Czech technical university in Prague
Faculty of mechanical engineering

BACHELOR THESIS

Cable-driven Parallel Manipulator With Adjustable Platform Geometry

2020 **Juraj Lieskovský**

ČVUT
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Lieskovský** Jméno: **Juraj** Osobní číslo: **473729**

Fakulta/ústav: **Fakulta strojní**

Zadávající katedra/ústav: **Ústav mechaniky, biomechaniky a mechatroniky**

Studijní program: **Teoretický základ strojního inženýrství**

Studijní obor: **bez oboru**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Paralelní lanový manipulátor s měnitelnou geometrií platformy

Název bakalářské práce anglicky:

Cable-driven parallel manipulator with adjustable platform geometry

Pokyny pro vypracování:

1) Seznamte se s problematikou lanových manipulátorů.
2) Vytvořte kinematický simulační model paralelního lanového manipulátoru.
3) Do modelu zahrňte kontrolu kolizí a singulárních poloh.
4) Ověřte rozšíření pracovního prostoru vlivem měnitelné geometrie platformy.

Seznam doporučené literatury:

[1] Kraus, K.: Redundantní lanový manipulátor, Diplomová práce, FS ČVUT v Praze, Praha, 2016.
[2] Merlet, J.-P.: On the redundancy of cable-driven parallel robots, in 5th European Conf. on Mechanism Science (Eucomes), Guimares, pp. 31–39, 2014.
[3] Pott, A., Bruckmann, T. (eds): Cable-Driven Parallel Robots, CableCon 2019, Mechanisms and Machine Science, Volume 74. Springer, ISBN 978-3-030-20750-2.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Petr Beneš, Ph.D., odbor mechaniky a mechatroniky FS

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **30.04.2020** Termín odevzdání bakalářské práce: **07.08.2020**

Platnost zadání bakalářské práce: ______________

Ing. Petr Beneš, Ph.D. podpis vedoucí(ho) práce	doc. Ing. Miroslav Španiel, CSc. podpis vedoucí(ho) ústavu/katedry	prof. Ing. Michael Valášek, DrSc. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

.
Datum převzetí zadání Podpis studenta

Annotation sheet

Title: Cable-driven parallel manipulator with adjustable platform geometry

Název: Paralelní lanový manipulátor s měnitelnou geometrií platformy

Author: Juraj Lieskovský

Academic year: 2019/2020

Study program: Theoretical fundamentals of mechanical engineering

Department: Department of mechanics, biomechanics and mechatronics

Supervisor: Ing. Petr Beneš, Ph.D.

Scope of work: Pages: 57
Figures: 18
Attachments: 1 CD

Keywords: cable-driven, parallel manipulator, self collision, singularities

Klíčová slova: paralelní lanový manipulátor, kolize, singularity

Abstract: This thesis deals with the issue of cable-driven parallel manipulators with an emphasis on their workspace restrictions, particularly singularities and self collisions. Further it examines the possibility of extending the workspace of a spacial cable-driven parallel manipulator through adjustable platform geometry confirming it by a series of simulations.

Abstrakt: Práce se zabývá problematikou paralelních lanových manipulátorů se zaměřením na omezení jejich pracovních prostorů, především singularit a kolizí. Dále zkoumá možnost rozšíření pracovního prostoru paralelního lanového manipulátorů vlivem měnitelné geometrie platformy, která je potvrzena sérií simulací.

I declare that I carried out this bachelor thesis independently, and only with the cited sources

In Prague ....

Juraj Lieskovský

I would like to thank Ing. Petr Beneš, Ph.D. for all the invaluable advice, my parents for supporting me throughout my studies and everyone who was of help in ways big and small during the writing of this thesis.

Contents

List of Figures

List of Listings

List of Tables

Chapter 1

Introduction

J-P. Merlet defines a generalized parallel manipulator as a *"closed-loop kinematic mechanism whose end-effector is linked to the base by several independent kinematic chains"* in [4]. Such manipulators are becoming advantageous in comparison to their serial counterparts as negatives such as more complex kinematic relations and the problem of regulating mechanisms with actuation redundancies are being overcome.

Actuators of parallel mechanisms are most often fixed to the frame reducing the mass of links manipulating the platform upon which the end-effector is mounted resulting in high wrench-to-weight ratios. *Cable-driven parallel manipulators (CDPMs)* are exceptional in this aspect with the mass of their cables often not being considered in simulation models. They further appeal with large and adaptable workspaces as well as low cost.

In this thesis we will focus on such manipulators with an emphasis on their workspace restrictions, particularly *singularities* and *self collisions*. The extension of a spacial *CDPMs'* workspaces through adjustable platform geometry is also explored. For this purpose a kinematic model with the added capabilities of detecting both *singularities* and *self collisions* is created and a series of simulations performed.

1.1 Classification of *CDPMs*

A unique characteristic of *CDPMs* is the uni-directional nature of forces exerted by the cables onto the platform. In consequence manipulators driven by cables only are classified in [5], according to the number of cables k and the desired degrees of freedom (DOF) of the end-effector N into:

- *incompletely restrained positioning mechanisms (IRPMs)*: $k \leq N$
- *completely restrained positioning mechanisms (CRPMs)*: $k = N + 1$
- *redundantly restrained positioning mechanisms (RRPMs)*: $k > N + 1$

While *CRPMs* and *RRPMs* are capable of exerting wrenches $\boldsymbol{w}$ spanning $\mathbb{R}^6$ in all non-singular configurations (section 2.3.1), *IRPMs* rely on gravity and other external forces to determine the resulting pose of the manipulator. This reliance limits maximal acceleration of the manipulator as well as wrenches it is capable of exerting.

1.2 Redundancy of parallel manipulators

Depending on the number of actuators m and the number of DOF of the mechanism n two types of redundancy in parallel manipulators can be distinguished [4]:

- *kinematic redundancy:* $n > N$
- *actuation redundancy:* $m > n$

All *CRPMs* and *RRPMs* are by definition actuation redundant adding complexity to the problem of direct kinematics and force distribution among the cables of *CDPMs*. Kinematic redundancies on the other hand are rarely present in parallel manipulators but can be generally used to extend their workspaces.

1.3 Aim and motivation

While planar *CRPMs* such as the *SkyCam* (figure 1.1), and spacial *RRPMs* operating in singular configurations (see section 2.3.1) such as the *CoGiRo* [6] are finding use, spacial *RRPMs* operating in non-singular configurations for example robots from the *IPAnema* family (figure 1.2) are struggling to find an industry application largely due to cable interference.

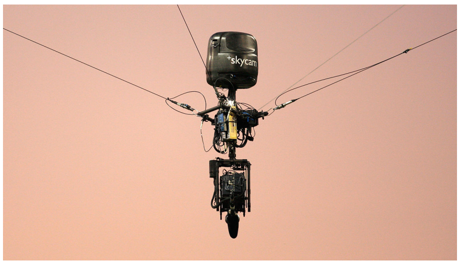


Figure 1.1: SkyCam - photo by Despeaux - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=7852833

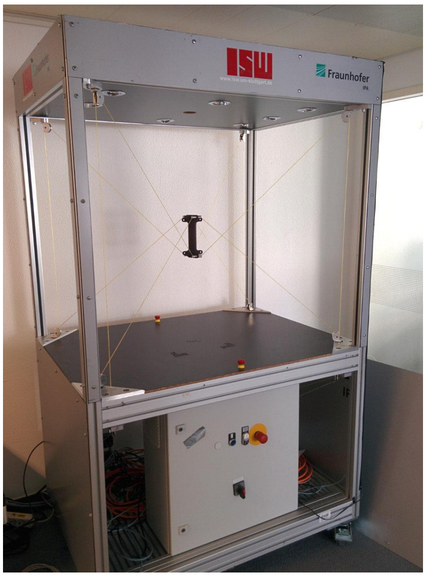


Figure 1.2: IPAnema mini - example of a *RRPM* operating in non-singular configurations [1]

We will explore the possibility of extending the workspace of a *RRPMs* by adding kinematic redundancy to their mechanisms. This will be done on a manipulator of novel design, capable of exerting wrenches $\boldsymbol{w}$ spanning $\mathbb{R}^6$ outside of the typical workspace delimited by points through which its cables exit the workspace, with the kinematic redundancy being introduced in the form of a platform with adjustable geometry. By such design cable interference with objects in the extended workspace should be largely avoided.

1.4 Structure of the thesis

First we will overview kinematics and statics of *CDPMs* as well as explore the limiting factors of their workspaces: singularities, joint limits and self collision, drawing from literature.

Then in chapter 3 we will propose the manipulator subject to this thesis and describe both its mechanical and collision model. The kinematic relations and equations of static equilibrium will be derived allowing for the identification of singular configurations. Further we will describe an applicable method of collision detection along with a novel algorithm necessary for its execution.

In order to validate the extension of the manipulator's workspace a simulation model capable of detecting self collisions and singularities in a given pose will be created in *Matlab* and described in chapter 4. The chapter will also contain the descriptions and results of simulations performed on the model confirming the claim.

Chapter 2

Exploring the topic

2.1 Kinematics of *CDPMs*

Let $\boldsymbol{q} \in \mathbb{R}^m$ be a vector of actuated joint parameters and $\boldsymbol{X} \in \mathbb{R}^n$ a vector of unique parameters describing any particular pose of a manipulator.

Generally each cable of a *CDPM* is seperately actuated, therefore $k = m$. Most *CDPMs* in application are *CRPMs* or *RRPMs* and therefore by definition actuation redundant. If $m > n$, as for all acutation redundant manipulators, the systems of equations describing *direct kinematics* of the mechanism mapping $\boldsymbol{q} \rightarrow \boldsymbol{X}$ and $\dot{\boldsymbol{q}} \rightarrow \dot{\boldsymbol{X}}$ are not functions. For this reason only *inverse kinematics* of *CDPMs* are commonly described in literature [7], [8].

