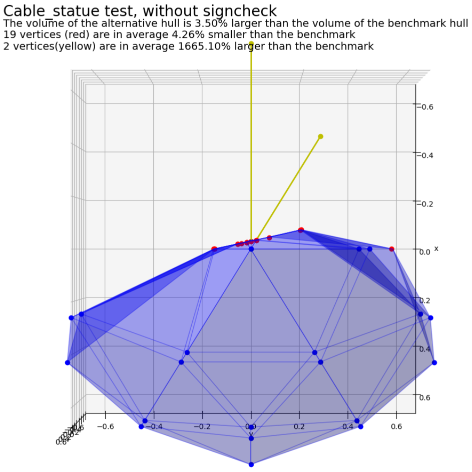


Figure 44: Cable-Statue collision free workspace for test case 4: a hull computed with signcheck is compared with the benchmark hull

Figure 45: Cable-Statue collision free workspace for test case 4: a hull computed without signcheck is compared with the benchmark hull

of the thickness of the statue. When such a collision happens, the distal endpoint of the cable is within the statue and so is the platform. For this reason, the missed detection will be accounted for during the platform-statue collision detection.

5 Platform-Statue Interference

In literature different methods are presented for platform obstacle-collision. Blanchet[4] and[3] are proposing to check for the collision of the swept space of the platform and an obstacle. Nguyen[?] proposes AABB or OOBB trees to detect whether the platform is colliding with an obstacle at a certain position.

AABB or OOBB trees are very accurate, but unnecessary for the structure as simple as the cuboid used in this research. The swept space of the platform is very accurate to detect collisions between sampling points of the line search algorithm of the hull method. It is comparable with the sign check method used for cable-cable collision detection. This method could also be used when the cables are replaced by the edges of the platform and the centerline of the statue. To make sure the platform-statue collision detection does not cost unnecessary computing power, no sign check or swept space will be used. When the platform has collided with the statue between two points at which there is no collision, a cable-statue interference will be detected in between, because this algorithm includes a sign check.

In this research, three different checks ensure the detection of a possible collision for the platform and statue:

1. First, it is checked whether any of the corner points lie inside of the statue.

2. Secondly, it is checked whether any of the edges interfere with the statue. This is done with the algorithm used for cable-statue interference, but without using the sign check.

3. Thirdly it is checked whether the statue resides within the platform without touching any edges or corners. This is done by defining another line that forms a diagonal through the center of the platform. This line cannot interfere with the statue either.

5.1 Testing the performance

The performance of the method was tested in two ways: first the method was evaluated by visually inspecting the plotted hulls to check whether the hull is calculated correctly and is limited by collisions. The second test is conducted to evaluate the performance of the platform-statue collision algorithms with respect to the benchmark algorithm. For the examples in this section a statue is modelled as a cylinder with a height of 1, a radius of 0.1 and the center of the bottom circle at $[-0.5, 0, -0.5]$.

5.1.1 Verification of the method

To investigate whether the hull is correctly limited by cable-statue collisions, the cable-statue collision hull was calculated for the simpler case introduced in section 3.3.1, but then with a larger platform with dimensions $0.8x0.4x0.6$. The statue is placed with its longitudinal axis on the origin, so inside of the platform. Now the platform's movement is restricted in the x and y direction and unrestricted in the z direction. The hull was calculated without the line search method and without the extra diagonal edge within the platform. The result with the platform and the statue at the origin and in the middle of the hull is shown in figure 46 and it's top view in 47. The shape of the hull is as expected: no collisions take place when the platform is moving within $[-0.3, 0.3]$ in the x direction and $[-0.1, 0.1]$ in the y direction. This evaluation of the hull shows that the workspace hull is correctly limited by platform-statue collisions when using the detection algorithm. After this evaluation the extra diagonal edge within the platform was added.

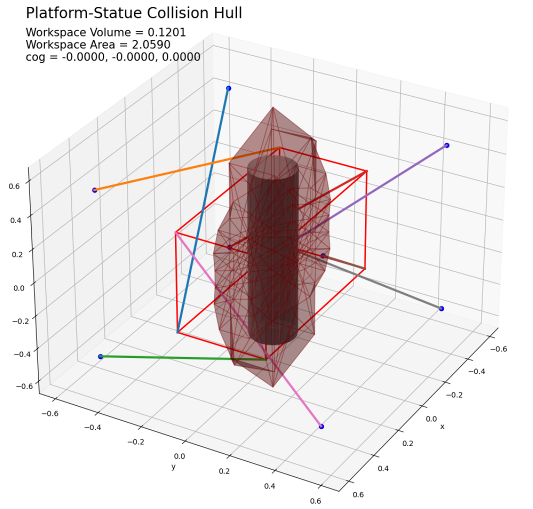


Figure 46: Trivial test case to verify the platform-statue collision free workspace with the statue in the platform

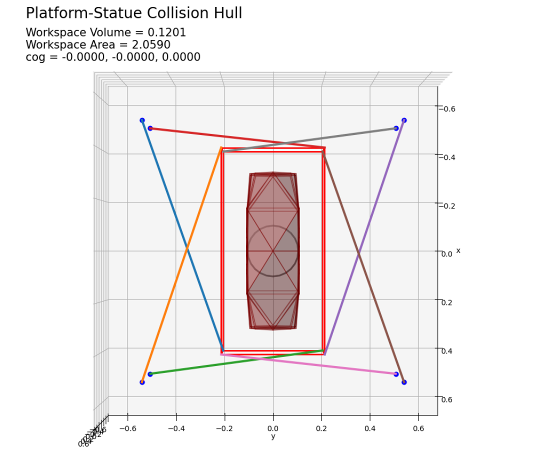


Figure 47: Trivial test case to test the cable-statue collision free workspace with the statue in the platform

5.1.2 Evaluating the performance

The benchmark is calculated with the same detection method, but the hull method is used without line search and with small incremental steps instead. The comparison is made to find out whether collisions are missed when not using a sign check method. The functionality of the line search entirely depends on the inital step size of the line search and the ratio of CDPR platform, CDPR frame and statue. When the ratio is off, the line search could miss a detection. The comparison with the benchmarks is shown in figure 48 to 51. For both test case 1 and 3, the algorithm misses one detection and for test case 4, it misses a detection two times. This detection is missing because the initial step of the linesearch is so large, the new platform position is at the other side of the statue. These missed collisions do not pose a problem since the cable-statue collision detection will detect the platform switching sides with the statue in most cases, because then cables switch sides with the platform too. Whether the two methods offer a workable solution is tested in the next paragraph.

5.1.3 Combining the cable-statue and platform-statue collision detection

To test whether the cable-statue and platform-statue collision detection combined make up for the flaws in both methods, their two hulls are combined and compared with their combined benchmarks. The hulls are combined by choosing for every vertex the one closest to the origin. The comparison between the combined hulls and their combined benchmarks is shown in figure 52 until 55. The difference for one vertex for test case 1 is explained by the fact that a platform-statue collision is missed because the platform has ended up on the other side. This flaw is not fully compensated by the cable-statue-collision because the closest point between the cable and statue is outside of the cable length at first. The flaw in test case 4 is caused b the fact that a cable-statue collision is not detected because a projection of the cable on the statue doesn't collide, but they have collided on the way to this position.

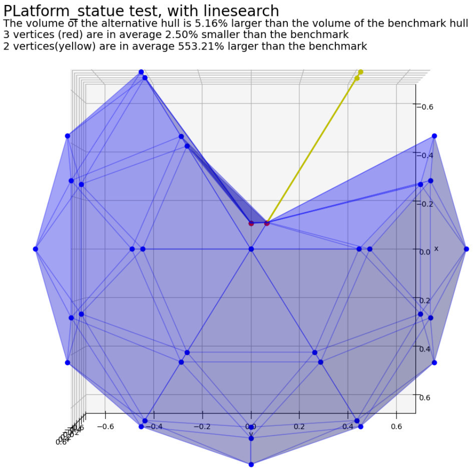


Figure 48: Platform-Statue collision free workspace for test case 1: a hull computed is compared with the benchmark hull

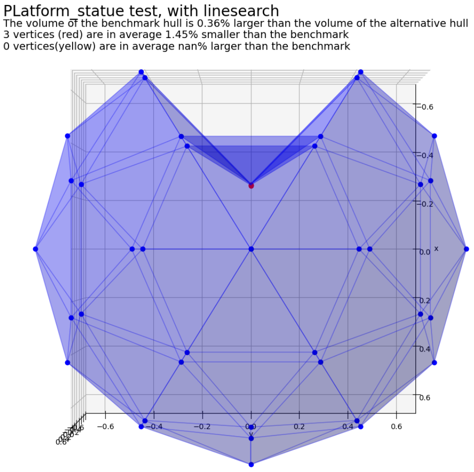


Figure 49: Platform-Statue collision free workspace for test case 2: a hull computed without signcheck is compared with the benchmark hull

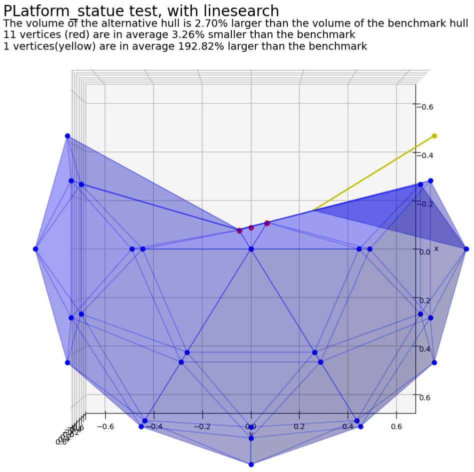


Figure 50: Platform-Statue collision free workspace for test case 3: a hull computed without signcheck is compared with the benchmark hull

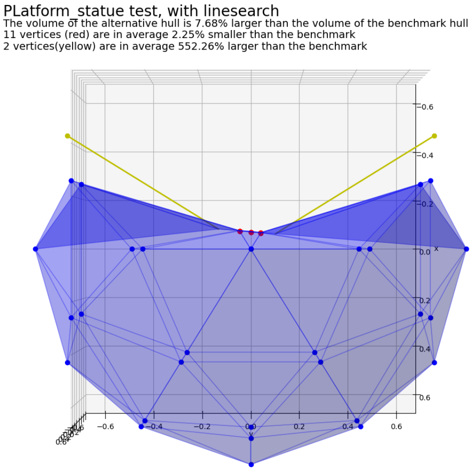


Figure 51: Platform-Statue collision free workspace for test case 4: a hull computed without signcheck is compared with the benchmark hull

5.1.4 Tweaked hull method for platform-statue collision detection

The flaw in test case 1 is solved by changing the hull method for the platform-statue detection. During the first two iterations of the linesearch, there is an equally distributed grid of four points

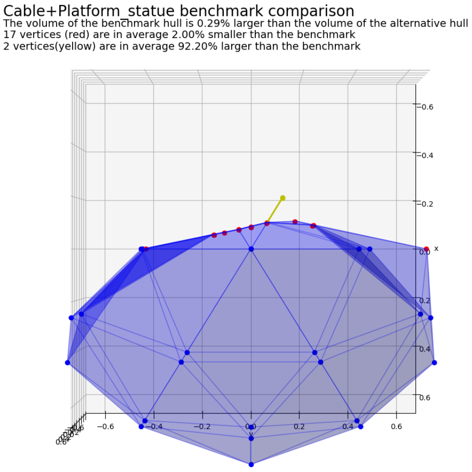


Figure 52: Combined Platform-Statue and platform-statue collision free workspace for test case 1 compared with the benchmark

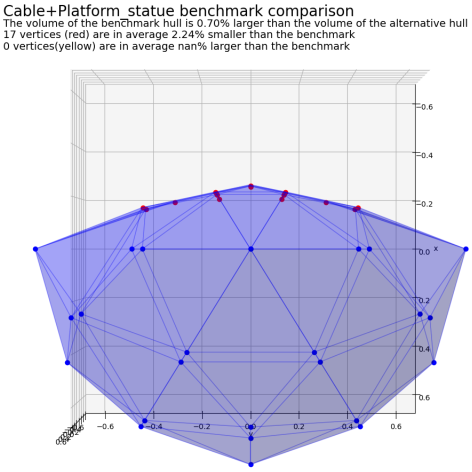


Figure 53: Combined Platform-Statue and platform-statue collision free workspace for test case 2 compared with the benchmark

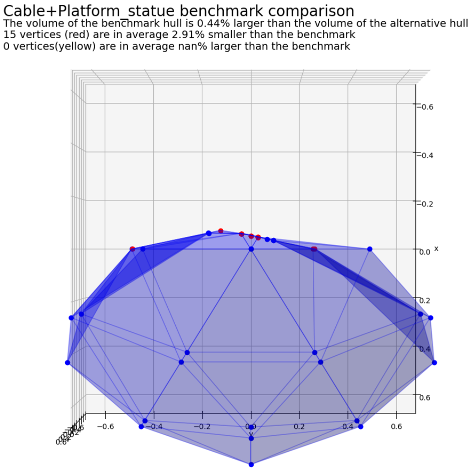


Figure 54: Combined Platform-Statue and platform-statue collision free workspace for test case 3 compared with the benchmark

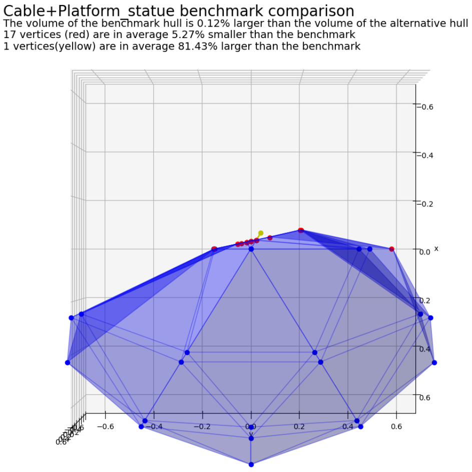


Figure 55: Combined Platform-Statue and platform-statue collision free workspace for test case 4 compared with the benchmark

with the line search point as outer point. All these points have to be collision-free in order for the algorithm to continue with the maximum value.

5.2 Pitfalls & Thoughts

- When the camera is sticking out of the platform, some vertices and edges that relate to the camera position should be added to the platform-statue collision detection as well.
- A possible idea could be a platform that has its distal end points sticking out for more control. In this case the edges should be left out of the collision detection algorithm.
- When the camera is sticking out, a different workspace size has to be created. So the reference workspace that needs to be achieved is depending on the distance between the center of the platform and the camera.
- The cylinder could collide with the platform without detection when it is touching the sides without touching the edges.
- The used method only works when the platform has a convex shape.

6 Total Workspace

6.1 Implementing Rotation

The platform needs to be able to pan, tilt and to rotate around the statue with 180 degrees. To implement the rotations in the model, first the platform rotations have to be transformed to world coordinates. When moving to different positions around the statue, the neutral orientation of the camera is horizontal, while pointing in the direction of the statue z-axis. This means the camera's x and y-axes shift depending on the position with respect to the statue. The tilting motion around the y-axis of the camera is a rotation around the world x and y axes. To test whether rotations of the platform are possible, five different orientations are tested at every platform position:

- The neutral orientation, in platform coordinates: $[0, 0, 0]$.
- Tilting down, in platform coordinates: $[0, -\alpha, 0]$.
- Tilting up, in platform coordinates: $[0, \alpha, 0]$.
- Panning left, in platform coordinates: $[0, 0, -\beta]$.
- Panning right, in platform coordinates: $[0, 0, \beta]$.

The hull method function was changed to be compatible with the rotations. A new position of the platform is only accepted when all the five rotations are possible. This is straightforward for the cable force evaluation, because the cable forces only depend on the position. The cable-cable and cable-statue interference detection, however, depend on the old position as well. This is tackled by first checking whether the new position's neutral orientation is admissible compared to the previous position's neutral orientation. When this is the case, the other orientations of the new position are checked to be admissible by comparing them to the new position's neutral orientation. This can be translated to the way the trajectory is planned. So these calculations verify whether a platform can move from the neutral orientation at one position to the neutral orientation at another position and whether one of the four panning or tilting orientations can be reached from the neutral position.

6.1.1 Relaxed Rotation Requirements

As mentioned in the previous section, it is verified whether the platform can move from the neutral orientation of one position to the neutral orientation of another position. The neutral orientations at different positions are the same in platform coordinates but different in world coordinates. This means that the platform is rotated along the way. Moreover, the platform does not have to be able to pan 180 degrees in every position, but only the amount necessary for scanning under a sufficient angle. In Figure 56, the range for scanning under sufficient angle is visualized for two positions. In Figure 57, different zones for which different rotation around the z-axis is required. The zones are not necessary, since the required range can be calculated for every seperate position. To conclude, the rotational requirements are position dependent and this relaxes the requirements for accepting a position.

The platform is not rotating around the platform base 0 0 0 but on the point on the surface statue it is focusing on, this creates a shift in the position of the centre. This is now not accounted for in the code, the platform turns around its central axis.