When determining inverse kinematics, acutated joint parameters q_i, $i \in \langle 1, k \rangle$ of actuation redundant mechanisms are generally functions of $\boldsymbol{X}$ for all $\boldsymbol{X} \in \mathbb{R}^n$:

$$q_i = f(\boldsymbol{X}), \qquad i \in \langle 1, k \rangle \tag{2.1.1}$$

While the map $\dot{\boldsymbol{X}} \rightarrow \dot{\boldsymbol{q}}$ most often written in form:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{X})\, \dot{\boldsymbol{X}} \tag{2.1.2}$$

where $\boldsymbol{J}(\boldsymbol{X})$ is the *inverse kinematics Jacobian* [8]:

$$\boldsymbol{J}(\boldsymbol{X}) = \begin{bmatrix} \frac{\partial q_1}{\partial X_1} & \cdots & \frac{\partial q_1}{\partial X_n} \\ \vdots & \ddots & \\ \frac{\partial q_m}{\partial X_1} & & \frac{\partial q_m}{\partial X_n} \end{bmatrix} \tag{2.1.3}$$

is a function only on a subset of $\boldsymbol{X} \in \mathbb{R}^n$ where $rank(\boldsymbol{J}) = n$ resulting in the occurence of *Jacobian singularities* described in section 2.3.1.

2.2 Statics of *CDPMs*

In [5],[7],[8] equations of static equilibrium of *CDPMs* are expressed in the form:

$$\boldsymbol{w} = \boldsymbol{A}(\boldsymbol{X})\,\boldsymbol{\tau} \tag{2.2.1}$$

where $\boldsymbol{\tau}$ is a vector of cable tension, $\boldsymbol{w}$ is the sum of external wrenches exerted onto the end-effector and $\boldsymbol{A}(\boldsymbol{X}) \in \mathbb{R}^{N \times k}$ is the *structure matrix* of the *CDPM.*

Determining the tension vector $\boldsymbol{\tau}$ to a given $\boldsymbol{w}$ is the core problem of dynamic control of *CDPMs.* As *CRPMs* and *RRPMs* are actuation redundant there are infinite tension vectors $\boldsymbol{\tau}$ compensating the wrench $\boldsymbol{w}$ in all non-singular poses of the manipulator $\boldsymbol{X}$. Various methods of calculating a favourable vector $\boldsymbol{\tau}$ exist, considering such factors as cable elasticity [9] or slack [10]. In [8] Lamaury and Gouttefarde describe the *Mikelsons' barycenter approach* to calculating the vector $\boldsymbol{\tau}$ as:

$$\boldsymbol{\tau} = \boldsymbol{A}^{+}\boldsymbol{w} + \boldsymbol{N}\boldsymbol{\lambda} \tag{2.2.2}$$

where $\boldsymbol{A}^{+}$ is the Moore-Penrose (pseudo-inverse) inverse matrix of $\boldsymbol{A}$, providing the least-square solution of a system of linear equations without a unique solution [11], $\boldsymbol{N} = null(\boldsymbol{A})$ and $\boldsymbol{\lambda} = \begin{bmatrix}\lambda_1 & \lambda_2\end{bmatrix}^T$ is an arbitrary vector using which the minimal or maximal tension in the cables is adjusted.

For the purposes of this paper, the form of the *structure matrix* $\boldsymbol{A}$ will need to be derived in order to analyse *force-closure singularities* described in section 2.3.1.

2.3 Workspace restrictions

As desrcibed by J.-P. Merlet in [12], the workspace of a parallel manipulator is restricted by three factors.

- singular configurations
- self collision
- joint limits

2.3.1 Singular configurations

Xiumin Diao and Ou Ma in [13] classify two types of singularities, occurring in cable-driven parallel manipulators:

- *Jacobian singularities*
- *force-closure singularities*

Jacobian singularities

Jacobian singularities of *CDPM* occur in such configurations $\boldsymbol{X}$ of the manipulator in which the *inverse kinematics Jacobian*:

$$\boldsymbol{J}(\boldsymbol{X}) = \begin{bmatrix} \frac{\partial q_1}{\partial X_1} & \cdots & \frac{\partial q_1}{\partial X_n} \\ \vdots & \ddots & \\ \frac{\partial q_m}{\partial X_1} & & \frac{\partial q_m}{\partial X_n} \end{bmatrix} \tag{2.3.1}$$

becomes rank-deficient [13] resulting in the loss of one or more DOF. In addition near Jacobian singular configurations a small change in configuration $\dot{\boldsymbol{X}}$ may require almost infinite joint velocities $\dot{\boldsymbol{q}}$. Because of the aforementioned reasons avoiding Jacobian singularities through both trajectory planning and manipulator design is preferable.

Force-closure singularities

CRPM and *RRPM* are able to resist all wrenches $\boldsymbol{w} \in \mathbb{R}^N$ applied to its platform, under the condition that column vectors of the sturcture matrix $\boldsymbol{A}(\boldsymbol{X})$ span $\mathbb{R}^N$ and tensions in the calbes τ_i , $i \in \langle 1, k \rangle$ are positive.

Following Xiumin Diao and Ou Ma in [7], we may express the force-closure condition of a *CDPM* as:

$$\forall\ \boldsymbol{w} \in \mathbb{R}^N, \quad \exists\ \boldsymbol{\tau} > \boldsymbol{0}, \quad \ni \boldsymbol{A\tau} = \boldsymbol{w} \tag{2.3.2}$$

The condition 2.3.2 is satisfied if and only if positive linear combinations column vectors of $\boldsymbol{A}$, span $\mathbb{R}^N$ [14]. Further we will refer to such property of a set of vectors as *positive span.*

Let $\boldsymbol{a}_i$, $i \in \langle 1, N \rangle$ be the row vectors of $\boldsymbol{A}(\boldsymbol{X}) = \begin{bmatrix} \boldsymbol{a}_1 & \dots & \boldsymbol{a}_N \end{bmatrix}^T$, column vectors of $\boldsymbol{A}(\boldsymbol{X})$ *positively span* $\mathbb{R}^N$ if:

$$rank(\boldsymbol{A}) = N \ \wedge\ (\,min(\boldsymbol{a}_i) < 0 \ \wedge\ max(\boldsymbol{a}_i) > 0\,)\,,\ \forall\, i \in \langle 1, N \rangle \tag{2.3.3}$$

If the condition 2.3.3 is not satisfied, the manipulator is unable resist all wrenches $\boldsymbol{w} \in \mathbb{R}^N$ and the pose of the manipulator $\boldsymbol{X}$ is considered *force-closure singular.*

An alternative method for verifying the force-closure condition is proposed in [7]. It is based on the calculation of $\boldsymbol{V} = \hat{\boldsymbol{A}}_p^{-1}$, where $\hat{\boldsymbol{A}}_p \in \mathbb{R}^{N \times N}$ consists of six linearly independent column vectors of $\boldsymbol{A}$ and the consequent transformation of $\boldsymbol{A}$ by multiplication with $\boldsymbol{V}^T$:

$$\boldsymbol{A}' = \boldsymbol{V}^T \boldsymbol{A}\,, \quad rank(\boldsymbol{A}') = rank(\boldsymbol{A}) \tag{2.3.4}$$

The resulting matrix $\boldsymbol{A}'$ has a positive element in each row, eliminating the need to verify: $max(\boldsymbol{a}_i) > 0\,,\ \forall\, i \in \langle 1, N \rangle$, when confirming that column vectors of $\boldsymbol{A}'$ *positively span* $\mathbb{R}^N$ according to the condition 2.3.3.

2.3.2 Self collision

Self collisions of *CDPMs* are a rarely discussed topic despite them greatly reducing the workspaces of *CRPMs* and *RRPMs*. The issue of their occurrence is most commonly avoided by the platform of the manipulator maintaining constant orientation. In contrast we will create a collision model adaptable to all *CDPMs* exploring the problematic in this thesis.

A practical form in which to represent solid bodies for collision detection are meshes. This is particularly so when considering self collision in parallel manipulators as they consist of closed kinematic chains. Cylindrical elements of mechanisms, for example cables, can be modelled as capsules each characterised by a line segment s and radius ρ as shown in figure 2.1. The problem of determining interference among two capsules is at its core a question of calculating the minimal distance between line segments s_i and s_j at the core of the capsules, further referred to only as the *proximity* of the two line segments δ_{ij}. Consequently we may claim that interference between the two capsules occurs when $\epsilon_{ij} = \delta_{ij} - (\rho_i + \rho_j) \leq 0$. Due to the simplicity of collision detection between capsules it is advantageous to model solid bodies of small thickness's as single layers of capsule mesh.

We will create and describe an algorithm for determining interference between capsules in section 3.5 building upon an approach to detecting collisions of cylindrical elements of a manipulator based on calculating the distance of skew lines running through their centres, described by J.-P. Merlet in [12].

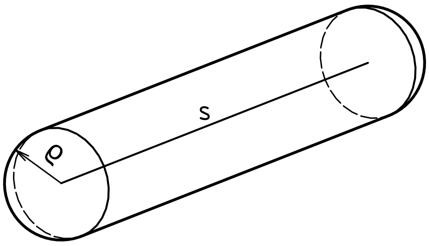


Figure 2.1: Representation of a capsule

2.3.3 Joint limits

As the joints through which the cables are connected to the platform generally satisfy all interference free configurations and the required range of leads through which the cable exits the workspace is reduced to approximitally one octant of cartesian space by force-closure singularities the workspace of a *CDPM* is not commonly further restricted by joint-limits.

Two types of leads are typically used for exit points of the cables: eyelets and pulleys [1]. Eyelets, while causing excessive wear on the cable, have the distinct advantage of providing an almost stationary point where the linearity of the cables ends. Pulleys with a panning axis in the direction of the winch (figure 2.3) at least maintain a predictable end-point of cable linearity while also reducing cable wear. For this reason, even despite their inetria and a more complex mathematical model, pulleys are used in the majority of existing *CDPMs* [1].

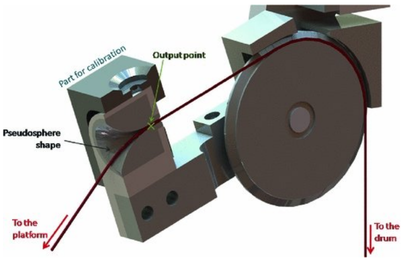


Figure 2.2: Cut CAD view of the eyelet in the pulley assembly [2]

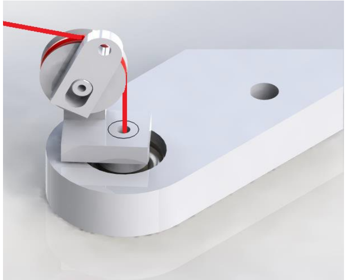


Figure 2.3: Pulley assembly of the cable-robot demonstrator [3]

Chapter 3

Theory

3.1 Proposed manipulator

Beside the platform with variable geometry the proposed manipulator will be similar to other eight cable *RRPMs* such as in figure 1.2. It will be composed of a frame with eight actuated winches, the platform consisting of two bodies with one degree of relative motion facilitated by an actuated prismatic joint and $k = 8$ cables connecting the platform to the frame. The resulting mechanism will therefore have $n = 7$ DOF and $m = 9$ actuators.

There are many viable cable configurations of a spacial cable manipulator [15], [16] with varying degrees of mobility and actuation properties. In [17] Kraus optimizes where cables of an eight cable redundant manipulator attach to the platform based on the *condition number* [18] of the *structure matrix* $\boldsymbol{A}$. We will use a comparable configuration for the purposes of this paper (figure 3.1).

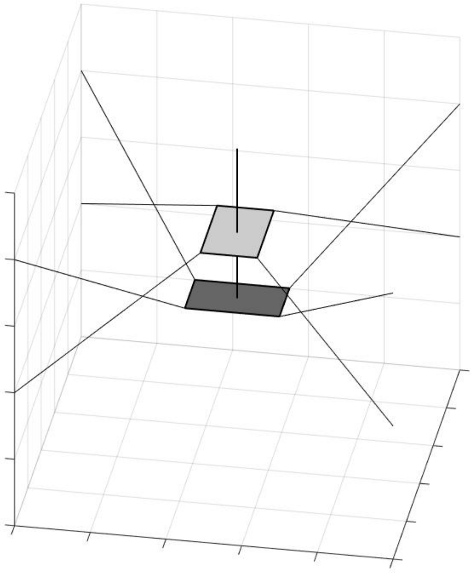


Figure 3.1: Proposed configuration

3.1.1 Mechanical model

We will consider the frame of the manipulator as stationary assigning it the reference coordinate system CS_1 and attribute a local coordinate system to each of the two platform's bodies CS_2 and CS_3 (figure 3.2). Throughout this section we will use upper left indices to denote in which coordinate system elements of a vector are written.

Transformations, expressed using *homogeneous transformation matrices*, between the reference coordinate system CS_1 and the local coordinate systems CS_2 and CS_3 are:

$$\boldsymbol{T}_{12} = \begin{bmatrix} \boldsymbol{E}_3 & {}^1\boldsymbol{r}_{12} \\ \boldsymbol{0} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{R}(x, \phi_{x_{12}}) & \boldsymbol{0} \\ \boldsymbol{0} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{R}(y, \phi_{y_{12}}) & \boldsymbol{0} \\ \boldsymbol{0} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{R}(z, \phi_{z_{12}}) & \boldsymbol{0} \\ \boldsymbol{0} & 1 \end{bmatrix} \tag{3.1.1}$$

$$\boldsymbol{T}_{13} = \boldsymbol{T}_{12}\boldsymbol{T}_{23} = \boldsymbol{T}_{12} \begin{bmatrix} \boldsymbol{E}_3 & {}^2\boldsymbol{r}_{23} \\ \boldsymbol{0} & 1 \end{bmatrix} \tag{3.1.2}$$

where $\boldsymbol{E}_3$ is the identity matrix, $\boldsymbol{R}$ are rotational matrices and $\boldsymbol{r}_{12}$, $\boldsymbol{r}_{23}$ are the radius vectors describing translation between the origins of the two subscribed coordinate systems. The particular form of $\boldsymbol{r}_{23}$ is:

$${}^2\boldsymbol{r}_{23} = s_{23}\,{}^2\boldsymbol{u}_{23} = s_{23} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \tag{3.1.3}$$

s_{23} being the distance between the origins of the two local coordinate systems CS_2 and CS_3.

The cables represented by vectors $\boldsymbol{c}_i$, $i \in \langle 1, 8\rangle$ are spanned between points A_i, B_i:

$$\boldsymbol{c}_i = \boldsymbol{r}_{A_i} - \boldsymbol{r}_{B_i} \tag{3.1.4}$$

where vectors $\boldsymbol{r}_{A_i}$, $\boldsymbol{r}_{B_i}$ describe the position of points A_i, B_i in relation to the origin of CS_1. While values of $\boldsymbol{r}_{A_i}$ are constant by design of the manipulator, vectors $\boldsymbol{r}_{B_i}$ must be extracted from a set of equations:

$$\begin{bmatrix} {}^1\boldsymbol{r}_{B_i} \\ 1 \end{bmatrix} = \boldsymbol{T}_{12} \begin{bmatrix} {}^2\boldsymbol{b}_i \\ 1 \end{bmatrix} , \qquad i \in \langle 1, 8\rangle \tag{3.1.5}$$

as they are dependant on the configuration of the manipulator, where:

$$ {}^{2}\boldsymbol{b}_i = {}^{2}\boldsymbol{r}_i \qquad i \in \langle 1, 4 \rangle \tag{3.1.6}$$

$$ {}^{2}\boldsymbol{b}_j = {}^{3}\boldsymbol{b}_j = {}^{3}\boldsymbol{r}_{23} + {}^{3}\boldsymbol{r}_j \qquad j \in \langle 5, 8 \rangle \tag{3.1.7}$$

are vectors describing the position of anchoring points B_i and B_j in relation to the origin of coordinate system CS_2. Notably, vector ${}^{2}\boldsymbol{r}_{23}$ is variable while ${}^{2}\boldsymbol{r}_i$ and ${}^{3}\boldsymbol{r}_j$ are constant vectors describing the position of anchoring points B_i and B_j in local coordinate systems CS_2 and CS_3 of the two platform's bodies.

Further we may define directional vectors of cables $\boldsymbol{u}_i$:

$$\boldsymbol{u}_i = \frac{\boldsymbol{c}_i}{\|\boldsymbol{c}_i\|}\,, \qquad i \in \langle 1, 8 \rangle \tag{3.1.8}$$

which will be used to derive the *inverse kinematics Jacobian matrix* $\boldsymbol{J}$ and *structure matrix* $\boldsymbol{A}$ of the proposed manipulator in sections 3.2 and 3.3.