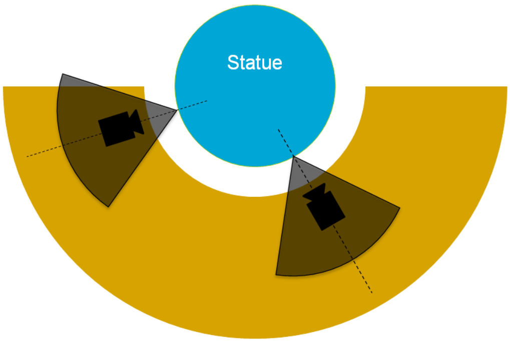


Figure 56: Different camera positions and its required range

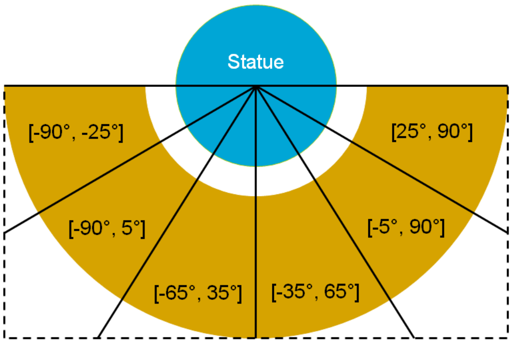


Figure 57: The required camera range visualized with six different sectors

6.2 The total workspace

According to Nguyen[18], when a collision is found when moving to a position X, there is probably not a trajectory at all that is able to move to that position. Following this argumentation, a workspace created with the Hull Method is representative for the actual workspace, or at least a conservative estimation. Nothing can be said, however, about trajectories that are not used for the hull method. For a representative workspace, all different workspace requirements have to be considered.

This is achieved by first calculating the hull for every requirement separately. The same starting point and a standard icosahedron with distance to the vertices of 1 is used for all the requirements. Subsequently, the norm of the distance to the vertices is calculated for every requirement. The requirement that has the smallest norm at a vertex is the limiting requirement and its norm is used for the total workspace. To conclude, the total hull is the minimum of all the separate requirement hulls. The hull can consequently be plotted with information about the limiting requirement of every vertex. In Figure 58, the total hull of test case 2 is plotted with a panning and tilting of 20 degrees. The hull has diamond shape dots at the vertices which are color coded per limiting requirement. Green represents the cable forces and pink the cable-statue collisions. In Figure 59, the total hull of all the benchmark hulls is visualized. The hulls are almost the same and the benchmark hull is even larger because the benchmark stepsize is coarser than the last steps of the line search method.

6.3 Subhulls

The hull method has problems with approaching non-convex surfaces. The required workspace for scanning the statue is half a hollow cylinder, so the side facing the statue is non-convex as well. The workspace shape is made more convex by dividing it into parts. In Figure 60, the workspace is divided into two parts and the shapes are less convex. The only prerequisite is that the platform can reach both hull starting positions from its overall starting point.

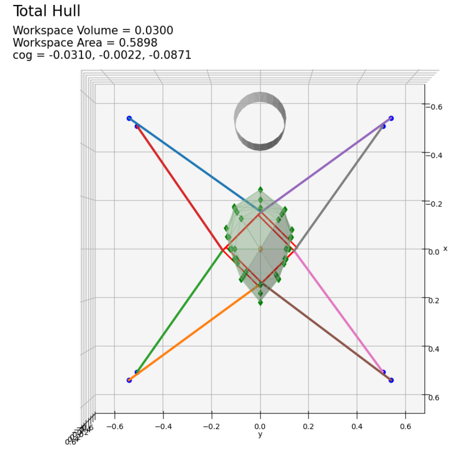


Figure 58: Total Hull of Test Case 2 for a panning and tilting of 20 degrees in both directions

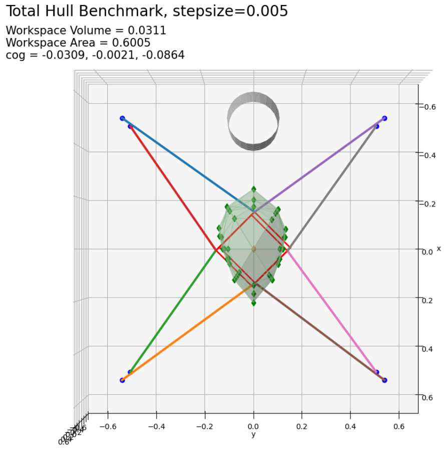


Figure 59: Total Benchmark Hull of Test Case 2 for a panning and tilting of 20 degrees in both directions

The workspace separation method, however, has downsides too: calculating multiple hulls requires more computing time, when making a graphical representation of the workspace, merging the hulls is a challenge and lastly, this changes the trajectory that is verified with the method and it is uncertain whether movement between positions in different hulls is possible. For the purpose of this research, the required workspace is divided into two symmetrical parts just as in Figure 60. In the case of symmetrical CDPR designs, only one side of the workspace has to be calculated.

6.3.1 Remarks

- When a z-axis rotation is evaluated at every new position, how to be sure the rotation is possible in between positions? Answer: you can't be sure and you have to assume the platform is rotating while moving, so it always keeps the statue in sight.
- So far the platform rotates around its COM, but it is intended to let it rotate around the point on the surface of the statue it is scanning.

7 Performance Index

In[22] the complement of the total workspace is used. The coverage of the required workspace is a straightforward performance index. In this chapter, the calculation of the required workspace and its comparison with the actual workspace are described.

7.1 Calculating the Required Workspace

Since the statue is defined as a cylinder, the required workspace is defined as half of a hollow cylinder around the statue. The four boundaries are given by the top of the statue, the bottom

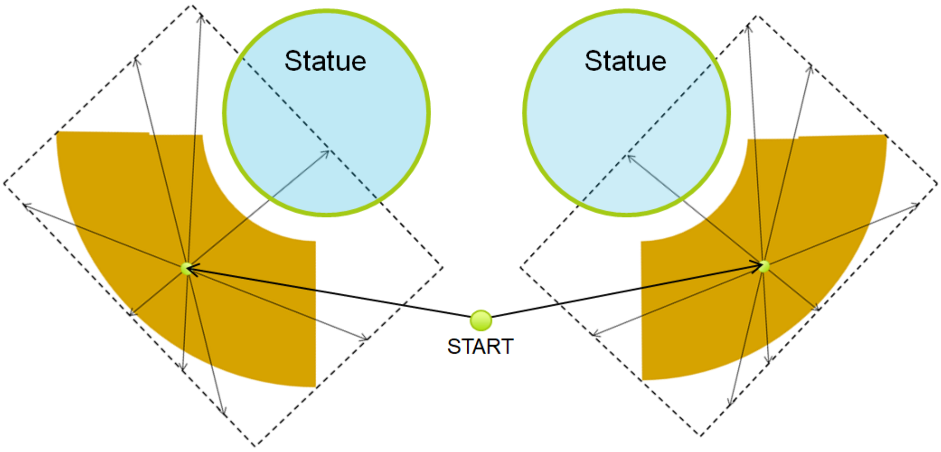
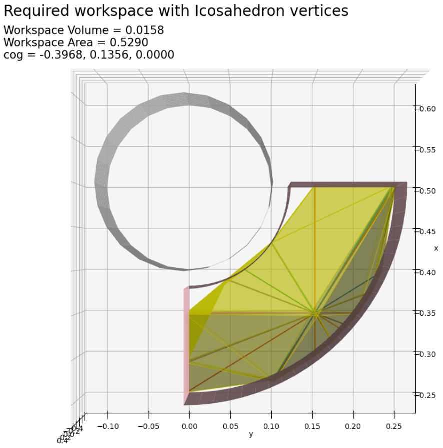


Figure 60: Two hull methods are used to approximate half of a hollow cylinder

of the statue, the minimum scanning depth of the camera and the maximum scanning depth of the camera. To make a statement on which part of the required workspace is occupied by the calculated workspace, the required workspace is approximated with a hull that is starting from the same position as the CDPR workspace hull. This is done by using the hull method. A vertex position is accepted when it lies within the boundaries of the required workspace. A top view of the required workspace and the hull approximation is shown in Figure 61 and 62.



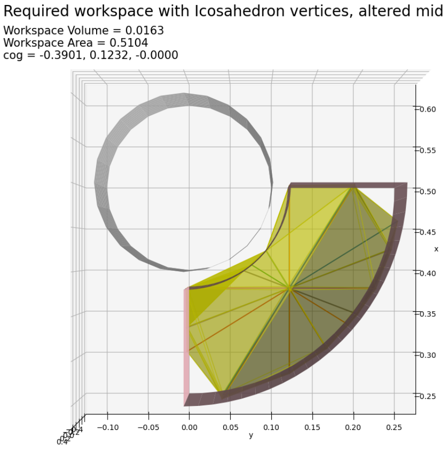


Figure 61: The approximation of the required workspace with the hull method

Figure 62: The approximation of the required workspace with the hull method, while using a different starting position

The different shape of the hulls in the pictures is caused by using a different starting position for the hull method. Because of the top view, it is not fully clear how well the hull method is approximating the required workspace. Figure 63 and 64 show that the 42 vertices of the hull are approximating the required workspace roughly.

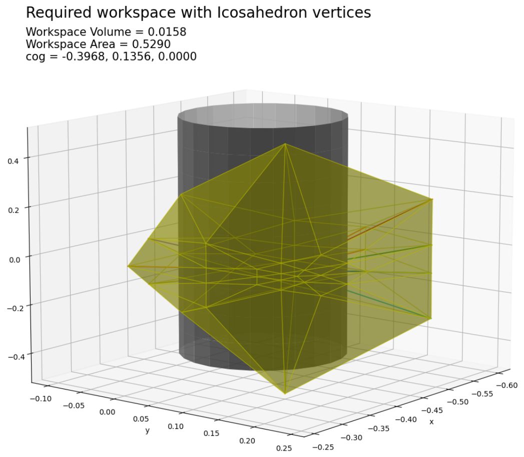


Figure 63: The approximation of the required workspace with the hull method, side view 1

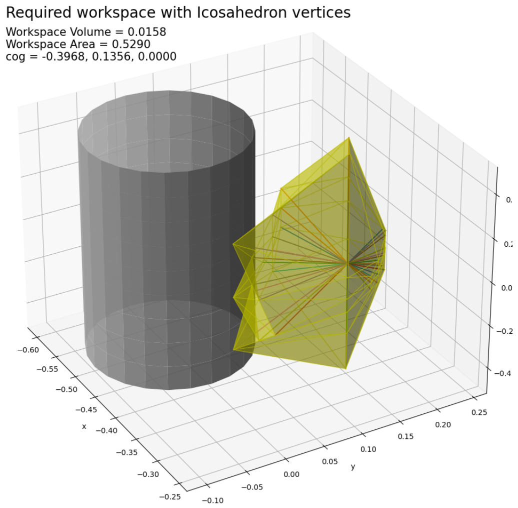


Figure 64: The approximation of the required workspace with the hull method, side view 2

7.2 Comparing the hulls

For comparison, the actual workspace is trimmed with the required workspace boundaries. Both the actual workspace hull and the required workspace hull are created from the same starting point. The trimming is done by calculating the norm of all the vectors of the hull from the starting position to the vertices. These norms are compared with the norms of the required hull. The smallest norm for every vertex is kept. The vertices of this trimmed hull are either on the required workspace boundary or somewhere in the required workspace. The volume of the trimmed hull can then be compared with the volume of the required hull and the coverage can be expressed with a percentage. An example of the actual hull, the required hull and the trimmed hull for an actual hull that is not covering the required workspace fully, are shown in 65, 67 and 69. Figure 66 67 and 70 show the hulls for an actual workspace that does cover the required workspace fully. It is clearly visible, that the trimmed WFW workspace in Figure 69 is not fully covering the required workspace, while the trimmed Cable-Cable workspace in Figure 70 does. By using the Hull volume calculation proposed in[23], the volumes of the required workspace and the trimmed workspace can be calculated, as well as a percentage of coverage of the trimmed workspace.

7.3 Remarks

-

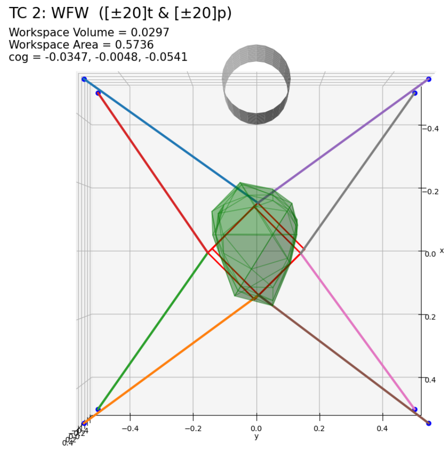


Figure 65: The WFW Hull of test case 2 with 20 degrees tilting and panning in both directions

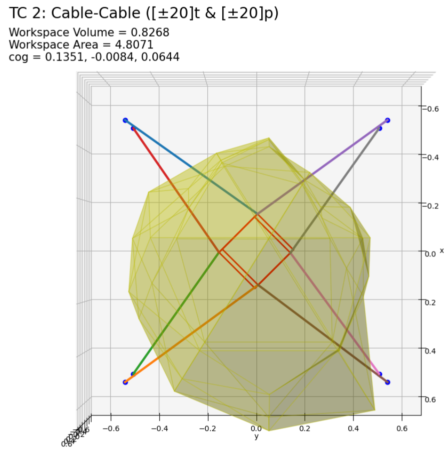


Figure 66: The Cable-Cable Hull of test case 2 with 20 degrees tilting and panning in both directions

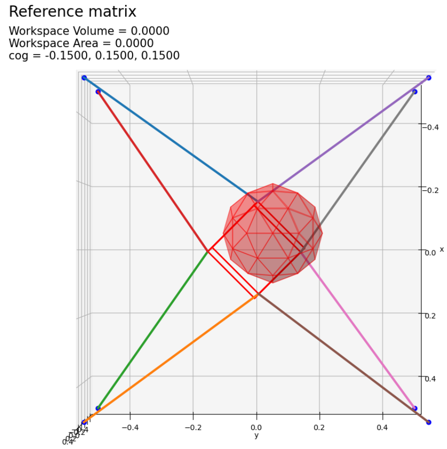


Figure 67: The required workspace

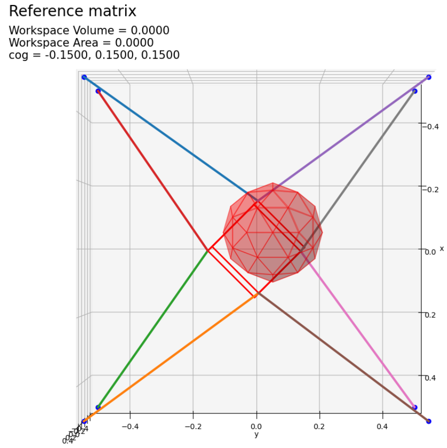


Figure 68: The required workspace, copied for symmetry

8 Shifting to real-life dimensions

In Appendix 11 all the parameters that were used for testing are described. For the testing of designs, however, the real-life dimensions are used. Those real-life dimensions are the following:

- A statue approximated by a cylinder of 2m high and 0.5m diameter on a 1m high pedestal is subject to the scanning as a proof of concept.

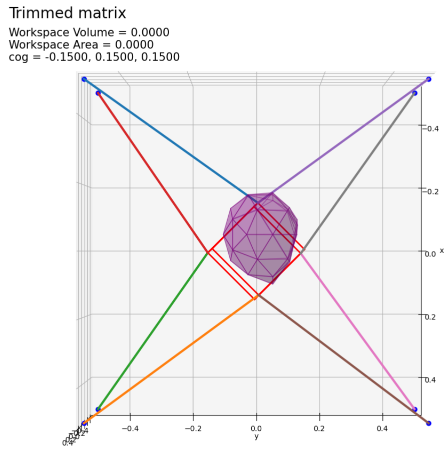


Figure 69: The WFW workspace trimmed with the required workspace

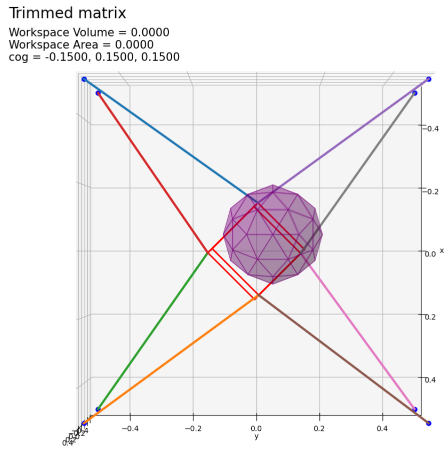


Figure 70: The cable-cable workspace, trimmed with the required workspace

- The exhibition room is 4 meters high and has a width and length of eight meters. This poses a constraint on the size of the robot.

Other dimension that follow from these dimensions:

- Positioning of the origin: at the base of the pedestal
- Cable width: when the width is too low, the benchmark method doesn't work and a safety margin of 5 cm should be present between the cables, so the width is increased to 5 cm.
- Starting point of the platform is the neutral position of the platform, this should be inside the total workspace of the CDPR. Furthermore, it has to be between the two quarter hollow cylinders that pose the required workspace. For now I propose to place it in the middle of the rectangular surface seperating the two identical workspaces.
- Lambda_maxInit is now 6.8 to let the hull theoretically cover all the corners of the frame.
- eps, or the error margin of the hull method
- mid, this is the start of the hull method and should lie somewhere in the required workspace. At this moment, the mid point is calculated by shifting the starting point 45 degrees around the z-axis.
- stepsize_bench is the stepsize for the benchmark method. The diameter of the cables is a reasonable trade-off between misses and computational speed. So 0.05 m
- min_depth or minimal depth of the camera is determined by the working distance of the Dino-Lite AM4113T Digital Microscope[1]. Magnification rates of up to 240x are achievable, but for this experiment a necessary magnification rate of 20x is assumed, which needs a working distance of $48.7mm$. The

- max_depth of the reference workspace is determined by the Specim IQ Spectral Camera[25]. It needs an object distance of $150mm - \infty$. For this research, the maximum object distance will be 500 mm

-

9 Designs

To create a design in Engineering the following steps have to be followed:

1. Problem Definition
2. Requirements
3. Brainstorm/Evaluate, Create New Solution
4. Develop solution
5. Test Solution
6. Feed results to the solution creation step

9.1 The problem

A platform has to rotate around a still-standing statue and has to face it, pan and tilt from as many positions around the statue as possible. The following question arises: *How can the geometrical parameters of a Cable-Driven Parallel Robot be optimally designed to obtain the necessary workspace characteristics for moving around a high 3D object with a large yaw motion of the end-effector?*

9.2 Requirements

The rules for the robot are as follows:

1. The platform/cables/statue are not allowed to touch each other.
2. The platform is connected to the frame only through cables.
3. No extra (passive) actuator is mounted on the platform.
4. Only geometrical parameters are altered.
5. A statue approximated by a cylinder of 2m high and 0.5m diameter on a 0.5m high pedestal is subject to the scanning.
6. The statue should be scanned 180° around the statue.
7. The camera should be able to pan and tilt in the range of ±35°.
8. The frame can be maximum 4 meters high and eight meters wide and deep.
9. Artwork cannot move or rotate.

The exact dimensions for the workspace requirements are shown in Figure 71

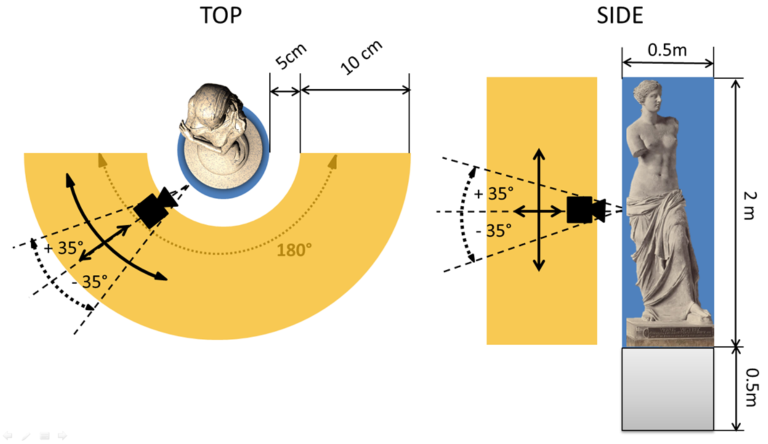


Figure 71: A visual representation of the requirements.

9.3 Creation of new ideas

In this section, first some ideas and guidelines from literature are presented, and then a categorization is presented based on platform shape. Different cable lay-outs for three different platform shapes are presented. The platforms are connected to a cuboid shaped frame which attachment points are number as shown in Figure 72.

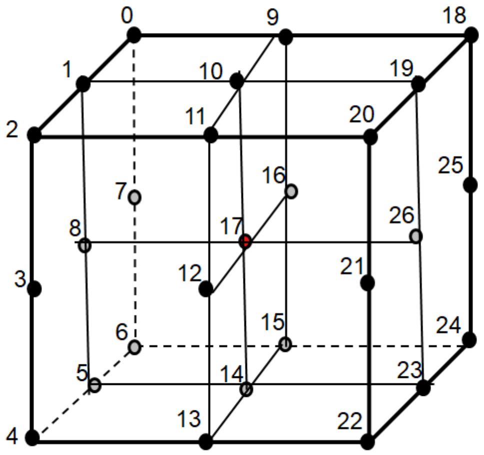


Figure 72: The numbered frame attachment points, in the end the pedestal is 1m.

9.3.1 Ideas from literature

The ideas taken from literature show a promising workspace and are based on the IPAnema-Falcon from[23], shown in figure 73, a suspended and not-suspended version of the CoGiRo from[9], with a suspended version shown in figure 75 and the locomotion robot by Perreault[22], shown in figure 74. In the work of Perreault, the design is altered by optimization and the optimized geometry is closely resembling a Falcon-IPAnema lay-out, but with crossed cables.

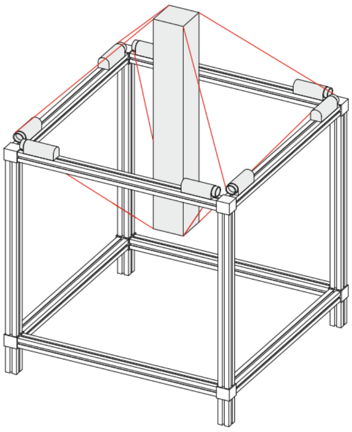


Figure 73: The Falcon-IPAnema robot[23]

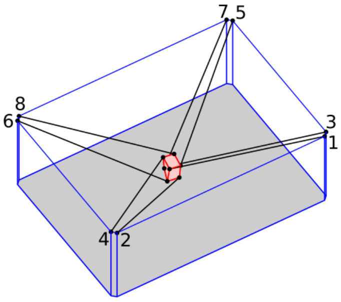


Figure 74: The design from Perreault[22]

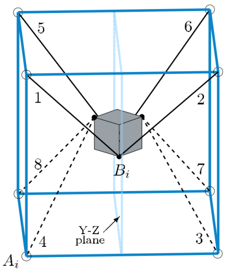


Figure 75: The CoGiRo CDPR[9]

Furthermore, some literature on designs was consulted. Many guidelines and promising designs are found in[30]. The following advice comes from this work:

1. ”If no external wrenches are involved, then extra redundant tendons are useful to improve the shape of the workspace, but not its quality.”[30]

2. ”If large external wrenches need to be supported, it may be a good idea to add a higher number of redundant tendons.”[30]

3. ”Connection points should be put together as often as possible. This usually increases substantially the rotational 24 workspace.”[30] (1R2T). Both for distal and proximal endpoints. Joining connection points decreases the amount of cable-cable collisions as well.

4. ”Manipulators with m = 2n tendons can be designed to have good decoupling properties at home posture. They offer a good workspace also for large rotations. This can be considered a reasonable limit on the maximum number of tendons to employ.”[30]

5. ”In few simple words, large workspaces of high quality can be obtained with large numbers of tendons, but one has to pay a price for this in terms of actuator cost, force requirements and computation time.”[30]

6. ”The 3R3T class is different from the other ones in that autocollisions play a central role; thus, the question of optimal manipulators in this class is still open.” A solution to this problem offered by[30] is to put the endpoints of the cables together.

Verhoeven also introduced optimal designs for the classes 2T, 1R2T, 2R3T and 3R3T. A design is called optimal by Verhoeven when the workspace that belongs to it has the most optimal combination of being large and far away from singularities. The designs for class 1R2T are shown in Figure 76. The goal of the design is to both enable translation and a large rotation, which could be seen as a top view of a cable robot rotating around a statue, without considering panning and tilting. So both design b and d in 76 can be used as inspiration for a 3D design and configuration such as a and c should be avoided.

In Figure 77, the 2R3T configuration that is best performing is design b. Some lessons could be drawn from this experiment: using more cables does not necessarily improve the workspace and designs that have cables forming a triangle with a common point at the frame have the best rotational workspace.

For this work, the most relevant designs are the 3R3T designs. The 3R3T designs tested and presented by Verhoeven are shown in figure 78. Design a and e are both design with the purpose of decoupling translation and rotation, but do not perform well. The design b, originating from[13], is able to show a large range of rotations but has problems with translation. Design c and d from[?] are performing well, design d is is better at rotation and design c better at translation. Design f is capable of rotating like design d, but has better translation. The downside of design e and f however, is that they show many collisions for large rotations, while design a, b, c and d do not. Especially design d could be promising, due to the large rotations and lack of collisions. It again consist of only cables forming triangles while intersecting at the frame.

Other considerations to be made during CDPR design are shared in the work of Lafourcade[14]. The following steps are taken by Lafourcade:

1. A platform shape is chosen. In the work of Lafourcade, a T-Platform is chosen since three points are the minimum of points to control an object in 6 DOF. A symmetrical platform is chosen.

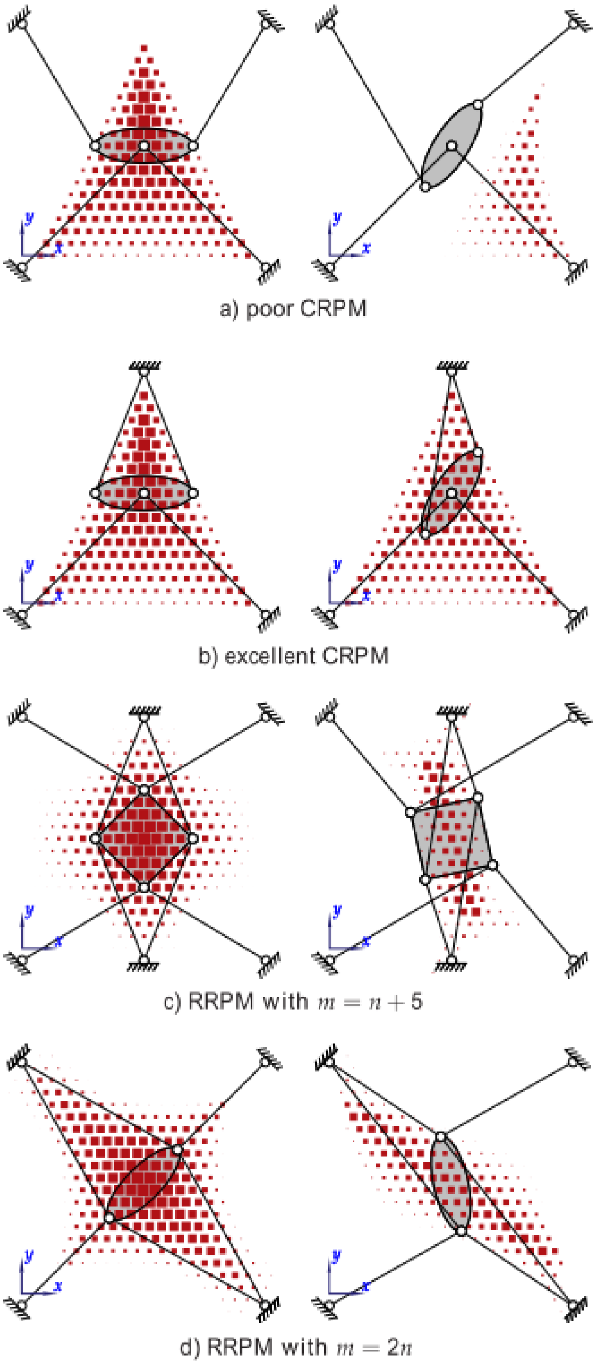
2. The cables are divided over the three attachment points. The cables need to be symmetrically divided and when only one cable is attached to a point it can only resist a torque in one direction which creates a singularity.

3. Then the frame attachment points of the cables are chosen. In Figure 79 a compilation of the images from[14] that show the design iterations, is shown. In a the cables go to the corners, which creates a large theoretical workspace. It is the space between the four corners. This image, however, shows that a singularity is reached for moderate rotations. This problem is tackled in b, but now a singularity in y-direction appears. This singularity is solved with the design in c. One of the triangles is reversed: instead of sharing an endpoint at the frame, they share on at the platform. The remaining three cables are easier to attach

4. The CDPR is viewed from all sides to evaluate at which points singularities will appear.

5. More cables are added to increase the size of the singularity free workspace. A distal endpoint is theoretically able to move in the shape between the frame points. However, the actual size of the workspace is always smaller because the cable tension reaches infinity at the sides. For this reason, it is preferable to have a large enough buffer between the theoretical workspace and the required workspace of a distal endpoint. The theoretical workspace can be made larger by adding frame attachment points at tactical places. To elaborate, the theoretical workspace in which the distal endpoints can move is larger in Figure 80 b than in Figure 80 a.

The steps by Lafourcade are useful for designing a statue-scanning CDPR too. First, a three point symmetrical platform can be chosen too, to minimize the amount of cable attachment



a) poor CRPM

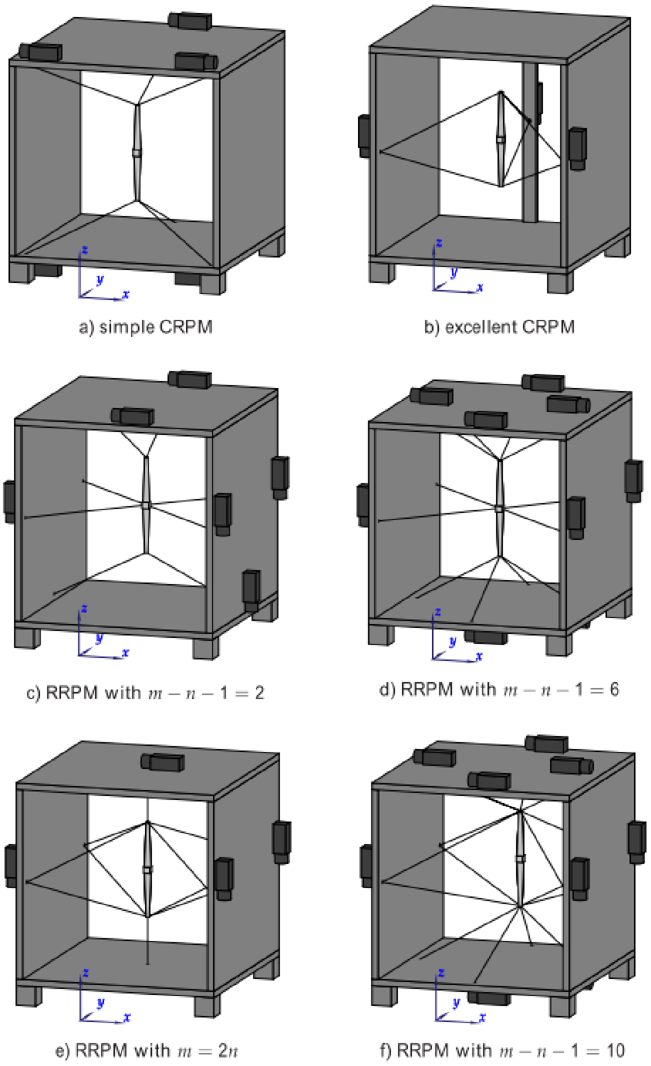
b) excellent CRPM

c) RRPM with $m = n + 5$

d) RRPM with $m = 2n$

Figure 76: The tested 1R2T cable configurations presented in[30]

points. Furthermore, the design for the roll motion by Lafourcade can be used for designing the yaw motion of the statue scanning CDPR, because they both have to be large. The results at



a) simple CRPM

b) excellent CRPM

c) RRPM with $m - n - 1 = 2$

d) RRPM with $m - n - 1 = 6$

e) RRPM with $m = 2n$

f) RRPM with $m - n - 1 = 10$

Figure 77: The tested 2R3T cable configurations presented in[30]

this step show once again that cable triangles with common attachment point at the frame are important for rotation and when another configuration is needed, the triangle should have its common attachment point at the platform. Also reviewing the design from the sides is a step that should be added to the design process. Finally, enlarging the theoretical workspace with extra cables is a step that should be taken too.