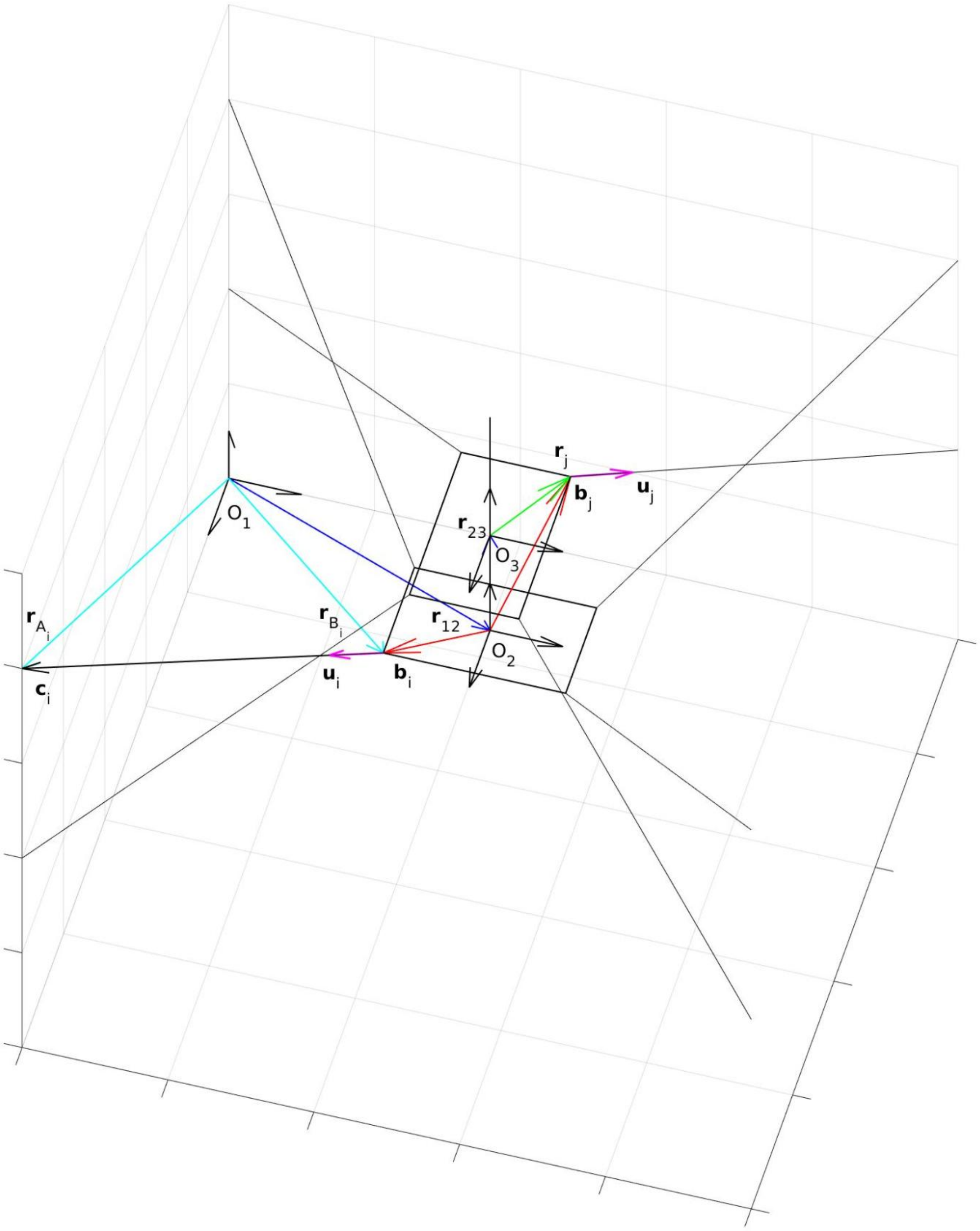


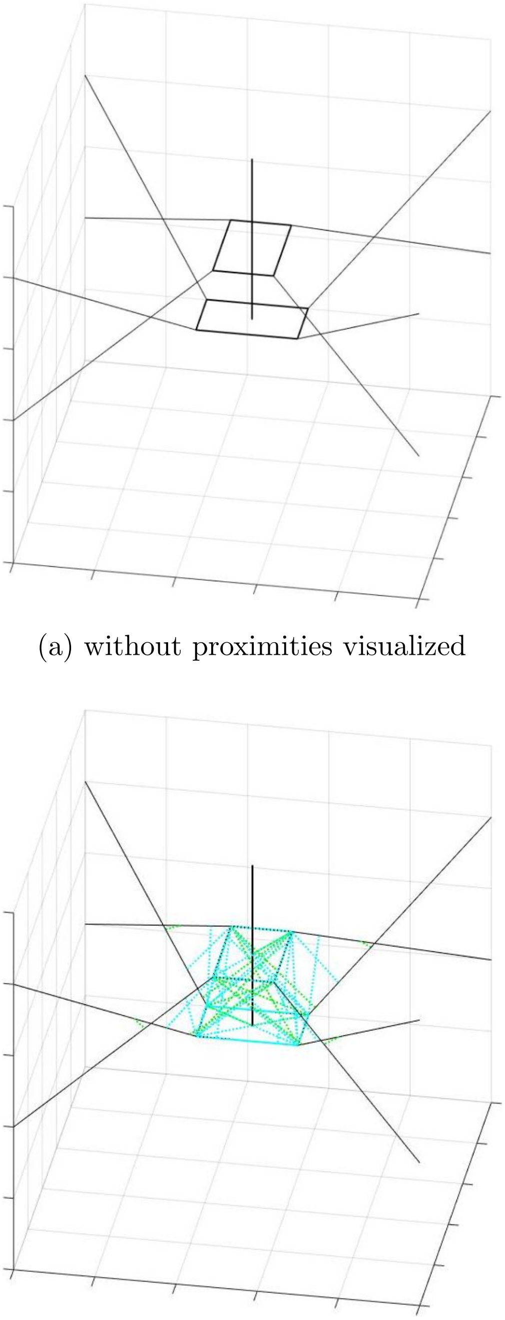
Figure 3.2: Diagram of the mechanical model

3.1.2 Collision model

The collision model of the manipulator will consist entirely of capsules mentioned in section 2.3.2 greatly simplifying the process of interference checking.

As all cables of the manipulator should remain under tension, deviation of their shape from a line should be small enough to be compensated by a radius ρ larger than the radius of the physical cable, without a detrimental effect on the range of motion, thus allowing them to be modelled as single capsules. The frame of the manipulator may take various forms generally not restricting the workspace of the manipulator. For this reason it will not be included in the collision model. Two possible approaches for its incorporation in the model, both not increasing the complexity of the model, would be to represent the frame by additional capsules or monitor that end-points of the line-segments, at the core of capsules forming the platform, remain within certain coordinates. The two bodies of the platform will be modelled as four capsules forming a rectangle each and the column of the prismatic joint as a single capsule moving in conjunction with the lower of the two bodies. This representation of the platform is particularly suitable when cables of the manipulator are modelled as single capsules, continuous with the mesh of the manipulator.

Collision detection between the two bodies of the platform is redundant as they cannot interfere due to the limits of the prismatic joint. Therefore all possible interference must occur either between two cables, cable and a body of the platform or a cable and the column.



(a) without proximities visualized

(b) with proximities visualized

Figure 3.3: Collision model

3.2 Kinematics

Let $\boldsymbol{q} \in \mathbb{R}^m$ be a vector of actuated joint parameters and $\boldsymbol{X} \in \mathbb{R}^n$ a vector of unique parameters describing any particular configuration of a mechanism. In order to derive the *inverse kinematics Jacobian* $\boldsymbol{J}$ we must derive the maps of $\boldsymbol{X}$ and it's derivates onto $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$:

- *actuator position*: $\boldsymbol{X} \rightarrow \boldsymbol{q}$
- *actuator velocity*: $\dot{\boldsymbol{X}} \rightarrow \dot{\boldsymbol{q}}$

3.2.1 Actuator position

From equations 3.1.1 and 3.1.2 describing homogenous tranformations between the coordinate systems of the manipulator we may extract parameters $X_i\,,\ i \in \langle 1, 7 \rangle$ describing any particular configuration of the manipulator:

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{r}_{12} \\ \boldsymbol{\phi}_{12} \\ s_{23} \end{bmatrix}, \qquad \boldsymbol{X} \in \mathbb{R}^7 \tag{3.2.1}$$

where:

$$\boldsymbol{\phi}_{12} = \begin{bmatrix} \phi_{x_{12}} & \phi_{y_{12}} & \phi_{z_{12}} \end{bmatrix}^T \tag{3.2.2}$$

Through the actuation of winches the lenght of individual cables is changed, therefore we will define $q_i\,,\ i \in \langle 1, 8 \rangle$ as:

$$q_i = \|\boldsymbol{c}_i\|, \qquad i \in \langle 1, 8 \rangle \tag{3.2.3}$$

and q_9 as the distance between the origins of the two local coordinate systems of the platform's bodies:

$$q_9 = s_{23} \tag{3.2.4}$$

Resulting in nine equations describing $q_i\,,\ i \in \langle 1, 9 \rangle$ as functions of $\boldsymbol{X}$.

3.2.2 Actuator velocity

Instead of differentiating the equations of actuator positions we will derive the map of $\dot{\boldsymbol{X}}$:

$$\dot{\boldsymbol{X}} = \begin{bmatrix} \boldsymbol{v}_{12} \\ \boldsymbol{\omega}_{12} \\ \dot{s}_{23} \end{bmatrix}, \qquad \dot{\boldsymbol{X}} \in \mathbb{R}^7 \tag{3.2.5}$$

onto $\dot{\boldsymbol{q}} \in \mathbb{R}^9$ from the geometry of the manipulator (figure 3.4):

$$\dot{q}_i = (\boldsymbol{v}_{12} + \boldsymbol{\omega}_{12} \times \boldsymbol{b}_i) \cdot \boldsymbol{u}_i \qquad i \in \langle 1, 4 \rangle \tag{3.2.6}$$

$$\dot{q}_j = (\boldsymbol{v}_{12} + \boldsymbol{\omega}_{12} \times \boldsymbol{b}_j + \dot{s}_{23}\, \boldsymbol{u}_{23}) \cdot \boldsymbol{u}_j \qquad j \in \langle 5, 8 \rangle \tag{3.2.7}$$

$$\dot{q}_9 = \dot{s}_{23} \tag{3.2.8}$$

where equations 3.2.6 and 3.2.7 can be manipulated into forms:

$$\dot{q}_i = \boldsymbol{u}_i^T \boldsymbol{v}_{12} + (\boldsymbol{b}_i \times \boldsymbol{u}_i)^T \boldsymbol{\omega}_{12} \qquad i \in \langle 1, 4 \rangle \tag{3.2.9}$$

$$\dot{q}_j = \boldsymbol{u}_j^T \boldsymbol{v}_{12} + (\boldsymbol{b}_j \times \boldsymbol{u}_j)^T \boldsymbol{\omega}_{12} + (\boldsymbol{u}_j \cdot \boldsymbol{u}_{23})\, \dot{s}_{23} \qquad j \in \langle 5, 8 \rangle \tag{3.2.10}$$

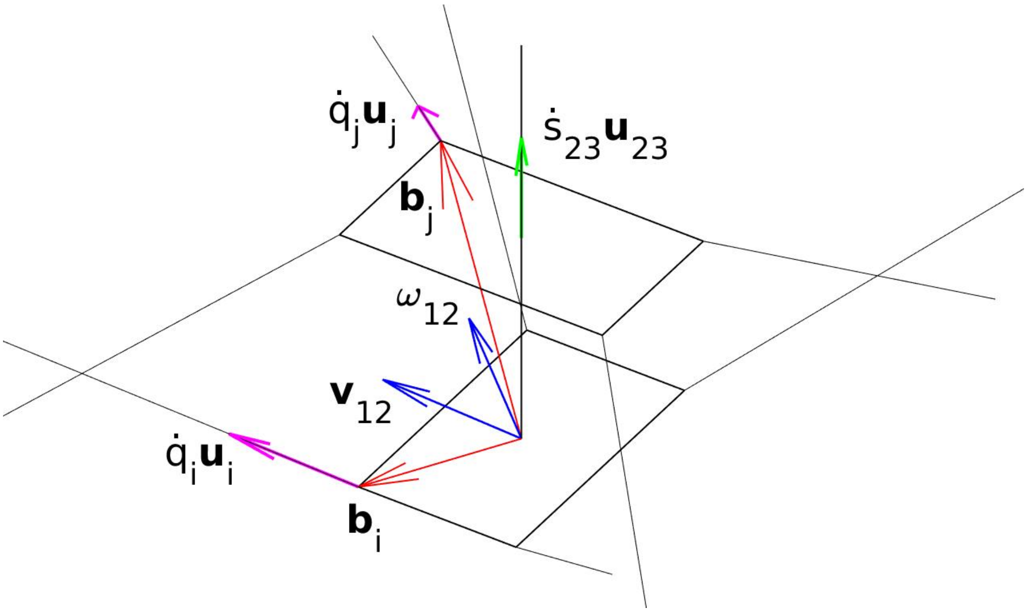


Figure 3.4: Velocity relations

From equations 3.2.8, 3.2.9 and 3.2.10 we may construct the map of $\dot{\boldsymbol{X}}$ onto $\dot{\boldsymbol{q}}$ in the form:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{X})\,\dot{\boldsymbol{X}} \qquad \boldsymbol{J} \in \mathbb{R}^{9\times 7} \tag{3.2.11}$$

where $\boldsymbol{J}(\boldsymbol{X})$ is the *inverse kinematics Jacobian*:

$$\boldsymbol{J}(\boldsymbol{X}) = \begin{bmatrix} \boldsymbol{u}_1^T & (\boldsymbol{b}_1 \times \boldsymbol{u}_1)^T & 0 \\ \boldsymbol{u}_2^T & (\boldsymbol{b}_2 \times \boldsymbol{u}_2)^T & 0 \\ \boldsymbol{u}_3^T & (\boldsymbol{b}_3 \times \boldsymbol{u}_3)^T & 0 \\ \boldsymbol{u}_4^T & (\boldsymbol{b}_4 \times \boldsymbol{u}_4)^T & 0 \\ \boldsymbol{u}_5^T & (\boldsymbol{b}_5 \times \boldsymbol{u}_5)^T & \boldsymbol{u}_{23} \cdot \boldsymbol{u}_5 \\ \boldsymbol{u}_6^T & (\boldsymbol{b}_6 \times \boldsymbol{u}_6)^T & \boldsymbol{u}_{23} \cdot \boldsymbol{u}_6 \\ \boldsymbol{u}_7^T & (\boldsymbol{b}_7 \times \boldsymbol{u}_7)^T & \boldsymbol{u}_{23} \cdot \boldsymbol{u}_7 \\ \boldsymbol{u}_8^T & (\boldsymbol{b}_8 \times \boldsymbol{u}_8)^T & \boldsymbol{u}_{23} \cdot \boldsymbol{u}_8 \\ 0 \;\; 0 \;\; 0 & 0 \;\; 0 \;\; 0 & 1 \end{bmatrix} \tag{3.2.12}$$

3.3 Static Equilibrium

Let $\boldsymbol{w}_{l\,O_2} \in \mathbb{R}^6$ be the wrench of external forces and torques acting upon a body l in refrence to point O_2:

$$\boldsymbol{w}_{l\,O_2} = \begin{bmatrix} \sum_j \boldsymbol{F}_j^E \\ \sum_j \boldsymbol{M}_j^E \end{bmatrix} \tag{3.3.1}$$

and $\boldsymbol{\zeta} \in \mathbb{R}^k$ a vector containing scalar values of forces produced by actuators of the manipulator.

We may define forces $\boldsymbol{F}_i\,,\ i \in \langle 1, 8 \rangle$ exerted by the cables onto the platform as:

$$\boldsymbol{F}_i = \zeta_i \boldsymbol{u}_i\ , \quad i \in \langle 1, 8 \rangle \tag{3.3.2}$$

assuming they are co-linear with vectors $\boldsymbol{c}_i$. Further we may define forces and torques acting between the two bodies of the platform as $\boldsymbol{F}_{23} = -\boldsymbol{F}_{32}$ and $\boldsymbol{M}_{23} = -\boldsymbol{M}_{32}$, where $\zeta_9\,\boldsymbol{u}_{23} = F_{z_{23}} \cdot \boldsymbol{u}_{23} = -F_{z_{32}} \cdot \boldsymbol{u}_{23}$.

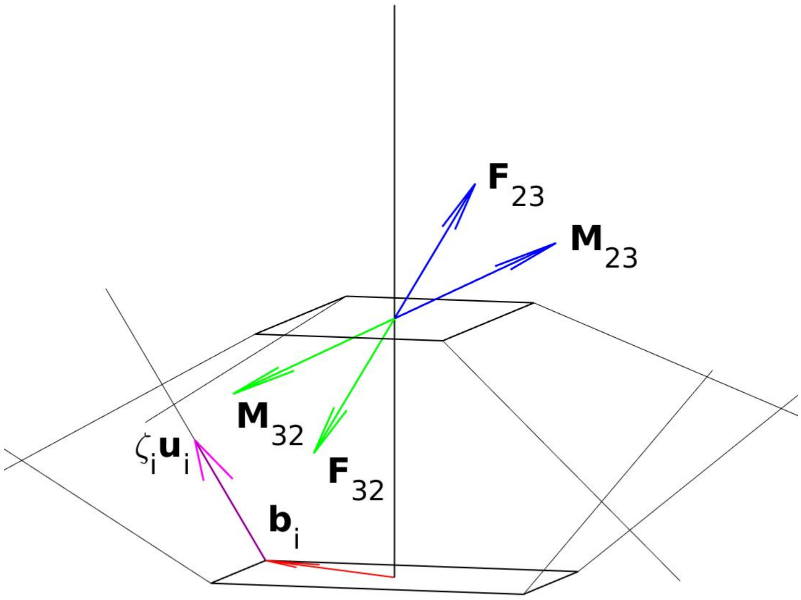


Figure 3.5: Diagram of forces acting upon the platform

3.3.1 Static equilibrium of the platform

The equations of static equilibrium for the two seperate bodies are:

$$\begin{bmatrix} \boldsymbol{F}_1 & \dots & \boldsymbol{F}_4 \\ \boldsymbol{M}_1 & \dots & \boldsymbol{M}_4 \end{bmatrix} + \begin{bmatrix} \boldsymbol{F}_{32} \\ \boldsymbol{M}_{32} \end{bmatrix} + \boldsymbol{w}_{3\,O_2} = \boldsymbol{0} \tag{3.3.3}$$

$$\begin{bmatrix} \boldsymbol{F}_5 & \dots & \boldsymbol{F}_8 \\ \boldsymbol{M}_5 & \dots & \boldsymbol{M}_8 \end{bmatrix} + \begin{bmatrix} \boldsymbol{F}_{23} \\ \boldsymbol{M}_{23} \end{bmatrix} + \boldsymbol{w}_{2\,O_2} = \boldsymbol{0} \tag{3.3.4}$$

By adding 3.3.3 and 3.3.4 we obtain the equations of static equilibrium for the platform viewed as one body:

$$\begin{bmatrix} \boldsymbol{F}_1 & \dots & \boldsymbol{F}_8 \\ \boldsymbol{M}_1 & \dots & \boldsymbol{M}_8 \end{bmatrix} + \boldsymbol{w} = \boldsymbol{0} \tag{3.3.5}$$

where $\boldsymbol{w}$ is the sum of wrenches applied to the two platfrom's bodies in reference to point O_2:

$$\boldsymbol{w} = \boldsymbol{w}_{2\,O_2} + \boldsymbol{w}_{3\,O_2} \tag{3.3.6}$$

Equation 3.3.5 can be manipulated into:

$$-\begin{bmatrix} \boldsymbol{u}_1 & \dots & \boldsymbol{u}_8 \\ \boldsymbol{b}_1 \times \boldsymbol{u}_1 & \dots & \boldsymbol{b}_8 \times \boldsymbol{u}_8 \end{bmatrix} \begin{bmatrix} \zeta_1 \\ \vdots \\ \zeta_8 \end{bmatrix} = \boldsymbol{w} \tag{3.3.7}$$

Equation 3.3.7 expresses the wrench $\boldsymbol{w}$ as a function of $\boldsymbol{X}$ and $\zeta_i\,,\ i \in \langle 1, 8 \rangle$. It can be rewritten into the common form described in 2.2:

$$\boldsymbol{w} = \boldsymbol{A}(\boldsymbol{X})\,\boldsymbol{\tau} \tag{3.3.8}$$

where $\boldsymbol{\tau}$ is the cable tension vector:

$$\boldsymbol{\tau} = \begin{bmatrix} \zeta_1 & \dots & \zeta_8 \end{bmatrix}^T \tag{3.3.9}$$

and $\boldsymbol{A}$ the *structure matrix* of the platform:

$$\boldsymbol{A}(\boldsymbol{X}) = -\begin{bmatrix} \boldsymbol{u}_1 & \dots & \boldsymbol{u}_8 \\ \boldsymbol{b}_1 \times \boldsymbol{u}_1 & \dots & \boldsymbol{b}_8 \times \boldsymbol{u}_8 \end{bmatrix} \tag{3.3.10}$$

crucial for solving the problem of tension distriubition among cables and *force-closure singularities* 2.3.1.

3.3.2 Actuation force in the prismatic joint

For the two bodies of the platform to remain in equilibrium ζ_9 must compensate the effect of cable forces F_i , $i \in \langle 1, 4\rangle$ and F_i , $i \in \langle 5, 8\rangle$ onto the respective bodies of the platform in the direction of the vector $\boldsymbol{u}_{23}$:

$$\zeta_9 = -\sum_{i=1}^{4} \boldsymbol{F}_i \cdot \boldsymbol{u}_{23} = \sum_{i=5}^{8} \boldsymbol{F}_i \cdot \boldsymbol{u}_{23} \tag{3.3.11}$$

We may therefore express ζ_9 as a function of $\boldsymbol{X}$ and ζ_i , $i \in \langle 5, 8\rangle$ in vector form as:

$$\zeta_9 = -\begin{bmatrix} \boldsymbol{u}_1 \cdot \boldsymbol{u}_{23} & \dots & \boldsymbol{u}_4 \cdot \boldsymbol{u}_{23} \end{bmatrix} \begin{bmatrix} \zeta_1 \\ \vdots \\ \zeta_r \end{bmatrix} \tag{3.3.12}$$

3.4 Collision detection

Self collision of the manipulator may occur when certain capsules of the collision model interfere. In agreement with section 3.1.2 we will check for interference between the capsules of the collision model in three phases.