Finally the paper from Tadokoro[27] offers some optimal designs for 8 cables and large rotations while taking cable interference into account. These designs were later reviewed by Verhoeven and are shown in Figure 78c and 78d. Both geometries can have a rotation of ± 30 degrees around the y-axis, with the Kawamura design of Figure 78c having a larger workspace. This design however, is not able to rotate with ± 60 degrees around the z-axis, while the Nashioka design of Figure 78d can. Additionally, Tadokoro introduces heuristics to find an optimal design for a workspace with large rotations, they are as follows:

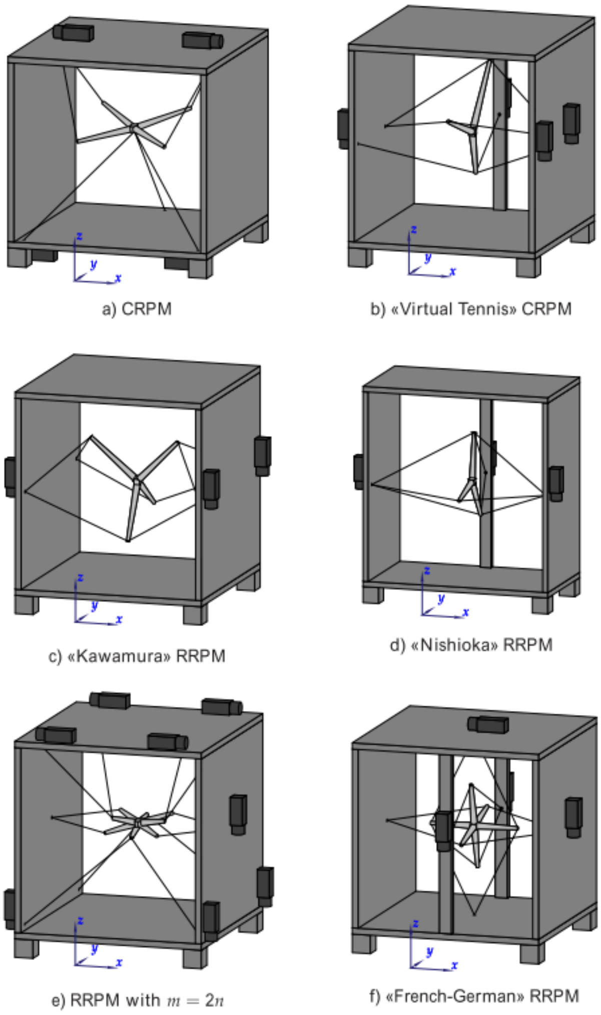


Figure 78: The tested 3R3T cable configurations presented in[30]

1. Every wire should have another wire that can reverse its moment and force.
2. The moment arm of cables on the center of gravity should be as large as possible.
3. For a large workspace, it is desirable to have long cables.
4. Cable collisions should be considered and the space between a cable and the platform should be kept large.
5. The geometry should be as symmetric as possible, especially around the axes of rotation.

9.3.2 T-Platform

The idea of using a T-shaped platform was taken from Lafourcade[14]. It is useful because it has as little different attachment points as possible and rotations are decoupled. Also Lafourcade's

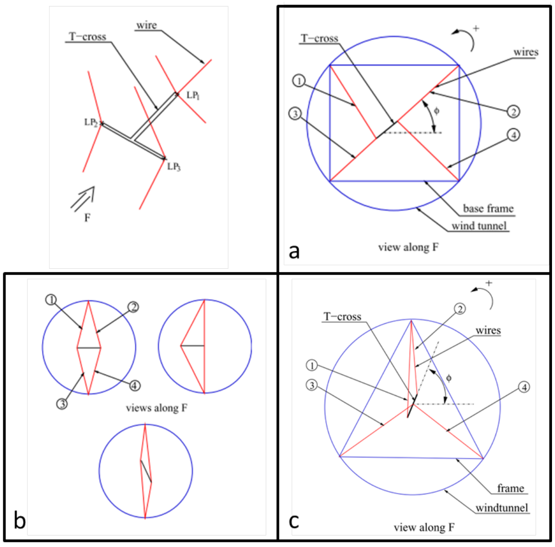


Figure 79: Different design iterations by Lafourcade[14] to find ideal frame attachment points

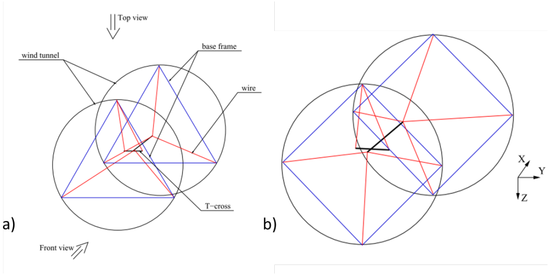


Figure 80: The change in theoretical workspace between two designs, taken frome[14]

insights in how to design the cable lay-out for large rotations and avoiding singularity are used. A difference in the design is the fact that for this work, mainly a large yaw movement is necessary and roll is not required at the base. Because of the statue, the T should be horizontal, when vertical, the stem of the T could interfere with the statue during tilting. Furthermore, cables from behind the statue can only be attached to the sides of the head of the T, otherwise they would collide with the statue. It is however useful to make use of the space behind the statue to find a cable-lay-out which enables the largest workspace. A schematic overview of the T-Platform orientation and the possible attachment points is shown in Figure 81. The T-shape has extra protruding rods at both ends so the cables do not interfere with the statue for large yaw motions. Disadvantages of the T-Platform could be that the decoupling is largest at the base position and orientation. The decoupling means that the cables attached to the stem of the T are responsible for tilting and the cables attached at the head of the T are responsible for yaw and panning.

Different cable lay-outs were created based on the designs of Lafourcade and Verhoeven. During the brainstorm and categorization, advice from Lafourcade, Verhoeven and Tadokoro was considered. When putting together connection points as much as possible, only three connection points at the platform are necessary for 3R3T. Also only three connection points at the frame are needed. Adding more proximal endpoints can increase the theoretical workspace of the distal

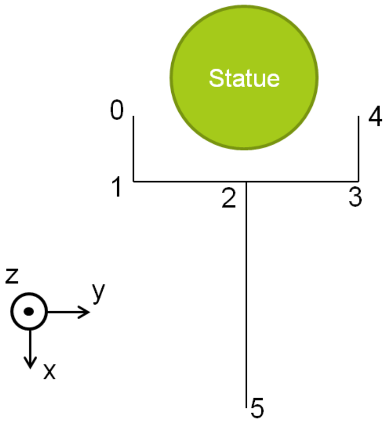


Figure 81: Schematic overview of the T-Platform and its orientation with respect to the statue. Possible cable attachment points are numbered.

endpoints, but there are also chances of singularities and collisions in this case. Extra cables can be used to carry high wrenches, but can increase the overall tension and make the system prone to cable-cable collision. According to[14] and[27], it is not allowed for T-platforms to have only a single cable at a distal endpoint, this means a singularity. Furthermore, symmetry is necessary for a reliable system[27]. According to Verhoeven, no more than 12 cables are necessary. This means the following cable distributions are possible between point 0-5 of Figure 81, with the first entry being point 5.

- 7 cables: 3-2-2
- 8 cables: 4-2-2, 2-3-3
- 9 cables: 3-3-3, 5-2-2
- 10 cables: 2-4-4, 4-3-3, 6-2-2
- 11 cables: 3-4-4, 5-3-3, 7-2-2
- 12 cables: 2-5-5, 4-4-4, 6-3-3, 8-2-2

For a few designs, cables were also connected at the T crossing. To create designs, first designs presented in Verhoevens and Lafourcades work were transformed to the T-Platform orientation. To get a better understanding of how the cables influence the workspace, the focus will at first lie on the configurations with 8, 9 and 10 cables. According to literature 8 cables are already sufficient, but it will be tested whether extra cables are able to enlarge the workspace by decreasing the space with singular positions. When the wrenches are not to high for the cables, these amounts can suffice to control the platform. From seven cables on, hypotheses will be tested to narrow down the choice of configurations with more cables. The hypotheses to test are as follows:

For the bird's beak configuration (the four cables going past the statue look like an open beak):

1. What is the influence of the amount of cables on the front side. So is it better to have three, four or five cables? A trade-off has to be made between amount of cables and costs of winches.

2. **3 cables:** In the case of three cables, is it better to have an Y or mercedes configuration?
3. **4 cables:** Should it form an X or a + and in the case of a plus?
4. **4 cables:** What is the general influence of making the frame shorter?
5. **4 cables:** Should the horizontal cables of the + be pointed inward more?

Testing other designs: Lafourcade:

1. How good is the 8 cable reversed bird's beak? (8 cables). Is it bad to have only two cables going to the back, or having them merely horizontally?
2. Should I add side cables? So is there a large added value when two side cables at the back are added to the 8 cable 4-2-2 designs? This means ending up with the 10 cable 8-(4-3-3) design for example. What would be the added value?
3. Does adding another bird's beak at the front help? Such as at the 10 cable 7-(4-3-3) design.
4. How good is a double bird's beak? Such as at the 10 cable 8-(4-4-2) Designs?
5. is it a good idea to add cables at the T-crossing?

Nishioka:

1. Does adding extra cables have a positive effect on the T-Platform?

Falcon:

1. How good is a Falcon
2. How good is a T-Falcon?
3. How good is a Falcon/ T-Falcon with cables going to the back (behind the statue)

To summarize I can choose how to use bird beaks, side cables, top/bot cables and front cables for all kinds of configurations. I want a suitable configuration with as less cables as possible.

Before testing, a suitable test case should be made. This includes dimensions of frame and platform that show a large workspace and panning and tilting requirements that allow a workspace to be formed. Still panning and tilting is included from the beginning since it is important for a design that this is enabled. From the first results it became clear that mainly the WFW is constraining the workspace when the platform is chosen in a way to avoid interferences. The WFW will mainly be looked at.

9.3.3 T-Platform Iterations

Experiment1 (T_8_6_442_V1):
Panning and tilting of 5 degrees. Mid 20 degrees from neutral position. It seems that the cable-platform collision does not play a role. It takes a substantial amount of time to calculate, so it will be turned off during this experiment.

- **Platform width** Test 1: lf=8, hf=4, wf=8, lp=0.8, wp=4, hp=0.2. Coverage is 52.25 %.
- Test 2: lf=8, hf=4, wf=8, lp=0.8, wp=3, hp=0.2. Coverage is 35.14 %.

- Test 3: lf=8, hf=4, wf=8, lp=0.8, wp=5, hp=0.2. Coverage is 58.15 %. It seems like cable-statue collision is limiting the workspace. How can this be increased?

- Test 4: lf=8, hf=4, wf=8, lp=0.8, wp=5.5, hp=0.2. Coverage is 56.46 %. Is the trimmed WFW even larger than the one of test 3? Yes but only very slightly, so wp=5 is a good distance. The trimmed WFW is significantly larger than the one of test 1. The following questions arise: is shortening the length of the frame having the same effect?

- **Frame length**Test 5: lf=7.5, hf=4, wf=8, lp=0.8, wp=4, hp=0.2. Coverage is 54.65%. So the workspace does increase. This is due to a WFW workspace increase. What happens when the frame is even smaller:

- Test 6: lf=7, hf=4, wf=8, lp=0.8, wp=4, hp=0.2. Coverage is 55.63%. The WFW is still getting larger. Al large part of the required workspace is now made smaller due to cable-platform collisions, but the frame will be made even smaller:

- Test 7: lf=6, hf=4, wf=8, lp=0.8, wp=4, hp=0.2. Coverage is 53.79%. There is less coverage, but the WFW Hull is even larger. Unfortunately, now cable-statue collision plays a large role, this will be handled next. But first let's try to make the WFW even larger(especially in the radial direction, it is also limited above and below, and this isn't changing):

- Test 8: lf=5, hf=4, wf=8, lp=0.8, wp=4, hp=0.2. Coverage is 43.14%. Now the WFW got a little smaller, In radial direction it increased even more, but the positive x direction, the WFW hull decreased. It was kinda flattened. Maybe this can be countered by shortening the stem?

- **Shortening the stem** Test 9: lf=5, hf=4, wf=8, lp=0.4, wp=4, hp=0.2. Coverage is 42.58% So shortening the stem didn't help. The WFW got a little better with 10%, especially radially, but now cable-cable are playing a role. Now test 7 is done with a shorter stem:

- Test 10: lf=6, hf=4, wf=8, lp=0.4, wp=4, hp=0.2. Coverage is 48.80% It appears that shortening the stem is making the WFW smaller in positive x-direction.

- Test 11: lf=6, hf=4, wf=8, lp=1.2, wp=4, hp=0.2. Coverage is 48.80% Making the stem larger has as as outcome that the WFW is smaller everywhere, so the lp = 0.8 seems reasonable. For now the stem will not be changed. Now for test 7, the teeth are made longer to try to avoid cable-statue collision

- **Protruding rods platform** Test 12: lf=6, hf=4, wf=8, lp=0.8, wp=4, hp=0.3. Coverage is 60.28 %. There is still cable-statue collision so let's add more teeth length.

- Test 13: lf=6, hf=4, wf=8, lp=0.8, wp=4, hp=0.4. Coverage is 64.59 %. Largest workspace so far, the impairing factors are the WFW in positive z-direction and WFW and cable-statue collision in radial direction. The teeth can still be made longer and the back frame points can be put closer to the statue.

- **Changing frame coordinates in x-direction** Test 14: [back points x:+1] lf=6, hf=4, wf=8, lp=0.8, wp=4, hp=0.4. Coverage is 68.18 %. Even larger workspace, now the cable-statue collision has become more impairing. Comparing workspace, it can be seen that the WFW is significantly larger in both positive z-direction, as radially. Cable-statue became a bit smaller radially. From now on cable-statue is included again. The teeth are made larger again for less cable-statue collision

- **Protruding rods platform** Test 15: [back points x:+1] lf=6, hf=4, wf=8, lp=0.8, wp=4, hp=0.5. Coverage is 75.92%. Making the protruding rods even larger:
- Test 16: [back points x:+1] lf=6, hf=4, wf=8, lp=0.8, wp=4, hp=0.7. Coverage is 78.34%.
- **Changing frame coordinates in y-direction** Test 16: [back points x:+1] lf=6, hf=4, wf=5, lp=0.8, wp=4, hp=0.7. Coverage is 63.93%. This makes the WFW smaller in the radial direction

Summary: Mainly the WFW and the cable-statue collision play a large role for this design. The influence of different parameters found empirically is listed below:

- **Platform width**: There is an ideal width, when the width is even larger, the workspace does not change quickly anymore. This could be caused by the angle of the cables with the platform, the smaller the angles between back cables and platform the better? This was also tested by placing the proximal attachment points in the back closer to the statue.
- **Frame length**: Changing the frame length creates a different workspace shape. When the frame is long, the workspace is long and more slender. When it is shorter, the workspace is shorter and more wide. The frame length should be tweaked to find a WFW matching the required workspace.
- **Shortening the stem** Making the stem too short is causing cable interference. And making the WFW smaller x direction. Making the stem too large also decreases the workspace, so a sweet spot is to be found.
- **Protruding rods** Making the rods longer decreases the cable-statue collision, they can be made longer than the statue diameter. When optimizing, there maybe should be an upper limit on the platform size. The optimization does not take inconvenience into account.
- **Changing frame coordinates in x-direction**. Putting the proximal endpoints behind the statue more close to the statue is increasing the workspace.
- **Changing frame coordinates in y-direction**. Putting the proximal endpoints of the statue more close to the statue is increasing the workspace.

Experiment2 (T_8_6_442_V1):
Testing different panning and tilting for the most optimal design.

- Test 1: [back points x:+1] lf=6, hf=4, wf=8, lp=0.8, wp=4, hp=0.7. tilting ± 5 degrees, tilting ± 10 degrees. Coverage: 71.64 %
- Test 2: [back points x:+1] lf=6, hf=4, wf=8, lp=0.8, wp=4, hp=0.7. tilting ± 5 degrees, panning ± 5 degrees. Coverage: 78.34 %
- Test 3: [back points x:+1] lf=6, hf=4, wf=8, lp=0.8, wp=4, hp=0.7. tilting ± 5 degrees, panning ± 15 degrees. Coverage: 64.68 %.
- Test 4: [back points x:+1] lf=6, hf=4, wf=8, lp=0.8, wp=4, hp=0.7. tilting ± 5 degrees, panning ± 20 degrees. Coverage: 58.20 %.
- Test 5: [back points x:+1] lf=6, hf=4, wf=8, lp=0.8, wp=4, hp=0.7. tilting ± 5 degrees, panning ± 25 degrees. Coverage: 50.41 %.