- cable to cable
- cable to platform body
- cable to column

First step of each phase is calculating the proximity between line-segments at the core of each cable's capsule (rows) and all capsules of the platform (columns), producing the matrix $\boldsymbol{D} \in \mathbb{R}^{8\times 17}$:

$$\boldsymbol{D} = \begin{bmatrix} \boldsymbol{D}_I & \boldsymbol{D}_{II} & \boldsymbol{D}_{III} \end{bmatrix} \tag{3.4.1}$$

Matrix $\boldsymbol{D}_I$ is the result of the first phase of proximity calculations. As proximity $\delta_{ii} = 0$ and $\delta_{ij} \equiv \delta_{ji}$ for $i \in \langle 1, 8 \rangle$, it is necessary to calculate only elements above the matrix's diagonal:

$$\boldsymbol{D}_I = \begin{bmatrix} 0 & \delta_{12} & \delta_{13} & \delta_{14} & \delta_{15} & \delta_{16} & \delta_{17} & \delta_{18} \\ 0 & 0 & \delta_{23} & \delta_{24} & \delta_{25} & \delta_{26} & \delta_{27} & \delta_{28} \\ 0 & 0 & 0 & \delta_{34} & \delta_{35} & \delta_{36} & \delta_{37} & \delta_{38} \\ 0 & 0 & 0 & 0 & \delta_{45} & \delta_{46} & \delta_{47} & \delta_{48} \\ 0 & 0 & 0 & 0 & 0 & \delta_{56} & \delta_{57} & \delta_{58} \\ 0 & 0 & 0 & 0 & 0 & 0 & \delta_{67} & \delta_{68} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \delta_{78} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.4.2}$$

Each line segment of a cable capsule shares an end-point with two line-segments of the manipulator's capsules. Therefore $\boldsymbol{D}_{II}$ will take the form:

$$\boldsymbol{D}_{II} = \begin{bmatrix} 0 & \delta_{1\,10} & \delta_{1\,11} & 0 & \delta_{1\,13} & \delta_{1\,14} & \delta_{1\,15} & \delta_{1\,16} \\ 0 & 0 & \delta_{2\,11} & \delta_{2\,12} & \delta_{2\,13} & \delta_{2\,14} & \delta_{2\,15} & \delta_{2\,16} \\ \delta_{3\,9} & 0 & 0 & \delta_{3\,12} & \delta_{3\,13} & \delta_{3\,14} & \delta_{3\,15} & \delta_{3\,16} \\ \delta_{4\,9} & \delta_{4\,10} & 0 & 0 & \delta_{4\,13} & \delta_{4\,14} & \delta_{4\,15} & \delta_{4\,16} \\ \delta_{5\,9} & \delta_{5\,10} & \delta_{5\,11} & \delta_{5\,12} & 0 & \delta_{5\,14} & \delta_{5\,15} & 0 \\ \delta_{6\,9} & \delta_{6\,10} & \delta_{6\,11} & \delta_{6\,12} & 0 & 0 & \delta_{6\,15} & \delta_{6\,16} \\ \delta_{7\,9} & \delta_{7\,10} & \delta_{7\,11} & \delta_{7\,12} & \delta_{7\,13} & 0 & 0 & \delta_{7\,16} \\ \delta_{8\,9} & \delta_{8\,10} & \delta_{8\,11} & \delta_{8\,12} & \delta_{8\,13} & \delta_{8\,14} & 0 & 0 \end{bmatrix} \tag{3.4.3}$$

The product of the last phase $\boldsymbol{D}_{III}$ expresses the proximity between the line-segments representing each cable to the line-segment of the column:

$$\boldsymbol{D}_{III} = \begin{bmatrix} \delta_{1\,17} & \delta_{2\,17} & \delta_{3\,17} & \delta_{4\,17} & \delta_{5\,17} & \delta_{6\,17} & \delta_{7\,17} & \delta_{8\,17} \end{bmatrix}^T \tag{3.4.4}$$

The resulting matrix $\boldsymbol{D}$ consists of values $\delta_{i\,j} \geq 0$ and zero values that do not factor in the problem of interference as it is safe to assume that a cable will not collide with a platform's segment, immediately proximal to its origin, while not colliding with the remaining segments of the platform.

Second step, after calculating the matrix $\boldsymbol{D}$, is to determine the proximity of capsules surrounding the segments. With such intent we will define $\boldsymbol{E} = \begin{bmatrix} \boldsymbol{E}_I & \boldsymbol{E}_{II} & \boldsymbol{E}_{III} \end{bmatrix}$ as:

$$\boldsymbol{E} = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\epsilon_{1\,2} & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\epsilon_{1\,3} & \epsilon_{2\,3} & 1 & 1 & 1 & 1 & 1 & 1 \\
\epsilon_{1\,4} & \epsilon_{2\,4} & \epsilon_{3\,4} & 1 & 1 & 1 & 1 & 1 \\
\epsilon_{1\,5} & \epsilon_{2\,5} & \epsilon_{3\,5} & \epsilon_{4\,5} & 1 & 1 & 1 & 1 \\
\epsilon_{1\,6} & \epsilon_{2\,6} & \epsilon_{3\,6} & \epsilon_{4\,6} & \epsilon_{5\,6} & 1 & 1 & 1 \\
\epsilon_{1\,7} & \epsilon_{2\,7} & \epsilon_{3\,7} & \epsilon_{4\,7} & \epsilon_{5\,7} & \epsilon_{6\,7} & 1 & 1 \\
\epsilon_{1\,8} & \epsilon_{2\,8} & \epsilon_{3\,8} & \epsilon_{4\,8} & \epsilon_{5\,8} & \epsilon_{6\,8} & \epsilon_{7\,8} & 1 \\
1 & 1 & \epsilon_{3\,9} & \epsilon_{4\,9} & \epsilon_{5\,9} & \epsilon_{6\,9} & \epsilon_{7\,9} & \epsilon_{8\,9} \\
\epsilon_{1\,10} & 1 & 1 & \epsilon_{4\,10} & \epsilon_{5\,10} & \epsilon_{6\,10} & \epsilon_{7\,10} & \epsilon_{8\,10} \\
\epsilon_{1\,11} & \epsilon_{2\,11} & 1 & 1 & \epsilon_{5\,11} & \epsilon_{6\,11} & \epsilon_{7\,11} & \epsilon_{8\,11} \\
1 & \epsilon_{2\,12} & \epsilon_{3\,12} & 1 & \epsilon_{5\,12} & \epsilon_{6\,12} & \epsilon_{7\,12} & \epsilon_{8\,12} \\
\epsilon_{1\,13} & \epsilon_{2\,13} & \epsilon_{3\,13} & \epsilon_{4\,13} & 1 & 1 & \epsilon_{7\,13} & \epsilon_{8\,13} \\
\epsilon_{1\,14} & \epsilon_{2\,14} & \epsilon_{3\,14} & \epsilon_{4\,14} & \epsilon_{5\,14} & 1 & 1 & \epsilon_{8\,14} \\
\epsilon_{1\,15} & \epsilon_{2\,15} & \epsilon_{3\,15} & \epsilon_{4\,15} & \epsilon_{5\,15} & \epsilon_{6\,15} & 1 & 1 \\
\epsilon_{1\,16} & \epsilon_{2\,16} & \epsilon_{3\,16} & \epsilon_{4\,16} & 1 & \epsilon_{6\,16} & \epsilon_{7\,16} & 1 \\
\epsilon_{1\,17} & \epsilon_{2\,17} & \epsilon_{3\,17} & \epsilon_{4\,17} & \epsilon_{5\,17} & \epsilon_{6\,17} & \epsilon_{7\,17} & \epsilon_{8\,17}
\end{bmatrix}^T \tag{3.4.5}$$

Where $\epsilon_{ij} = \delta_{ij} - (\rho_i + \rho_j)$.

We may consider the pose of the manipulator interference free if all elements of $\boldsymbol{E}$ are positive.

3.5 Line segment proximity algorithm

In this section we will describe an algorithm used to calculate the proximity δ_{ij} of two line-segments $s_i = \overline{\boldsymbol{A}_i\boldsymbol{B}_i}$, $s_j = \overline{\boldsymbol{A}_j\boldsymbol{B}_j}$, defined as the minimal distance between two line-segments s_i, s_j in section 3.1.2. For the purposes of proximity calculation the relative position and length of the line-segments will be expressed by vectors $\boldsymbol{a}_{ij} = \boldsymbol{A}_i - \boldsymbol{A}_j$, $\boldsymbol{v}_i = \boldsymbol{B}_i - \boldsymbol{A}_i$, $\boldsymbol{v}_j = \boldsymbol{B}_j - \boldsymbol{A}_j$. We will also express where on the line-segments proximity occurs using scalars ξ_i, ξ_j, where $[\,\xi_i\,\xi_j\,] \in \langle 0, 1\rangle^2$, and determine vectors $\boldsymbol{n}_{ij}$, $\boldsymbol{d}_{ij} = \delta_{ij}\boldsymbol{n}_{ij}$:

$$\xi_i\boldsymbol{v}_i + \boldsymbol{d}_{ij} = \boldsymbol{a}_{ij} + \xi_j\boldsymbol{v}_j \tag{3.5.1}$$

these are mostly by-products of calculations, necessary for determining δ_{ij}, while also allowing us to visualize the results.

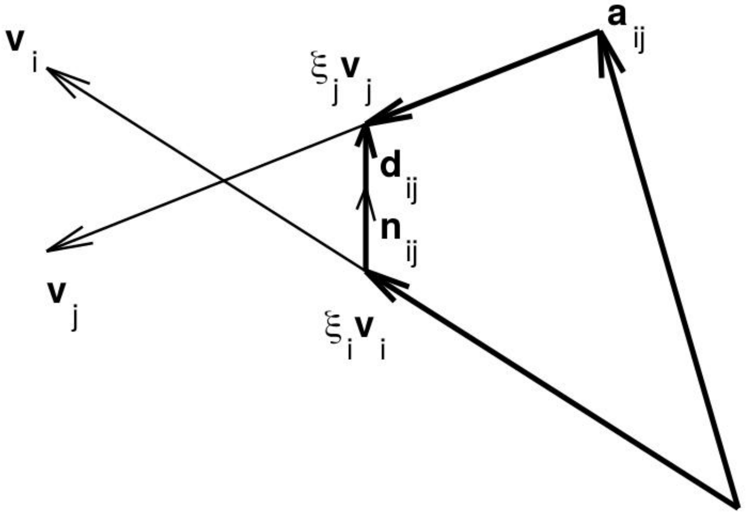


Figure 3.6: diagram of the relation between $\boldsymbol{v}_i$, $\boldsymbol{v}_j$, ξ_i, ξ_j, $\boldsymbol{n}_{ij}$ and $\boldsymbol{d}_{ij}$

Line-segments in space can be either parallel or skew in relation to each other. Along this distinction we will separate the algorithm into two parts.