- Test 6: [back points x:+1] lf=6, hf=4, wf=8, lp=0.8, wp=4, hp=0.7. tilting ± 5 degrees, panning ± 30 degrees. Coverage: 41.97 %. The platform-statue collision is limited in a weird way: found out why: I thought it was tilting, but actually it was panning that is large. Then it makes sense that the platform collides with the statue because the panning is around another point as the yaw.
- Test 7: [back points x:+1] lf=6, hf=4, wf=8, lp=0.8, wp=4, hp=0.7. tilting ± 5 degrees, panning ± 35 degrees. Coverage: 37.75 %.
- Test 8: [back points x:+1] lf=6, hf=4, wf=8, lp=0.8, wp=4, hp=0.7. tilting ± 0 degrees, tilting ± 0 degrees. Coverage: 83.89 %
- **Adding Tilting** Test 9: [back points x:+1] lf=6, hf=4, wf=8, lp=0.8, wp=4, hp=0.7. tilting ± 10 degrees, tilting ± 0 degrees. Coverage: 82.93 %
- Test 10: [back points x:+1] lf=6, hf=4, wf=8, lp=0.8, wp=4, hp=0.7. tilting ± 20 degrees, tilting ± 0 degrees. Coverage: 64.53 %
- Test 11: [back points x:+1] lf=6, hf=4, wf=8, lp=0.8, wp=4, hp=0.7. tilting ± 30 degrees, tilting ± 0 degrees. Coverage: 0 % Mid is not admissible for WFW
- Test 12: [back points x:+1] lf=6, hf=4, wf=8, lp=0.4, wp=4, hp=0.7. tilting ± 30 degrees, tilting ± 0 degrees. Coverage: 13.18 % It is a very weird workspace. Cable-Statue collision now plays a role too. Maybe it helps to increase the teeth length.
- Test 13: [back points x:+1] lf=6, hf=4, wf=8, lp=0.4, wp=4, hp=1. tilting ± 30 degrees, tilting ± 0 degrees. Coverage: 23.15 % The workspace is now larger again. Does it help to make the teeth even longer?
- Test 13b: [back points x:+1] lf=6, hf=4, wf=8, lp=0.4, wp=4, hp=1.2. tilting ± 30 degrees, tilting ± 0 degrees. Coverage: 0 % WFW is not admissible anymore. What happens when changing the length of the stem to zero?
- Test 13c: [back points x:+1] lf=6, hf=4, wf=8, lp=0, wp=4, hp=1. tilting ± 30 degrees, tilting ± 0 degrees. Coverage: 0 % WFW is not admissible anymore.
- Test 14: [back points x:+1] lf=6, hf=4, wf=8, lp=0.2, wp=4, hp=1. tilting ± 30 degrees, tilting ± 0 degrees. Coverage: 22.89 % Is the width of the platform still important?
- Test 15: [back points x:+1] lf=6, hf=4, wf=8, lp=0.2, wp=3.5, hp=1. tilting ± 30 degrees, tilting ± 0 degrees. Coverage: 22.44 % There is not a lot of change, can the width be decreased more?
- Test 16: [back points x:+1] lf=6, hf=4, wf=8, lp=0.2, wp=3, hp=1. tilting ± 30 degrees, tilting ± 0 degrees. Coverage: 17.15 % So now the width is becoming important
- Test 17: [back points x:+1] lf=6, hf=4, wf=7, lp=0.2, wp=4, hp=1. tilting ± 30 degrees, tilting ± 0 degrees. Coverage: 26.25 % So a frame that is less wide is good, it increases the cable-platform collision workspace. But how to improve the WFW?
- Test 18: [back points x:+1] lf=5.5, hf=4, wf=6, lp=0.2, wp=4, hp=0.6. tilting ± 30 degrees, tilting ± 0 degrees. Coverage: 26.68 % So not a real increase, how to improve the WFW further?

- Test 19: [back points x:+1][vertex 4, 2: [-1, 0, 0] lf=5.5, hf=4, wf=6, lp=0.2, wp=4, hp=0.6. tilting ± 30 degrees, tilting ± 0 degrees. Coverage: 19.29 % The WFW is only getting worse

- Test 20: [back points x:+1][vertex 22, 20: [-1, 0, 0] lf=5.5, hf=4, wf=6, lp=0.2, wp=4, hp=0.6. tilting ± 30 degrees, tilting ± 0 degrees. Coverage: 28.94 % The WFW is only getting worse. But the results of the cable-cable and cable-platform interaction are better.

- Test 21: [back points x:+1] lf=5.5, hf=5, wf=6, lp=0.2, wp=4, hp=0.6. tilting ± 30 degrees, tilting ± 0 degrees. Coverage: 64.94 % The WFW is very good, as well as the overall results. The proximal endpoints need to be more above the platform. Let's see whether having a normal height but reduced frame dimensions also has a positive effect:

- Test 22: [back points x:+1] lf=5, hf=4, wf=5, lp=0.2, wp=4, hp=0.6. tilting ± 30 degrees, tilting ± 0 degrees. Coverage: 41.89 % It does have a positive influence on the workspace. I leave it here.

- Test 23: [back points x:+1] lf=5, hf=4, wf=5, lp=0.2, wp=4, hp=0.6. tilting ± 0 degrees, tilting ± 0 degrees. Coverage: 83.69 % The workspace is almost the same as the one in test 8, but now the tilting is way better

- Test 24: [back points x:+1] lf=5, hf=4, wf=5, lp=0.2, wp=4, hp=0.6. tilting ± 35 degrees, tilting ± 35 degrees. Coverage: 5 % The workspace is almost the same as the one in test 8, but now the tilting is way better.

Thoughts: Many parameters can be tweaked, the current geometry is good for this design, but it is not clear whether it is good for the other designs. So different categories of lay-outs cannot really be compared with this design. I should test the Nishioka and T-Falcon independently. I should not do this as elaborately as for this lay-out. This lay-out will be called the Lafourcade-T and variations will be tested first in the next tests:

Testing the Lafourcade-T: mid on 20 degrees, [back points x:+1] lf=5, hf=4, wf=5, lp=0.2, wp=2, hp=0.6. The width of the platform is made smaller, 4m is mechanically not feasible. Four different tests will be done: only ± 20 degrees tilting, only ± 35 degrees panning, no tilting and panning, and both. These tests will be called: tilt, pan, none, both respectively. 25 Designs were evaluated and put into three categories: Lafourcade based, Nishioka Based and Falcon based. They are difficult to compare so the best performing designs was chosen in each category.

Lafourcade insights:

- The X cables at the front of T_8_6_442_V1 are better than the plus cables of T_8_6_442_V2. The side cables at the front are interfering with the platform.

- The Mercedes Logo configuration of the front cables of T_7_5_332_mercie are performing better than the Y cable configuration of T_7_5_332_Y

- T_9_7_522_V2 Shows that adding another cable to the end of the T at the front does not improve the performance of T_8_6_442_V1.

- T_10_8_433_back: the extra cables to the back have no added value.

- T_10_8_433_front, mid the extra cables only make the performance of T_8_6_442_V1 worse.

- T_10_8_4222 The top and bottom cables do have added value.

- T_10_6_442 performs better than a more Nishioka like geometry as in T_10_6_442_Nishioka

- The extra cables of T_12_10_633_back do not have added value.
- **Choices** I choose to optimize T_8_6_442_V1 first, because it is advantageous to have less cables, maybe later T_10_8_4222 can be optimized.

T-Falcon insights:

- The T-Falcon T_8_4_422_T_Falcon is capable of both panning and tilting and does not have cables enclosing the statue. It does not perform as well as the T_8_6_442_V1, but has the advantage of not enclosing.
- The T_10_5_433_T_Falcon and T_11_6_533_T_Falcon are both enclosing the statue with the cables, but do not show to be better than the Lafourcade designs. They are less interesting for optimization. T_10_5_433_T_Falcon has less cables and a better performance.
- **Choices** T_8_4_422_T_Falcon is chosen to be optimized since it does not enclose the statue.

Nishioka insights:

- The altered version of the nishioka design T_9_4_333_NishiokaMercie, with two cables diverging from the platform at the front, performs better than the original Nishioka Design T_9_4_333_Nishioka. The version with the cables diverging upwards was not tested, because the Mercedes configuration was better for an earlier design than the Y configuration. After testing the Y configuration is worse in this case too.
- The rotated Nishioka T_8_3_332_Nishioka_Rotated does not have a good performance, but could be optimized for good performance. It is interesting because it does not enclose the statue.
- **Choices**: Depends on how many cables I want. When keeping it low: I should pick the traditional Nishioka.

9.3.4 H-platform

The H-Platform is based on the design of the T-platform of Lafourcade. The thought behind the design was to use a T-platform with a vertical stem. This is not possible because the stem will interfere with the statue during tilting close to the statue. But when this stem is moved to the side it does not interfere anymore. To make the platform symmetrical, the same stem is added on the other side and the platform pictured in Figure 82 is created. The H-platform has more attachment points than the T-platform, so more combinations are possible.

The first main category is the amount of cables. The second category is the cable orientation towards the back, so at the side of the statue. A distinction is made between an open and closed vertical beak. A beak is necessary for large yaw rotation. Then extra distinction is made in what other types of cable configurations are added. The following list of possible configuration additions was made and is visualized in Figure 83:

- Back vertical open birdsbeak(spread) (4 cables)
- Back vertical closed birdsbeak(spread) (4 cables)
- Back vertical open birdsbeak (4 cables)
- Back vertical closed birdsbeak (2 cables)
- Back horizontal open birdsbeak (4 cables)

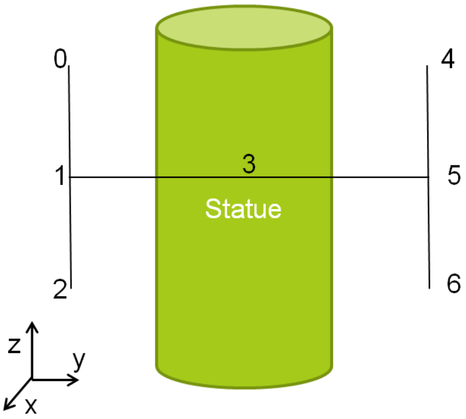


Figure 82: Schematic overview of the H-Platform and its orientation with respect to the statue. Possible cable attachment points are numbered.

- Front vertical open birdsbeak(spread) (4 cables)
- Front vertical closed birdsbeak(spread) (4 cables)
- Front vertical open birdsbeak (4 cables)
- Front vertical closed birdsbeak (2 cables)
- Front horizontal open birdsbeak (4 cables)
- Side cables, spread at platform
- Side cables, (at the vertical middle)
- Front cables, 2, 3, 4 or 5 of them in different configurations.
- Cables going to the top or bottom side of the frame (Not visualized in Figure 83.

First two seven cable design were created. The cables going to frame points behind the statue are now responsible for tilting and take over this function from the cables connected to the stem for the T-Platform design. The panning and tilting motions are now not decoupled anymore. For eight cable designs, it is either possible to add another cable at the front going to the middle of the H-platform, like the 6-(4-1-1-1-1) design or add another open or closed bird's beak. For 10 cables, side cables could be added, or more beaks, but before expanding the design space, experiments first have to show how effective it is to add extra cables.

9.3.5 H-Platform Experiments

The following tests were done (mid 20 degrees, tilt ± 20 degrees:

1. 3: H_8_6_41111 lf=5, hf=4, wf=5, lp=0, wp=2, hp=0.5 frameshift=[9: +1x, 15: +1x] Coverage: 12,06 %.

2. 3: H_8_6_41111 lf=6, hf=4, wf=5, lp=0, wp=2, hp=0.5 frameshift=[9: +1x +0.5z, 15: +1x] Coverage: 15.60 % height is not increasing much. THere is now less cable-cable interference.
3. 4: H_8_6_41111 lf=7, hf=4, wf=5, lp=0, wp=2, hp=0.5 frameshift=[9: +1x, 15: +1x] Coverage: 18.21 %
4. 5: H_8_6_41111 lf=8, hf=4, wf=5, lp=0, wp=2, hp=0.5 frameshift=[9: +1x, 15: +1x] Coverage: 19.46 %
5. 6: H_8_6_41111 lf=8, hf=4, wf=6, lp=0, wp=2, hp=0.5 frameshift=[9: +1x, 15: +1x] Coverage: 18.38 %
6. 7: H_8_6_41111 lf=8, hf=4, wf=5, lp=0, wp=2, hp=0.5 Coverage: 23.57 % The WFW is better now.
7. 8: H_8_6_41111 lf=8, hf=4, wf=4, lp=0, wp=2, hp=0.5 Coverage: 24.18 %
8. 9: H_8_6_41111 lf=8, hf=4, wf=4, lp=0, wp=2, hp=0.4 Coverage: 22.84 %
9. 10: H_8_6_41111 lf=8, hf=4, wf=4, lp=0, wp=2, hp=0.6 Coverage: 25.09 %
10. 11: H_8_6_41111 lf=8, hf=4, wf=4, lp=0, wp=2, hp=0.7 Coverage: 25.57 %
11. 12: H_8_6_41111 lf=8, hf=4, wf=4, lp=0.2, wp=2, hp=0.7 Coverage: 24.59 % When using the depth of the H-platform, cable-statue collision workspace increases, but the wfw decreases

Other designs:

1. H_8_4_221111_horfrontopen needed new parameters for better performance. These are: lf=8, hf=4, wf=6, lp=0, wp=1, hp=1. These will be used with all horizontal configurations. They are a separate category.

It also appears for the H-Platform that the size of the vertical platform rods is only influencing the workspace size for tilting, not the workspace size without tilting.

9.3.6 C-Platform

This category is created, because this design offers many connecting points. Many more different cable lay-outs can be thought of now and the platform can have any shape by only connecting the distal endpoints. The C-Platform is pictured in Figure 87. Part of the platform has been removed on the statue side, since no cables can be attached here. These cables otherwise would interfere with the statue. Many created designs can be viewed as variations on T- and H-platform designs, but with another dimension added. The designs are first categorized by the amount of cables. The other categories are:

- Designs with a closed birds beak at the front side. These designs are only found from 10 cables onward because they can not be symmetric in the xy plane without singularities.
- Designs with an open birds beak at the statue side with subcategories:
  - Separate cables at the front
  - Vertical beaks at the front

    - Horizontal beaks at the front

  - Designs based on the CaRISA design of Tempel[28].

  - Designs based on the WARP/ Nishioka design of Tadokoro[26].

9.3.7 Testing C-Platforms

TO test the C-PLatforms, another parameter was added: lp_b which is the length of the base part of the platform. lp is representing the protruding rods around the statue. First the C-Platform is compared with the T-Platform, by creating the same geometries, but then with height. These are described with pink squares on the white board. For the tests, tilting of ± 20 degrees is used and a mid position of 20 degrees.

  - C_8_6_41111: Not possible due to cable-platform interference

  - C_8_6_8x1: lf=5, hf=4, wf=5, lp=0.6, lp_b=0.2, wp=2, hp=0.2 frameshift=[9: +1x, 15: +1x] Coverage: %.

  - C_8_2_221111_H lf=5, hf=4, wf=5, lp=0, wp=1, hp=0.5 frameshift=[9: +1x, 15: +1x] Coverage: 46,97 %.

  - C_8_4_221111_H_2D lf=5, hf=4, wf=5, lp=0, wp=1, hp=0.5 frameshift=[9: +1x, 15: +1x] Coverage: 42,39 %. It turns out the 2D variation is way better then the other two.

  - C_8_IPAFalcon_X_VertCross lf=5, hf=4, wf=5, lp=0, wp=1, hp=0.5 frameshift=[9: +1x, 15: +1x] Coverage: 0 %. Obvious cable-cable interference. No significant wfw gain with respect to C_8_4_221111_H_2D.

  - C_8_4_8x1_Carisa_X_axis_TFalcon lf=5, hf=4, wf=5, lp=0, wp=1, hp=0.5 frameshift=[1: +1.25x, 19: +1.25x, 23 +1.25x, 5 +1.25x] Coverage: 0 %. The WFW is not better than the TFalcon one with the same sprint(B$_i$)$ize.C\_8\_TFalcon\_VertCrosslf = 5, hf = 4, wf = 5, lp = 0, wp = 1, hp = 0.5frameshift = [1 : +1.25x, 19 : +1.25x, 23 + 1.25x, 5 + 1.25x]Coverage : 0\%.TheWFWisonlyslightlybetterthantheTFalcononewiththesamesize.$

- C_8_TFalcon_VertCross_LONG (is like TFALCON 3D) lf=5, hf=4, wf=5, lp=1, lp_b=0.5, wp=0.75, hp=0.2 frameshift=[1: +0.25x, 19: +0.25x, 23 +0.25x, 5 +0.25x] Coverage: 0 %.

- C_8_4_8x1_Carisa_X_axis_TFalcon_LONGSQUARE lf=5, hf=4, wf=5, lp=1, lp_b=0.5, wp=0.75, hp=0.2 frameshift=[1: +0.25x, 19: +0.25x, 23 +0.25x, 5 +0.25x] Coverage: 0 %. Has a worse WFW than the previous example.

- C_8_4_8x1_FALCON_IPA lf=5, hf=4, wf=5, lp=1, lp_b=0.5, wp=0.75, hp=0.2 frameshift=[1: +0.25x, 19: +0.25x, 23 +0.25x, 5 +0.25x] Coverage: 52.59 %.

- C_8_IPAFalcon_Y_2D lf=5, hf=4, wf=5, lp=1, lp_b=0.5, wp=0.75, hp=0 frameshift=[1: +0.25x, 19: +0.25x, 23 +0.25x, 5 +0.25x] Coverage: 55.10 %. THe Ipanema Design shows a better result 2D than 3D.

- C_8_TFalcon_Y lf=5, hf=4, wf=5, lp=1, lp_b=0.5, wp=0.75, hp=0.2 frameshift=[1: +0.25x, 19: +0.25x, 23 +0.25x, 5 +0.25x] Coverage: 74.10 %. THe T Design shows a better result than the square design.

9.3.8 Crossed configurations

The T-Falcon configuration is given a height of 0.8. Next it is given either a vertical crossing of cables, or even a double crossing. The changes in the WFW are compared. It appears that crossing the cables is not making the workspace better. Placing some cables more towards the statue and letting them cross in this way is only decreasing the workspace. Placing some to the front is changing the shape of the workspace as seen in Figure **??**. It does appear that having vertical crossing improves the workspace as seen in Figure 84.

Crossed configurations for C_8_T_8_7_3321_Y_top:
It appears that only the vertical crossing at both sides has a considerable advantage, but is also causing cable-cable and cable-platform interference.

Crossed configurations for C_8_H_8_4_2222:
Horizontal crossing does not change a lot, but could be useful when the the frame cannot fully enclose the statue. Vertical crossing is not making the solution better.

Crossed configurations for IPAnema-Falcon:
Here the crossed configuration does have added value, as seen in Figure 86 for the CARISA lay-out and dimensions.

9.4 Adding the wrench

The forces of the cables and the wrench of the platform are balanced around point $[0,0,0]$ in platform coordinates. The platform wrench consists of the weight of the platform and the moment of the center of gravity around point $[0,0,0]$. This is caused by the fact that the center of gravity is not on point $[0,0,0]$. The moment of the center of gravity is the product of the weight and the distance to point $[0,0,0]$. This is calculated based on the platform dimensions and the platform type. In the next paragraphs a description of the calculation is given for every platform type. What first should be mentioned is that the total platform weight is considered 5 kg. The camera is assumed to be the SPECIM camera, which has a depth of 91 mm and weighs 1.3 kg. The CoG of the camera is 45.5 mm away from [0, 0, 0], since this point is the focal point. The contribution of the platform to the shift of the CoG is calculated next. The weight remaining for the frame of the platform is 3.7 kg. Due to symmetry in z and y direction, only a center of gravity distance along the y-axis is modelled.

9.4.1 T-Platform

The T-Platform is considered here as a homogeneous triangle with two protruding beams for cable attachments as viewed in Figure **??**. The weight of the triangle is considered to be 3.7 kg times the ratio of the length of $w_p + l_p$ to total length $2 * h_p + l_p + w_p$. The distance of this weight to $[0,0,0]$ is one third of l_p. The weight of the two protruding sticks is approximated as 3.7 kg multiplied with the ratio of $2 * h_p$ to total length $2 * h_p + l_p + w_p$. The distance of this weight to $[0,0,0]$ is $-h_p/2$. The CoG location is recovered by adding the mass times moment arm of the camera, protruding sticks and triangle and dividing this by the total weight.

9.4.2 H-Platform

. Only the camera contributes.

9.4.3 C-Platform

The C-Platform is modelled as a body of ribs as drawn in Figure **??**. The total length of the ribs is $4*h_p+4*l_p+4*l_{pb}+4*w_p$. Two ribs in y direction contribute to the shift of the CoG from point $[0,0,0]$. Their weight is 3.7 kg times the ratio $2*w_p$ to total length $4*h_p+4*l_p+4*l_{pb}+4*w_p$ and their moment arm is l_{pb}.

Four ribs in x direction contribute to the shift of the CoG from point $[0,0,0]$. Their weight is 3.7 kg times the ratio $4*l_p$ to total length $4*h_p+4*l_p+4*l_{pb}+4*w_p$ and their moment arm is $-0.5*l_p$.

Four other ribs (l_{pb} in x direction contribute to the shift of the CoG from point $[0,0,0]$. Their weight is 3.7 kg times the ratio $4*l_{pb}$ to total length $4*h_p+4*l_p+4*l_{pb}+4*w_p$ and their moment arm is $0.5*l_{pb}$.

Four ribs in z direction contribute to the shift of the CoG from point $[0,0,0]$. Their weight is 3.7 kg times the ratio $4*h_p$ to total length $4*h_p+4*l_p+4*l_{pb}+4*w_p$ and their moment arm for two ribs is $-0.5*l_p$ and for two ribs is $-0.5*l_{pb}$.

The CoG location is recovered by adding the mass times moment arm of the camera and ribs and dividing this by the total weight.

9.4.4 The rotated wrench

At the starting point, the wrench will be a negative force in z-direction and a positive moment around the y-axis. This moment changes however as the platform rotates. Tilting in any direction decreases the moment arm of the gravity around point [0, 0, 0]. This is accounted for by multiplying the moment with the cosine of the tilting angle. The yaw around the statue combined with panning decreases the moment around the y-axis but increases the moment around the x-axis. So the moment after tilting is multiplied with the total yaw rotation matrix.

10 Appendix 1: Cable-Cable Collision Test Parameters

10.1 Test case 1

```
TC_1 = dict(
    title_case=’TC 1’,
    # Get the frame coordinates and the relative platform coordinates:
    lf=1,
    hf=1,
    wf=1,
    d_wf=0,  # CDPRdimensioning default value is: 0
    d_hf=0,  # CDPRdimensioning default value is: 0
    lp=0.6,
    wp=0.1,
    hp=0.2,
    top_off_x=-0.10,
    top_off_y=0,
    top_off_z=0,
    bot_off_z=0,

    # Position of the platform and cable forces are
    # (just for drawin of the platform):
    location_platform=np.array([0, 0, 0]),

    # Rotation of the
    phi=0,  # rotation around x axis, cdpr_kinematics default value is: 0
    theta=0,  # rotation around y axis, cdpr_kinematics default value is: 0
    psi=15,  # rotation around z axis, cdpr_kinematics default value is: 0

    ShuffleKey=np.array([3, 2, 1, 0, 7, 6, 5, 4])
    )
```

10.2 Test case 2

```
TC_2 = dict(
    title_case='TC 2',
    # Get the frame coordinates and the relative platform coordinates:
    lf=1,
    hf=1,
    wf=1,
    d_wf=0,  # CDPRdimensioning default value is: 0
    d_hf=0,  # CDPRdimensioning default value is: 0
    lp=0.20,
    wp=0.20,
    hp=0.1,
    top_off_x=0,
    top_off_y=0,
    top_off_z=0,
    bot_off_z=0,

    # Position of the platform and cable forces are
    # (just for drawin of the platform):
    xr_p=0,
    yr_p=0,
    zr_p=0,
    location_platform=np.array([0, 0, 0]),

    # Rotation of the
    phi=0,  # rotation around x axis, cdpr_kinematics default value is: 0
    theta=10,  # rotation around y axis, cdpr_kinematics default value is: 0
    psi=-45,  # rotation around z axis, cdpr_kinematics default value is: 0

    ShuffleKey=np.array([3, 6, 1, 1, 3, 6, 4, 4])
    # ShuffleKey = np.array([0, 1, 2, 3, 4, 5, 6, 7])
    )
```

10.3 Test case 3

```
TC_3 = dict(
    title_case='TC 3',
    # Get the frame coordinates and the relative platform coordinates:
    lf=1,
    hf=1,
    wf=1,
    # In case of Falcon like robot, the next two parameters can disperse the
    # location of the proximal endpoints
    d_wf=0,  # CDPRdimensioning default value is: 0
    d_hf=0,  # CDPRdimensioning default value is: 0
    lp=0.6,
    wp=0.3,
    hp=0.4,
    top_off_x=-0.03,
    top_off_y=0.03,
    top_off_z=0,
    bot_off_z=1,

    # Position of the platform and cable forces are
    # (just for drawin of the platform):
    location_platform=np.array([0, 0, 0]),

    # Rotation of the
    phi=0,  # rotation around x axis, cdpr_kinematics default value is: 0
    theta=0,  # rotation around y axis, cdpr_kinematics default value is: 0
    psi=15,  # rotation around z axis, cdpr_kinematics default value is: 0

    ShuffleKey=np.array([2, 0, 6, 4, 5, 7, 1, 3])
    # ShuffleKey = np.array([0, 1, 2, 3, 4, 5, 6, 7])
    )
```

10.4 Test case 4

```
TC_4 = dict(
    title_case='TC 4',
    # Get the frame coordinates and the relative platform coordinates:
    lf=1,
    hf=1,
    wf=1,
    # In case of Falcon like robot, the next two parameters can disperse the
    # location of the proximal endpoints
    d_wf=0,  # CDPRdimensioning default value is: 0
    d_hf=0,  # CDPRdimensioning default value is: 0
    lp=0.6,
    wp=0.3,
    hp=0.4,
    top_off_x=-0.03,
    top_off_y=0.03,
    top_off_z=0,
    bot_off_z=1,

    # Position of the platform and cable forces are
    # (just for drawin of the platform):
    location_platform=np.array([0, 0, 0]),

    # Rotation of the
    phi=5,  # rotation around x axis, cdpr_kinematics default value is: 0
    theta=10,  # rotation around y axis, cdpr_kinematics default value is: 0
    psi=-5,  # rotation around z axis, cdpr_kinematics default value is: 0

    ShuffleKey=np.array([2, 0, 6, 4, 5, 7, 1, 3])
    # ShuffleKey = np.array([0, 1, 2, 3, 4, 5, 6, 7])
    )
```

11 Appendix B: Test Set-Up

```
# -*- coding: utf-8 -*-
import numpy as np
import scipy
import matplotlib.pyplot as plt
import math
# plt.close('all')
```

```
from IPython import get_ipython
get_ipython().magic('reset -sf')
import os
os.chdir("C:/Users/Mees/Documents/Master/Thesis/Code")
from CDPRFunctions import *
# from OldFunctions import
# from Class_Icosahedron import IcoSphereCreator
from mpl_toolkits.mplot3d import axes3d
from mpl_toolkits.mplot3d.art3d import Poly3DCollection
from mpl_toolkits.mplot3d.axes3d import Axes3D
from matplotlib.tri import Triangulation
import itertools as it
from scipy.spatial.transform import Rotation
import datetime
import CDPRDesigns
import pickle

# Path of choice:
my_path = 'C:/Users/Mees/Documents/Master/Thesis/Methods/Pictures/Total Hull/
                                         Adding Rotations/20pan-20tilt/TC2/'
if not os.path.isdir(my_path):
    os.makedirs(my_path)
# Design
CDPR_Design = CDPRDesigns.TC_2

# Design Parameters: ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
title_case = CDPR_Design['title_case']
# Get the frame coordinates and the relative platform coordinates:
lf = CDPR_Design['lf']
hf = CDPR_Design['hf']
wf = CDPR_Design['wf']
d_wf = CDPR_Design['d_wf']  # CDPRdimensioning default value is: 0
d_hf = CDPR_Design['d_hf']  # CDPRdimensioning default value is: 0
lp = CDPR_Design['lp']
wp = CDPR_Design['wp']
hp = CDPR_Design['hp']
top_off_x = CDPR_Design['top_off_x']
top_off_y = CDPR_Design['top_off_y']
top_off_z = CDPR_Design['top_off_z']
bot_off_z = CDPR_Design['bot_off_z']
# Position of the platform and cable forces are(just for drawin of the platform):
location_platform = CDPR_Design['location_platform']
# Rotation of the
phi = CDPR_Design['phi']  # rotation around x axis, cdpr_kinematics default value
                                         is: 0
theta = CDPR_Design['theta']  # rotation around y axis, cdpr_kinematics default
                                         value is: 0
psi = CDPR_Design['psi'] # rotation around z axis, cdpr_kinematics default value
                                         is: 0
ShuffleKey = CDPR_Design['ShuffleKey']
# ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

n = 6  # CDPRdimensioning default value is: 6 (DoF)
m = 8  # CDPRdimensioning default value is: 8 (cables)
wrench = np.array([0, 0, -50, 0, 0, 0])  # CDPRkinematics default value is:
# 0,0,0,0,0,0
fmin = 5  # CDPRkinematics default value is: 5
fmax = 150  # CDPRkinematics default value is: 150

# parameter for cable-collision
cable_width = 0.005
```

```
# Parameters for statue collision
statue_height = 1  # default value was 1
statue_radius = 0.1
statue_base = np.array([-0.5, 0, -0.5*statue_height])  # default x val was -0.5

# rotational requirements:
x_rot_range = [0, 0, 0]
y_rot_range = [-20, 0, 20] # tilting
z_rot_range = [-20, 0, 20] # panning

# Hull method parameters:
starting_point = [0, 0, 0] # Neurtal & starting position of plaftorm
lambda_minInit = 0
lambda_maxInit = 0.85
eps = 0.0025
mid = np.array([-0.05, 0.05, 0.05]) # starting point of the hull method
recur_lvl = 1
stepsize_bench = 0.005

# Reference Workspace parameters:
inner_radius = 0.12
outer_radius = 0.25
z_rot_iterations = 1
theta_workspace_divis = np.degrees(np.linspace(0, 1, z_rot_iterations + 1) * 0.5 *
                                          np.pi)


#Setting up the rotation matrix:
title_pan_tilt = f'[\u00B1{y_rot_range[2]}]t \
& [\u00B1{z_rot_range[2]}]p'
rot_eval_matrix = np.array([[x_rot_range[1], y_rot_range[1], z_rot_range[1]],
                            [x_rot_range[1], y_rot_range[0], z_rot_range[1]],
                            [x_rot_range[1], y_rot_range[2], z_rot_range[1]],
                            [x_rot_range[1], y_rot_range[1], z_rot_range[0]],
                            [x_rot_range[1], y_rot_range[1], z_rot_range[2]]])
# The first row of rotation_platform is the neutral position. This is a
# reference for later algorithms.
rotation_platform = np.concatenate(([np.array([phi, theta, psi])],
                                    rot_eval_matrix), axis=0)

# Generating the Proximal endpoints in world coordinates
# and the Distal Endpoints in body coordinates:
A_i, b0_i = cdpr_dimensioning(lf=lf, hf=hf, wf=wf, lp=lp, wp=wp, hp=hp,
                              top_off_x=top_off_x, top_off_y=top_off_y,
                              top_off_z=top_off_z, bot_off_z=bot_off_z,
                              d_wf=d_wf, d_hf=d_hf,
                              n=n, m=m)

# Generating the initial values for statue collision
D_ik, signs_init =\
    cdpr_cable_platform_init(location_platform,
                             rotation_platform[:, 0],
                             b0_i, m)


# Generating coordinates to draw the CDPR in every graph:
success, draw_platform, B_inew, \
    f_cables, status = cdpr_kinematics(A_i, b0_i,
                                       location_platform,
                                       rotation_platform=rotation_platform[0, :],
                                       s_ij_all_old=None,
                                       cable_width=cable_width,
                                       signs_init=signs_init,
                                       n=n, m=m,
                                       wrench=wrench,
                                       fmin=fmin, fmax=fmax,
                                       ShuffleKey=ShuffleKey
                                       )

#  initializing the hull method, Load Icosahedron:
file = open("IcoFile0_4", "rb")
IcoFile1_4 = np.load(file)
vmatrix = IcoFile1_4['vmatrix_'+str(recur_lvl)]
index_matrix = IcoFile1_4['Index_Matrix_'+str(recur_lvl)]


hull_general = lambda eval_func:\
    hull_method(eval_func, A_i, b0_i, statue_base, vmatrix, index_matrix,
                                             signs_init,
                rotation_platform=rotation_platform, n=n, m=m, wrench=wrench,
                fmin=fmin, fmax=fmax, ShuffleKey=ShuffleKey,
                lambda_minInit=lambda_minInit,
                lambda_maxInit=lambda_maxInit, eps=eps, mid=mid,
                cable_width=cable_width)

fig1 = plt.figure(figsize=(10, 10))
ax1 = fig1.add_axes([0, 0, 1, 1], projection='3d', azim=30)
color = 'g'
title = f'{title_case}: WFW  ({title_pan_tilt})'
axis_lim = 0.5

cdpr_plot(A_i, B_inew, draw_platform, ax1, axis_lim)
plt.plot(starting_point[0], starting_point[1], starting_point[2], 'go')
plt.plot(mid[0], mid[1], mid[2], 'ro')
plt.xlabel('x')
plt.ylabel('y')
statue_plot(ax1, statue_base, statue_radius, statue_height)
```

References

Am4113t. (n.d.). Retrieved 2021-04-28, from https://www.dino-lite.com/products_detail.php?index_m1_id=9&index_m2_id=35&index_id=95

Aref, M. M., & Taghirad, H. D. (2008). Geometrical workspace analysis of a cable-driven redundant parallel manipulator: KNTU CDRPM. *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 1958–1963. doi: 10.1109/IROS.2008.4650670

Bak, J. H., Hwang, S. W., Yoon, J., Park, J. H., & Park, J. O. (2019). Collision-free path planning of cable-driven parallel robots in cluttered environments. *Intelligent Service Robotics*, *12*(3), 243–253. Retrieved from https://doi.org/10.1007/s11370-019-00278-7 doi: 10.1007/s11370-019-00278-7

Blanchet, L., & Merlet, J. P. (2014). Interference detection for cable-driven parallel robots (CDPRs). *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, 1413–1418. doi: 10.1109/AIM.2014.6878280

Bolboli, J., Khosravi, M. A., & Abdollahi, F. (2019). Stiffness feasible workspace of cable-driven parallel robots with application to optimal design of a planar cable robot. *Robotics and Autonomous Systems*, *114*(April 2019), 19–28. Retrieved from https://doi.org/10.1016/j.robot.2019.01.012 doi: 10.1016/j.robot.2019.01.012

Bouchard, S., Gosselin, C. M., & Moore, B. (2008). On the ability of a cable-driven robot to generate a prescribed set of wrenches. *Proceedings of the ASME Design Engineering Technical Conference*, *2*(PARTS A AND B), 47–58. doi: 10.1115/DETC2008-49518

Gagliardini, L., Caro, S., Gouttefarde, M., Wenger, P., & Girin, A. (2014). Optimal design of cable-driven parallel robots for large industrial structures. *Proceedings - IEEE International Conference on Robotics and Automation*(May), 5744–5749. doi: 10.1109/ICRA.2014.6907703

Gouttefarde, M., & Caro, S. (2018). Design of Reconfigurable Cable-Driven Parallel Robots Design of Reconfigurable Cable-Driven Parallel Robots Lorenzo Gagliardini , Marc Gouttefarde , Stéphane Caro HAL Id : hal-01758141. In E. Ottaviano, A. Pelliccio, & V. Gattulli (Eds.), *Mechatronics for cultural heritage and civil engineering* (pp. 85–113). Retrieved from https://hal.archives-ouvertes.fr/hal-01758141

Gouttefarde, M., Collard, J.-f., Riehl, N., Baradat, C., Gouttefarde, M., Collard, J.-f., ... Selection, G. (2015). Geometry Selection of a Redundantly Actuated Cable-Suspended Parallel Robot. *IEEE Transactions on Robotics*, *31*(2), 501–510.