- parallel proximity
- skew proximity

3.5.1 Parallel proximity

Parallel proximity between two line-segments occurs when vector $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ are co-linear. The proposed algorithm recognises numerous configurations of parallel line-segments (figure 3.7) based on the values of $\boldsymbol{v}_i \cdot \boldsymbol{a}_{ij}$, $\boldsymbol{v}_i \cdot \boldsymbol{v}_j$, $\boldsymbol{v}_i \cdot \boldsymbol{v}_i$ (table 3.1) and determines vectors $\boldsymbol{n}_{ij}$, $\boldsymbol{d}_{ij}$ and scalars δ_{ij}, ξ_i, ξ_j appropriately (table 3.2).

In the case of parallel line-segments δ_{ij} can be measured either between two of their end-points or an infinite pairs of points on the line-segments (figures: 3.7b, 3.7e, 3.7i). In the later case, the proposed algorithm determines a single pair of scalars ξ_i and ξ_j, so that $[\, \xi_i \, \xi_j \,] \in \langle 0, 1 \rangle^2$.

A more comprehensive algorithm, determining all values of ξ_i and ξ_j, where proximity of s_i and s_j occurs, could be derived as an extension of the proposed algorithm. Also a more efficient algorithm, calculating δ_{ij} while not producing ξ_i and ξ_j, could be developed for real-time applications, sacrificing the option of visualization.

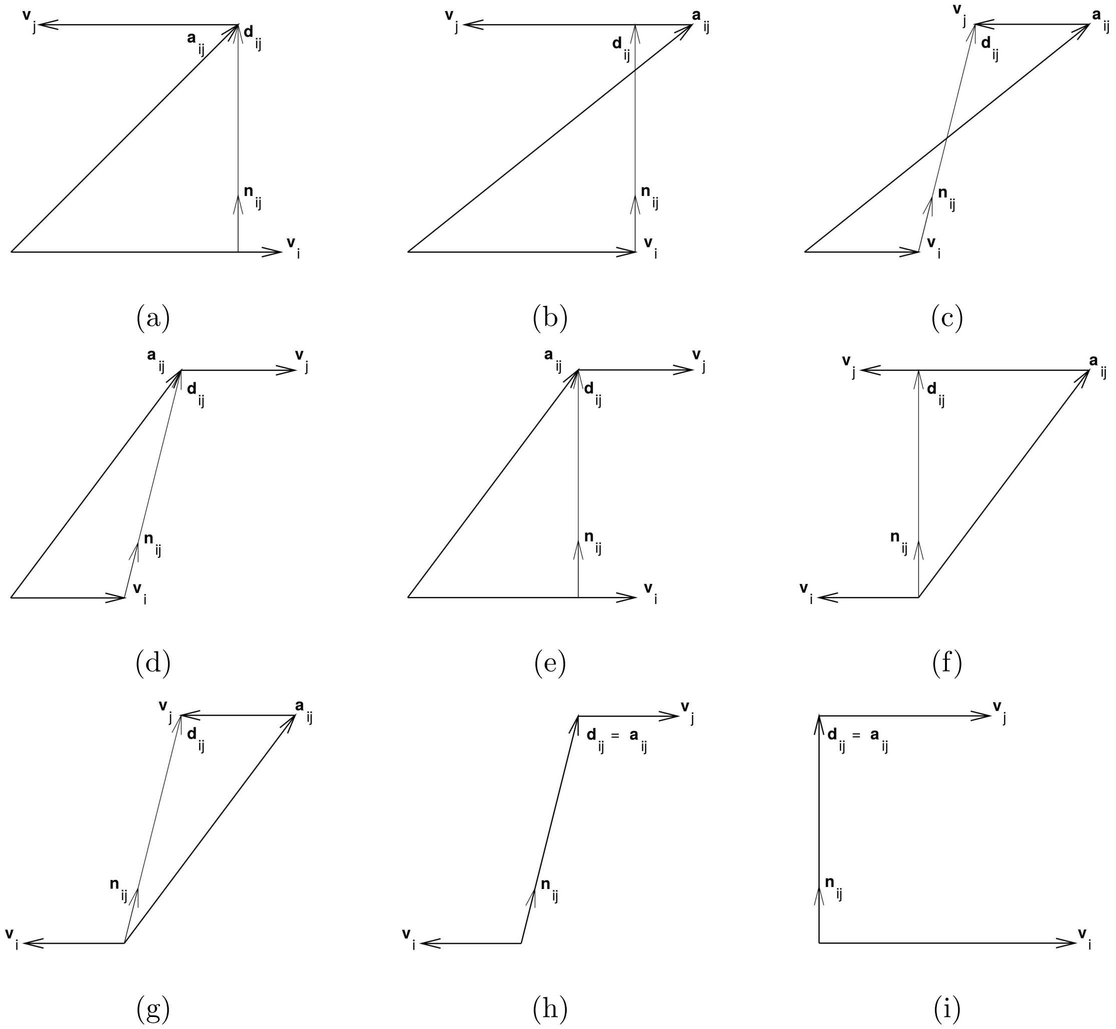


Figure 3.7: Possible configurations of parallel line-segments

Config.	Conditions			
a	$\boldsymbol{v}_i \cdot \boldsymbol{a}_{ij} > 0$	$\boldsymbol{v}_i \cdot \boldsymbol{v}_j < 0$	$\boldsymbol{v}_i \cdot \boldsymbol{v}_i - \boldsymbol{v}_i \cdot \boldsymbol{v}_j > \boldsymbol{v}_i \cdot \boldsymbol{a}_{ij}$	$\boldsymbol{v}_i \cdot \boldsymbol{v}_i > \boldsymbol{v}_i \cdot \boldsymbol{a}_{ij}$
b				$\boldsymbol{v}_i \cdot \boldsymbol{v}_i < \boldsymbol{v}_i \cdot \boldsymbol{a}_{ij}$
c			$\boldsymbol{v}_i \cdot \boldsymbol{v}_i - \boldsymbol{v}_i \cdot \boldsymbol{v}_j \leq \boldsymbol{v}_i \cdot \boldsymbol{a}_{ij}$	
d		$\boldsymbol{v}_i \cdot \boldsymbol{v}_j > 0$	$\boldsymbol{v}_i \cdot \boldsymbol{v}_i \leq \boldsymbol{v}_i \cdot \boldsymbol{a}_{ij}$	
e			$\boldsymbol{v}_i \cdot \boldsymbol{v}_i > \boldsymbol{v}_i \cdot \boldsymbol{a}_{ij}$	
f	$\boldsymbol{v}_i \cdot \boldsymbol{a}_{ij} < 0$	$\boldsymbol{v}_i \cdot \boldsymbol{v}_j > 0$	$\boldsymbol{v}_i \cdot \boldsymbol{v}_j \leq -\boldsymbol{v}_i \cdot \boldsymbol{a}_{ij}$	
g			$\boldsymbol{v}_i \cdot \boldsymbol{v}_j > -\boldsymbol{v}_i \cdot \boldsymbol{a}_{ij}$	
h		$\boldsymbol{v}_i \cdot \boldsymbol{v}_j < 0$		
i	$\boldsymbol{v}_i \cdot \boldsymbol{a}_{ij} = 0$			

Table 3.1: Parallel line-segment configuration conditions

Config.	$\boldsymbol{n}_{ij}$	δ_{ij}	$\boldsymbol{d}_{ij}$	ξ_i	ξ_j
a	$\frac{(\boldsymbol{v}_i \times \boldsymbol{a}_{ij}) \times \boldsymbol{v}_i}{\\|(\boldsymbol{v}_i \times \boldsymbol{a}_{ij}) \times \boldsymbol{v}_i\\|}$	$\boldsymbol{a}_{ij} \cdot \boldsymbol{n}_{ij}$	$\delta_{ij} \boldsymbol{n}_{ij}$	$\frac{\boldsymbol{v}_i \cdot \boldsymbol{a}_{ij}}{\boldsymbol{v}_i \cdot \boldsymbol{v}_i}$	0
b	$\frac{(\boldsymbol{v}_i \times \boldsymbol{a}_{ij}) \times \boldsymbol{v}_i}{\\|(\boldsymbol{v}_i \times \boldsymbol{a}_{ij}) \times \boldsymbol{v}_i\\|}$	$\boldsymbol{a}_{ij} \cdot \boldsymbol{n}_{ij}$	$\delta_{ij} \boldsymbol{n}_{ij}$	1	$\frac{\boldsymbol{v}_i \cdot \boldsymbol{a}_{ij} - \boldsymbol{v}_i \cdot \boldsymbol{v}_i}{-\boldsymbol{v}_i \cdot \boldsymbol{v}_j}$
c	$\frac{1}{\delta_{ij}} \boldsymbol{d}_{ij}$	$\\|\boldsymbol{d}_{ij}\\|$	$\boldsymbol{B}_j - \boldsymbol{B}_i$	1	1
d	$\frac{1}{\delta_{ij}} \boldsymbol{d}_{ij}$	$\\|\boldsymbol{d}_{ij}\\|$	$\boldsymbol{A}_j - \boldsymbol{B}_i$	1	0
e	$\frac{(\boldsymbol{v}_i \times \boldsymbol{a}_{ij}) \times \boldsymbol{v}_i}{\\|(\boldsymbol{v}_i \times \boldsymbol{a}_{ij}) \times \boldsymbol{v}_i\\|}$	$\boldsymbol{a}_{ij} \cdot \boldsymbol{n}_{ij}$	$\delta_{ij} \boldsymbol{n}_{ij}$	$\frac{\boldsymbol{v}_i \cdot \boldsymbol{a}_{ij}}{\boldsymbol{v}_i \cdot \boldsymbol{v}_i}$	0
f	$\frac{1}{\delta_{ij}} \boldsymbol{d}_{ij}$	$\\|\boldsymbol{d}_{ij}\\|$	$\boldsymbol{B}_j - \boldsymbol{A}_i$	0	1
g	$\frac{(\boldsymbol{v}_i \times \boldsymbol{a}_{ij}) \times \boldsymbol{v}_i}{\\|(\boldsymbol{v}_i \times \boldsymbol{a}_{ij}) \times \boldsymbol{v}_i\\|}$	$\boldsymbol{a}_{ij} \cdot \boldsymbol{n}_{ij}$	$\delta_{ij} \boldsymbol{n}_{ij}$	0	$\frac{\boldsymbol{v}_i \cdot \boldsymbol{a}_{ij}}{-\boldsymbol{v}_i \cdot \boldsymbol{v}_j}$
h	$\frac{1}{\delta_{ij}} \boldsymbol{d}_{ij}$	$\\|\boldsymbol{d}_{ij}\\|$	$\boldsymbol{A}_j - \boldsymbol{A}_i$	0	0
i	$\frac{1}{\delta_{ij}} \boldsymbol{d}_{ij}$	$\\|\boldsymbol{d}_{ij}\\|$	$\boldsymbol{A}_j - \boldsymbol{A}_i$	0	0

Table 3.2: $\boldsymbol{n}_{ij}$, $\boldsymbol{d}_{ij}$, δ_{ij}, ξ_i, ξ_j depending on configuration

3.5.2 Skew proximity

We will distinguish three types of proximity between skew line-segments s_i and s_j:

- Unbound proximity
- Partially bound proximity
- Fully bound proximity

depending on the extent to which the proximity is limited, by their bounds (end-points: $\boldsymbol{A}_i$, $\boldsymbol{B}_i$, $\boldsymbol{A}_j$, $\boldsymbol{B}_j$).