Guay, F., Cardou, P., Lucia, A., Ruiz, C., Caro, S., Guay, F., ... How, M. (2018). Measuring How Well a Structure Supports Varying External Wrenches To cite this version : HAL Id : hal-01941645 Measuring How Well a Structure Supports Varying External Wrenches.

Kamali, K., Joubair, A., & Bonev, I. A. (2018). Optimizing cable arrangement in cable-driven parallel robots to improve the range of available wrenches. *Proceedings of the ASME Design Engineering Technical Conference*, *5B-2018*, 1–9. doi: 10.1115/DETC2018-85905

Kanishk_Verma. (2018, August). *Program to find the area and volume of icosahedron.* https://www.geeksforgeeks.org/program-to-find-the-area-and-volume-of-icosahedron/. ([Online; accessed November 03, 2020])

Kawamura, S., Choe, W., Tanaka, S., & Pandian, S. R. (1995). an Ultrahigh Speed Robot FALCON using Wire Drive System. *IEEE international Conference on Robotics and Automation*, 215–220.

Lafourcade, P., & Llibre, M. (2003). First steps toward a sketch-based design methodology for wire-driven manipulators. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, *1*(Aim), 143–148. doi: 10.1109/AIM.2003.1225086

Lafourcade, P., Llibre, M., & Reboulet, C. (2002). Design of a Parallel Wire-Driven Manipulator for Wind Tunnels. *Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators, October 3–4, 2002, Quebec City, Quebec, Canada*, 187–194.

Lessanibahri, S., Cardou, P., & Caro, S. (2019). Kinetostatic modeling of a cable-driven parallel robot using a tilt-roll wrist. *Mechanisms and Machine Science*, *74*, 109–120. doi: 10.1007/978-3-030-20751-9{_}10

Martin, A., Caro, S., & Cardou, P. (2018). Geometric determination of the cable-cylinder interference regions in the workspace of a cable-driven parallel robot. *Mechanisms and Machine Science*, *53*, 117–127. doi: 10.1007/978-3-319-61431-1{_}11

Nguyen, D. Q. (2014). On the Study of Large-Dimension Reconfiguable Cable-Driven Parallel Robots.

Nguyen, D. Q., & Gouttefarde, M. (2015). On the Improvement of Cable Collision Detection Algorithms. *Mechanisms and Machine Science*, *32*(March 2016). doi: 10.1007/978-3-319-09489-2

Otis, M. J., Perreault, S., Nguyen-Dang, T. L., Lambert, P., Gouttefarde, M., Laurendeau, D., & Gosselin, C. (2009). Determination and management of cable interferences between two 6-DOF foot platforms in a cable-driven locomotion interface. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, *39*(3), 528–544. doi: 10.1109/TSMCA.2009.2013188

Perreault, S., Cardou, P., Gosselin, C. M., & Otis, M. J. (2010). Geometric determination of the interference-free constant-orientation workspace of parallel cable-driven mechanisms. *Journal of Mechanisms and Robotics*, *2*(3), 1–9. doi: 10.1115/1.4001780

Perreault, S., & Gosselin, C. M. (2008). Cable-driven parallel mechanisms: Application to a locomotion interface. *Journal of Mechanical Design, Transactions of the ASME*, *130*(10), 1023011–1023018. doi: 10.1115/1.2965607

Pott, A. (2018). *Cable-driven parallel robots: Theory and application* (Vol. 120). doi: 10.1007/978-3-319-76138-1

Singh, A., Sahoo, C., & Parhi, D. R. (2015). Design of a planar cable driven parallel robot using the concept of Capacity Margin Index. *Proceedings of 2015 IEEE 9th International Conference on Intelligent Systems and Control, ISCO 2015*(2). doi: 10.1109/ISCO.2015.7282300

Specim iq technical specifications. (n.d.). Retrieved 2021-04-28, from `https://www.specim.fi/iq/tech-specs/`

Tadokoro, S., Murao, Y., Hiller, M., Murata, R., Kohkawa, H., & Matsushima, T. (2002). A motion base with 6-DOF by parallel cable drive architecture. *IEEE/ASME Transactions on Mechatronics*, *7*(2), 115–123. doi: 10.1109/TMECH.2002.1011248

Tadokoro, S., Nishioka, S., Kimura, T., Hattori, M., Takamori, T., & Maeda, K. (1996). On Fundamental Design of Wire Configurations of Wire-Driven Parallel Manipulators with Redundancy. In *Proceedings of the 1996 japan-usa symposium on flexible automation. part 2 (of 2) - boston, ma, usa* (pp. 151–158).

Tempel, P., Alfeld, M., & van der Wijk, V. (2020). Design and Analysis of Cable-Driven Parallel Robot CaRISA: A Cable Robot for Inspecting and Scanning Artwork Chapter. *ROMANSY 23 - Robot Design, Dynamics and Control*(September), 419–426. doi: 10.1007/978-3-030-58380-4

Unknown. (2015, October). *Icosahedron.py.* `https://pastebin.com/Ek87Kyhj`. ([Online; accessed October 20, 2020])

Verhoeven, R. (2003). Analysis of the Workspace of Tendon-Based. *Duisburg Gerhard Mercator University.* Retrieved from `http://duepublico.uni-duisburg-essen.de/servlets/DerivateServlet/Derivate-5601/verhoevendiss.pdf`

Yi, W., Zheng, Y., Wang, W., Tang, X., Liu, X., & Meng, F. (2019). Optimal Design and Force Control of a Nine-Cable-Driven Parallel Mechanism for Lunar Takeoff Simulation. *Chinese Journal of Mechanical Engineering (English Edition)*, *32*(1). Retrieved from `https://doi.org/10.1186/s10033-019-0382-2` doi: 10.1186/s10033-019-0382-2

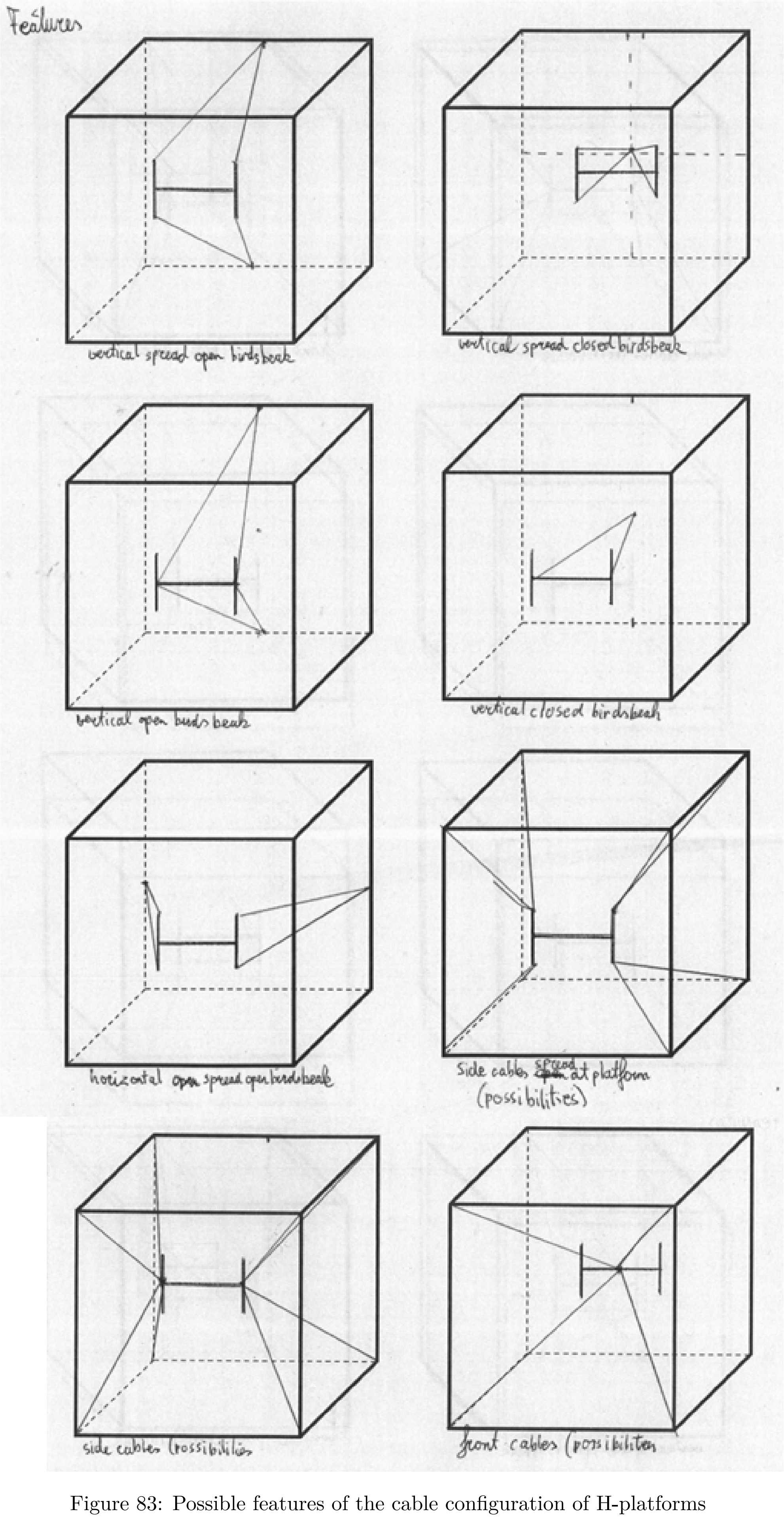

Features
vertical spread open birdsbeak
vertical spread closed birdsbeak
vertical open birds beak
vertical closed birdsbeak
horizontal open spread open birdsbeak
Side cables spread open at platform (possibilities)
side cables (possibilities
front cables (possibilities

Figure 83: Possible features of the cable configuration of H-platforms

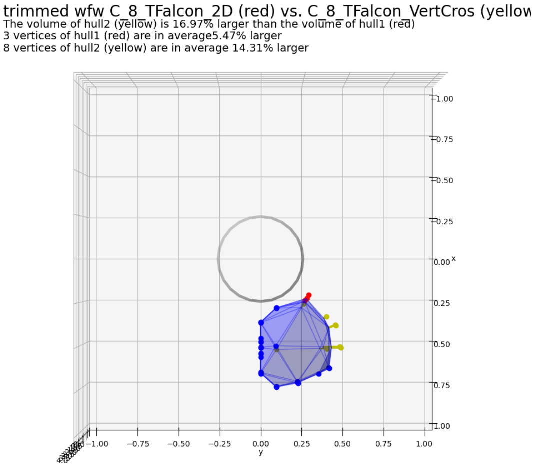


Figure 84: A vertical crossing does improve the WFW compared with a 2D lay-out.

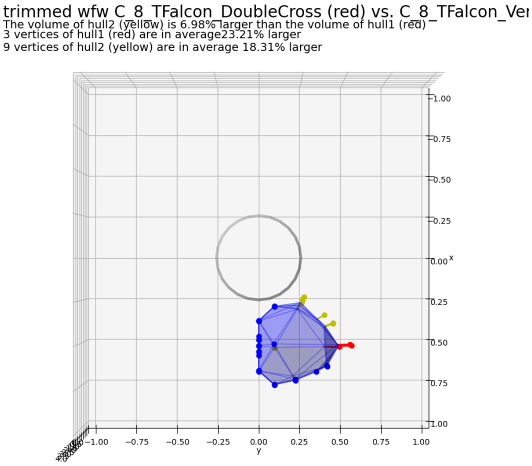


Figure 85: A double crossing is not better than only vertical crossing.

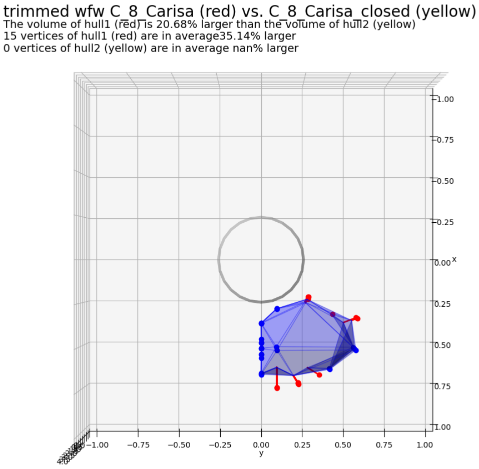


Figure 86: A double crossing is in the case of the CaRISA configuration better than only vertical crossing.

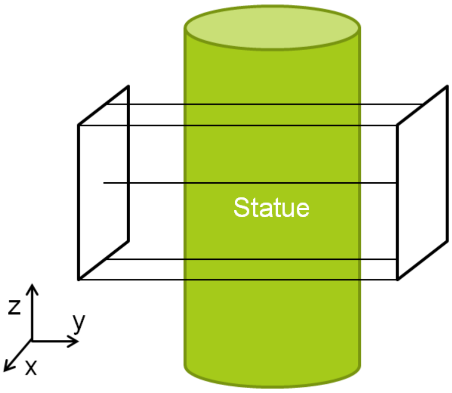


Figure 87: Schematic overview of the C-Platform and its orientation with respect to the statue. Possible cable attachment points are numbered according to the frame attachment points in Figure 72.

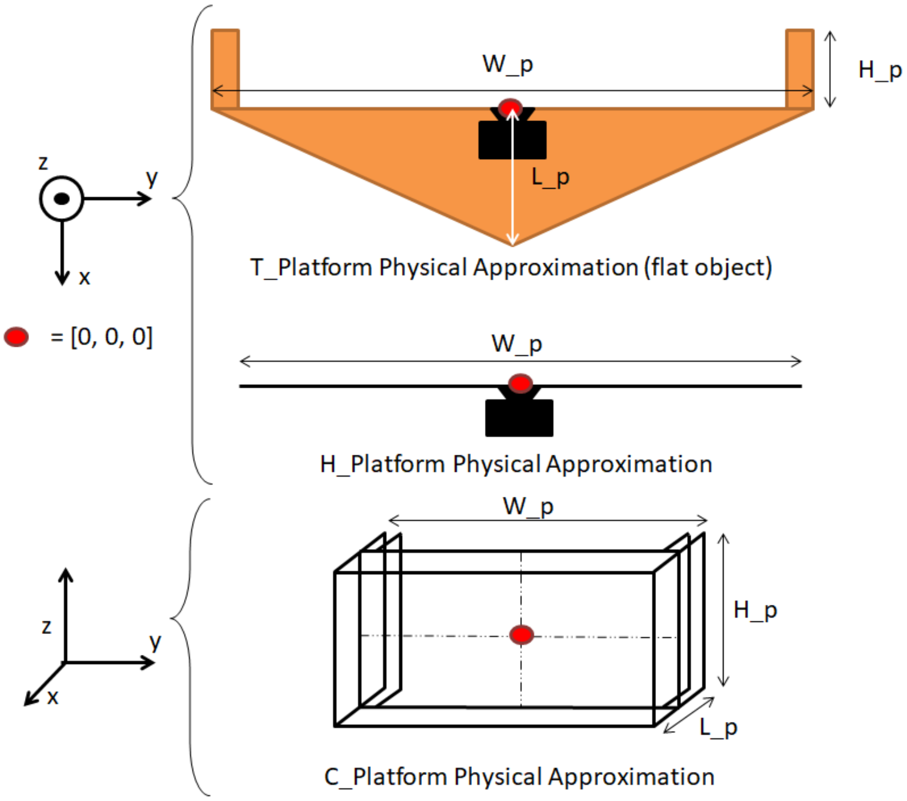
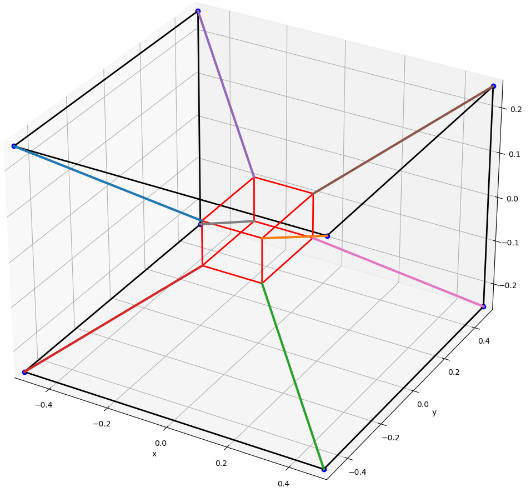

W_p
H_p
L_p
z
y
x
T_Platform Physical Approximation (flat object)
= [0, 0, 0]
W_p
H_Platform Physical Approximation
W_p
H_p
L_p
z
y
x
C_Platform Physical Approximation

Figure 88: The physical approximation of the different platforms



Most important Code

September 17, 2021

TU Delft
Delft, Netherlands
Mees Waal

Contents

1 Hull Method

```
import numpy as np
import matplotlib.pyplot as plt
import math
# from mpl_toolkits.mplot3d import axes3d
from scipy.spatial.transform import Rotation
from scipy.optimize import linprog
import itertools as it
import scipy
def hull_method(eval_func, A_i, b0_i, statue_base, statue_radius,
                vmatrix, index_matrix, signs_init,
                rotation_platform, n=6, m=8,
                wrench=np.linspace(0, 0, 6), fmin=5, fmax=150,
                ShuffleKey=np.array([0, 1, 2, 3, 4, 5, 6, 7]),
                lambda_minInit=True, lambda_maxInit=0.5, eps=0.0001,
                mid=np.array([0, 0, 0]), cable_width=0.005):

    Vertices_Amount = len(vmatrix[:, ])
    # print(f'The amount of vertices is: {Vertices_Amount}')
    # print(f'The amount of vertex indices is: {len(index_matrix[:,])*3}')
    workspace_vertices = np.zeros([3, Vertices_Amount])
    # Creating an initial value for s_ij_all, corresponding to the
    # situation in the middle of the Icosahedron.
    s_ij_all_old = None # made None here and not in the rotation for loop,
    # this way only the neutral position has a s_ij_all_old None.
    for irot in np.arange(1, len(rotation_platform[:, 0])):
        rotation_change = rotation_platform[irot, :]
        # smart orientation algorithm here for z_rotation,
        statue_main_angle = rotation_tostatue(statue_base, mid)
        rotation_change_new = rotation_change + np.array([0, 0, statue_main_angle]
                                                 )  # Wanted rotation is adjusted
                                                 with the necessary rotation of
                                                 the camera towards the statue))
        rotation_eval = rotation_platform[0, :] + rotation_change_new # To make
                                                 sure the x and y angles are
                                                 adapted to the new rotation of
                                                 the robot.
        mid_shifted = camera_pantilt_loc_shift(rotation_change, mid,
                                                 statue_main_angle,
                                               statue_radius)
        junkout = eval_func(A_i, b0_i,
                            mid_shifted,
                            rotation_eval,
                            s_ij_all_old=s_ij_all_old,
                            signs_init=signs_init,
                            cable_width=cable_width,
                            n=n, m=m,
                            wrench=wrench,
                            fmin=fmin, fmax=fmax,
                            ShuffleKey=ShuffleKey
                            )
        if irot == 1 and junkout[0]:
            # When the first iteration (the neutral position) is
            # accepted the old s_ij_all_old is first stored in
            # s_ij_all_old_last_accepted_position. Then a new one
            # is defined, to be used for the other rotations.
            s_ij_all_old = junkout[1] #needed for the next iterations in this for
                                                     loop
            s_ij_all_old_0 = s_ij_all_old #needed for the initial s_ij_all_old
        if not junkout[0]:
            # When there is an orientation for which no rotation
            # is possible: break and success will be false.
            # the s_ij_all_old_accepted_position is
            # turned into s_ij_all_old again
            print('NO! Starting position is not admissible to begin with')
            for vertex in range(len(workspace_vertices[0, :])):
                workspace_vertices[:, vertex] = mid
            return workspace_vertices, index_matrix, 0, 0, mid
    s_ij_all_old = s_ij_all_old_0

    lambda_store = np.zeros(Vertices_Amount)
    lambda_maxInit = lambda_maxInit * np.ones([Vertices_Amount])
    for i in range(Vertices_Amount):
        # print(f'vertex = {i}')
        # if (i % 5) == 0:
        #     print(f'{int(((i + 1) / Vertices_Amount) * 100)}%')
        if i != 999:
            v = np.array(vmatrix[i, :])
            # print(f'v is: {v}')
            lambda_min = 0
            lambda_max = lambda_maxInit[i]
            s_ij_all_old = s_ij_all_old_0
            for count in range(1000):
                # print(f'iteration = count}')
                lambda_it = 0.5 * (lambda_max + lambda_min)
                x = mid + lambda_it * v
                # print(lambda_it * v)
                # The for loop that loops through all the rotations is found
                # below:
                for irot in np.arange(1, len(rotation_platform[:, 0])):
                    rotation_change = rotation_platform[irot, :]
                    # smart orientation algorithm here for z_rotation,
                    statue_main_angle = rotation_tostatue(statue_base, x)
                    rotation_change_new = rotation_change + np.array([0, 0,
                                                             statue_main_angle])
                                                             # Wanted rotation is
                                                             adjusted with the
                                                             necessary rotation of
                                                              the camera towards
                                                             the statue))
                    rotation_eval = rotation_platform[0, :] + rotation_change_new
                                                             # To make sure the x
                                                             and y angles are
                                                             adapted to the new
                                                             rotation of the robot
                                                             .
                    x_shift = camera_pantilt_loc_shift(rotation_change, x,
                                                             statue_main_angle,
                                                       statue_radius)
                   out = eval_func(A_i, b0_i,
                                   x_shift,
                                   rotation_eval,
                                   s_ij_all_old,
                                   signs_init,
                                   cable_width,
                                   n=n, m=m,
                                   wrench=wrench,
                                   fmin=fmin, fmax=fmax,
                                   ShuffleKey=ShuffleKey
                                   )
                   if irot == 1:
                       # When the first iteration (the neutral position) is
                       # accepted the old s_ij_all_old is first stored in
                       # s_ij_all_old_last_accepted_position. Then a new one
                       # is defined, to be used for the other rotations.
                       s_ij_all_old_last_accepted_position = s_ij_all_old
                       if out[0]:
                           s_ij_all_old = out[1]
                   if not out[0]:
                       # When there is an orientation for which no rotation
                       # is possible: break and success will be false.
                       # the s_ij_all_old_accepted_position is
                       # turned into s_ij_all_old again
                       s_ij_all_old = s_ij_all_old_last_accepted_position
                       break
               # end of for loop that loops through all rotations.
               success = out[0]
               # print(success)
               if success:
                   lambda_min = lambda_it
                   if count == 0 and lambda_minInit:
                       #this part stops the iteration of the line search
                       #because the edge of the trimmed matrix is already
                       #investigated. lambda_minInit is only 0.001 for
                                                                 cable_cable_interference

                       workspace_vertices[:, i] = mid + lambda_min * v
                       lambda_store[i] = lambda_min
                       break
               else:
                   lambda_max = lambda_it
               # Stop condition - convergence:
               if lambda_max - lambda_min < eps:
                   # The 3 rows of workspace_vertices represent the x,y,z
                   # coordinate, the 12 columns represent the vertices.
                   workspace_vertices[:, i] = mid + lambda_min * v
                   lambda_store[i] = lambda_min
                   break
               # Stop condition - maximum iteration reached:
               if count == 999:
                   print(f'The linear Search did not converge to a\
                         solution for vertex {i}')
                   lambda_store[i] = lambda_min

   volume_workspace = 0
   area_workspace = 0
   cog_sum = np.array([0.0, 0.0, 0.0])
   for i in range(len(index_matrix[:, 0])):
       va = np.array(workspace_vertices[:, index_matrix[i, 0]])
       vb = np.array(workspace_vertices[:, index_matrix[i, 1]])
       vc = np.array(workspace_vertices[:, index_matrix[i, 2]])
```

```
        area_workspace += 0.5 * np.linalg.norm(np.cross((va - vb), (va - vc)))
        partial_volume = (1 / 6) * np.dot(np.cross((va - mid),
                                                   (vb - mid)), (vc - mid))
        volume_workspace += partial_volume
        cog_sum += (va + vb + vc + mid)\
            * partial_volume
    cog = (0.25 / volume_workspace) * cog_sum
    return workspace_vertices, index_matrix, volume_workspace, \
        area_workspace, cog
```

2 Optimization Command

```
result_optimization = scipy.optimize.differential_evolution(workspace_calculator,
                                         Design4Optimization['bounds'],
                                      args=(Design4Optimization, {'Nfeval':0}),
                                      strategy='best1bin',
                                      maxiter=300,
                                      popsize=15,
                                      tol=0.01,
                                      mutation=(0.5, 1),
                                      recombination=0.7,
                                      seed=3, callback=None,
                                      disp=True, polish=False,
                                      init='latinhypercube', atol=0,
                                      updating='immediate', workers=1)
```

3 WFW Calculation

```
result_optimization = scipy.optimize.differential_evolution(workspace_calculator,
                                         Design4Optimization['bounds'],
                                      args=(Design4Optimization, {'Nfeval':0}),
                                      strategy='best1bin',
                                      maxiter=300,
                                      popsize=15,
                                      tol=0.01,
                                      mutation=(0.5, 1),
                                      recombination=0.7,
                                      seed=3, callback=None,
                                      disp=True, polish=False,
                                      init='latinhypercube', atol=0,
                                      updating='immediate', workers=1)
```

4 Cable-Cable Interference Calculation

```
def cdpr_cablecable_nguy_perreault(A_i, b0_i,
                                   location_platform,
                                   rotation_platform,
                                   s_ij_all_old,
                                   signs_init,
                                   cable_width,
                                   mid=np.array([0, 0, 0]),
                                   n=6, m=8,
                               wrench=np.linspace(0, 0, 6),
                               fmin=5, fmax=150,
                               ShuffleKey=np.array([0, 1, 2, 3,
                                                    4, 5, 6, 7]),
                               ):
"""
This method combines both the cdpr_cablecable_nguy and
cdpr_cablecable_perreault method. By first checking the Nguyen conditions
and then checking whether the cables are so close to eachother they
collide with their diameters.

Mind that when proximal/distal endpoint are within one cable_width
close to each other, the cables are assumed not to collide
"""
# Setting up basic equations to calculate cables:
RE = Rotation.from_euler('xyz', rotation_platform, degrees=True)
R = RE.as_matrix()
# xr_p, yr_p and zr_p represent the distance to the centre of the platform
r = [location_platform[0] * np.ones(m),
     location_platform[1] * np.ones(m),
     location_platform[2] * np.ones(m)]
# The B_i is presenting the old geometry before shuffling and is used for
# plotting
B_i = R @ b0_i + r
B_inew = cable_shuffle(B_i, ShuffleKey)

l_i = B_inew - A_i
n_i_unit = np.zeros_like(l_i)
for mi in range(m):
    n_i_unit[:, mi] = l_i[:, mi] / np.linalg.norm(l_i[:, mi])
success = True
# Setting up the for-loop to evaluate all cable pairs:
# for cable_collisions in ShuffleKey['cable_indices_cc_collision']:
#     # this for loop loops through different cable groups
#     s_ij_all_old = 1
#     if not success:
#         break
# cable_collisions = ShuffleKey['cable_indices_cc_collision']
# cable_indices = np.array(cable_collisions)
# combinations = list(it.combinations(cable_indices, 2))
combinations = ShuffleKey['cable_indices_cc_collision']
n_all_values = np.zeros([3, len(combinations)])
s_ij_all = np.zeros(len(combinations))
for i in range(len(combinations)):
    if i != 999:
        n_1_index = combinations[i][0]
        n_2_index = combinations[i][1]
        n_i_unit_1 = n_i_unit[:, n_1_index]
        n_i_unit_2 = n_i_unit[:, n_2_index]
        A_1 = A_i[:, n_1_index]
        A_2 = A_i[:, n_2_index]
        B_1 = B_inew[:, n_1_index]
        B_2 = B_inew[:, n_2_index]
        # ---------------------------------------------------------------------
        # Perreault part
        cable_length_1 = np.linalg.norm(l_i[:, n_1_index])
        cable_length_2 = np.linalg.norm(l_i[:, n_2_index])
        n_r_unit = np.cross(n_i_unit_1, n_i_unit_2) \
            / np.linalg.norm(np.cross(n_i_unit_1, n_i_unit_2))
        # print(n_r_unit)
        distance_distal_endpoints = A_2 - A_1
        a_matrix = np.zeros([3, 3])
```

```
for fill_i in range(3):
    a_matrix[fill_i] = n_i_unit_1[fill_i],\
        -1 * n_i_unit_2[fill_i],\
        n_r_unit[fill_i]
# n_all_values: in a column n_val_1, n_val_2 and n_val_r
# can be found respectively (Perreault, 2008)

# filtering out the parallel cases:
if not np.linalg.norm(np.cross(l_i[:, combinations[i][0]],
                               l_i[:, combinations[i][1]]))\
        < 0.001:
    n_all_values[:, i] = np.linalg.solve(a_matrix,
                                         distance_distal_endpoints)
else:
    cable_distance = np.linalg.norm(np.cross(n_i_unit_1,
                                             (A_2 - A_1)))
    # the values in the next line make sure, the cables are not
    # detected as colliding in the perreault step
    n_all_values[:, i] = np.array([0, 0, cable_distance])
# ---------------------------------------------------------------

# Determening how the cables are oriented with
# respect to each other. (Nguyen, 2015)
A_1_B_1 = B_1 - A_1
A_1_B_2 = B_2 - A_1
A_1_A_2 = A_2 - A_1
B_1_B_2 = B_2 - B_1
s_ij_all[i] = np.sign(n_all_values[2, i])
# This sign change matters during the iteration of the hull method.
# Here the value of Perreaults method can easily be used to
# retrieve the sign.
# The first iteration doesn't use an old value of s_ij_all
# This first iteration is just meant for comparison.
if s_ij_all_old is None:
    s_ij_all_old = s_ij_all

# projecting the cables onto eachother and finding
# point of intersection
plane_n = np.cross(A_1_B_1, A_1_B_2)
if plane_n[0] != 0\
    or plane_n[1] != 0\
        or plane_n[2] != 0:
    t = (plane_n @ A_1_A_2) / (plane_n @ plane_n)
    A_2_proj = A_2 - (t * plane_n)
    A_2_proj_B_2 = B_2 - A_2_proj
else:
    # In this case plane_n is all zero,
    # which means A_1_B_1 = A_2_B_2
    # this means the cables cannot intersect so values are given
    # so an intersection will not be detected. The line of code
    # below makes d_ij 0 but this is allowed.
    A_2_proj_B_2 = B_2 - A_2

# Check at which point they are intersecting: within the
# cable lengths?
# Checking the intersection comes from Perreault:\
# Geometric Determination... Eq. 12
C_ij = np.zeros([3, 2])
for ic in range(3):
    C_ij[ic, 0] = -1 * A_1_B_1[ic]
    C_ij[ic, 1] = A_2_proj_B_2[ic]
C_ij_mp = scipy.linalg.pinv(C_ij)
```

```
            d_ij = C_ij_mp @ B_1_B_2
            # print('the position to be evaluated is:', location_platform)
            # print('its sign is:', s_ij_all[i])
            # When d_ij is lower than 1 for both cables, there is a\
            # crossing within the projection
            if np.linalg.norm(np.cross((B_1 - A_1), (B_2 - A_2)))\
                    < cable_width:
                # The condition above checks whether the cables are parallel
                success = True
            elif np.linalg.norm(A_i[:, n_1_index] - A_i[:, n_2_index])\
                <= cable_width\
                    or np.linalg.norm(B_inew[:, n_1_index]
                                      - B_inew[:, n_2_index]) <= cable_width:
                # The condition above checks whether the cables share a distal
                # or proximal endpoint. In that case there can't be a collision
                success = True
            elif (s_ij_all[i] == 0
                    and 0 < d_ij[0] <= 1
                    and 0 < d_ij[1] <= 1):
                # This conditions means the cables are crossing at
                # that moment
                success = False
                # print(f"""The function breaks for sign {d_ij} at
                # {s_ij_all[i]} between cables {combinations[i]}""")
                break
            elif (s_ij_all[i] != s_ij_all_old[i] and
                  0 < d_ij[0] <= 1 and 0 < d_ij[1] <= 1):
                # This condition means the cables have crossed
                success = False
                # print(f'The function breaks for {s_ij_all[i]}\
                #       between cables {combinations[i]}')
                # print(f'The function breaks for {d_ij}\
                #       between cables {combinations[i]}')
                break
            elif (abs(n_all_values[2, i]) <= cable_width and
                  0 < n_all_values[0, i] <= cable_length_1 and
                  0 < n_all_values[1, i] <= cable_length_2):
                success = False
                # print(f'The function breaks for {s_ij_all[i]}\
                #       between cables {combinations[i]} (Perreault)')
                # print(f'The function breaks for {d_ij}\
                #       between cables {combinations[i]} (Perreault')
                break
            else:
                success = True
        # if i == 27:
            # print(f'The sign for cablepair 1,7 is: {s_ij_all[25]}\
            #       and distances are:{d_ij}')
    if success:
        s_ij_all_old = s_ij_all
    return success, s_ij_all_old
```