Unbound proximity

An approach to calculating δ'_{ij}, ξ'_i, ξ'_j of two skew lines l_i and l_j is described by J.-P. Merlet in [12]. Let $s_i \in l_i$, $s_j \in l_j$ and $\boldsymbol{n}'_{ij}$ be the common perpendicular of skew lines l_i and l_j:

$$\boldsymbol{n}'_{ij} = \frac{\boldsymbol{v}_i \times \boldsymbol{v}_j}{\|\boldsymbol{v}_i \times \boldsymbol{v}_j\|} \tag{3.5.2}$$

$$\delta'_{ij} = \boldsymbol{n}'_{ij} \cdot \boldsymbol{a}_{ij} \tag{3.5.3}$$

$$\boldsymbol{d}'_{ij} = \delta'_{ij}\boldsymbol{n}'_{ij} \tag{3.5.4}$$

The relation between ξ'_i, $\boldsymbol{v}_i$, $\boldsymbol{v}_j$, $\boldsymbol{n}'_{ij}$, $\boldsymbol{a}_{ij}$ is determined in [12] as:

$$\frac{\xi'_i\|\boldsymbol{v}_i\|}{\boldsymbol{a}_{ij} \cdot (\boldsymbol{v}_j \times \boldsymbol{n}'_{ij})} = \frac{\|\boldsymbol{v}_i\|}{\|\boldsymbol{v}_i \times \boldsymbol{v}_j\|} \tag{3.5.5}$$

Equation 3.5.5 is then manipulated into:

$$\xi'_i = \frac{\boldsymbol{a}_{ij} \cdot (\boldsymbol{v}_j \times \boldsymbol{n}'_{ij})}{\|\boldsymbol{v}_i \times \boldsymbol{v}_j\|} \tag{3.5.6}$$

and similarly ξ'_j can be derived as:

$$\xi'_j = \frac{\boldsymbol{a}_{ij} \cdot (\boldsymbol{v}_i \times \boldsymbol{n}'_{ij})}{\|\boldsymbol{v}_i \times \boldsymbol{v}_j\|} \tag{3.5.7}$$

If $[\,\xi_i' \; \xi_j'\,] \in \langle 0,1\rangle^2$, proximity of the two lines occurs in the bounds of the line-segments $s_i = \overline{\boldsymbol{A}_i\boldsymbol{B}_i}$, $s_j = \overline{\boldsymbol{A}_j\boldsymbol{B}_j}$ and the proximity of the line-segments s_i, s_j (as shown in figure 3.6) is considered *unbound* and characterised by:

$$\boldsymbol{n}_{ij} = \frac{\boldsymbol{v}_i \times \boldsymbol{v}_j}{\|\boldsymbol{v}_i \times \boldsymbol{v}_j\|} \tag{3.5.8}$$

$$\delta_{ij} = \boldsymbol{n}_{ij} \cdot \boldsymbol{a}_{ij} \tag{3.5.9}$$

$$\boldsymbol{d}_{ij} = \delta_{ij}\boldsymbol{n}_{ij} \tag{3.5.10}$$

$$\xi_i = \frac{\boldsymbol{a}_{ij} \cdot (\boldsymbol{v}_j \times \boldsymbol{n}_{ij})}{\|\boldsymbol{v}_i \times \boldsymbol{v}_j\|} \tag{3.5.11}$$

$$\xi_j = \frac{\boldsymbol{a}_{ij} \cdot (\boldsymbol{v}_i \times \boldsymbol{n}_{ij})}{\|\boldsymbol{v}_i \times \boldsymbol{v}_j\|} \tag{3.5.12}$$

sign correction of $\boldsymbol{n}_{ij}$, in order that $\delta_{ij} > 0$, being omitted.

Partially bound proximity

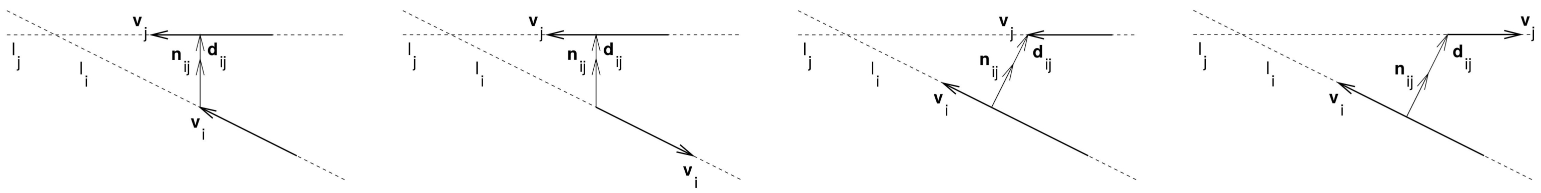
If $[\,\xi_i' \; \xi_j'\,] \notin \langle 0,1\rangle^2$, the proximity δ_{ij} of line-segments s_i, s_j may be equivalent to the smallest distance between an end-point of one the line-segments and the line upon which lies the other.

Potential configurations (figure 3.8) of *partially bound proximity* are checked as individual cases (a-d) of table 3.3 depending on the values of ξ_i' and ξ_j' producing $\boldsymbol{n}_{ij}''$, δ_{ij}'', $\boldsymbol{d}_{ij}''$ in sequence.

The algorithm is immediately stopped after a case satisfies $[\,\xi_i'' \; \xi_j''\,] \in \langle 0,1\rangle^2$. Whereupon the proximity of the two line-segments s_i, s_j is *partially bound* and $\boldsymbol{n}_{ij} = \boldsymbol{n}_{ij}''$, $\delta_{ij} = \delta_{ij}''$, $\boldsymbol{d}_{ij} = \boldsymbol{d}_{ij}''$, $\xi_i = \xi_i''$, $\xi_j = \xi_j''$

Case	Condition	$\boldsymbol{n}''_{ij}$	δ''_{ij}	$\boldsymbol{d}''_{ij}$	ξ''_i	ξ''_j
a	$\xi'_i < 0$	$\frac{(\boldsymbol{v}_j \times (\boldsymbol{A}_j - \boldsymbol{A}_i)) \times \boldsymbol{v}_j}{\|(\boldsymbol{v}_j \times (\boldsymbol{A}_j - \boldsymbol{A}_i)) \times \boldsymbol{v}_j\|}$	$(\boldsymbol{A}_j - \boldsymbol{A}_i) \cdot \boldsymbol{n}_{ij}$	$\delta_{ij}\boldsymbol{n}_{ij}$	0	$\boldsymbol{A}_i + \boldsymbol{d}_{ij} - \boldsymbol{A}_j$
b	$\xi'_i > 1$	$\frac{(\boldsymbol{v}_j \times (\boldsymbol{A}_j - \boldsymbol{B}_i)) \times \boldsymbol{v}_j}{\|(\boldsymbol{v}_j \times (\boldsymbol{A}_j - \boldsymbol{B}_i)) \times \boldsymbol{v}_j\|}$	$(\boldsymbol{A}_j - \boldsymbol{B}_i) \cdot \boldsymbol{n}_{ij}$	$\delta_{ij}\boldsymbol{n}_{ij}$	1	$\boldsymbol{B}_i + \boldsymbol{d}_{ij} - \boldsymbol{A}_j$
c	$\xi'_j < 0$	$\frac{(\boldsymbol{v}_i \times (\boldsymbol{A}_i - \boldsymbol{A}_j)) \times \boldsymbol{v}_i}{\|(\boldsymbol{v}_i \times (\boldsymbol{A}_i - \boldsymbol{A}_j)) \times \boldsymbol{v}_i\|}$	$(\boldsymbol{A}_i - \boldsymbol{A}_j) \cdot \boldsymbol{n}_{ij}$	$\delta_{ij}\boldsymbol{n}_{ij}$	$\boldsymbol{A}_j + \boldsymbol{d}_{ij} - \boldsymbol{A}_i$	0
d	$\xi'_j > 1$	$\frac{(\boldsymbol{v}_i \times (\boldsymbol{A}_i - \boldsymbol{B}_j)) \times \boldsymbol{v}_i}{\|(\boldsymbol{v}_i \times (\boldsymbol{A}_i - \boldsymbol{B}_j)) \times \boldsymbol{v}_i\|}$	$(\boldsymbol{A}_i - \boldsymbol{B}_j) \cdot \boldsymbol{n}_{ij}$	$\delta_{ij}\boldsymbol{n}_{ij}$	$\boldsymbol{B}_j + \boldsymbol{d}_{ij} - \boldsymbol{A}_i$	1

Table 3.3: Potential cases satisfying $[\,\xi''_i\ \xi''_j\,] \in \langle 0, 1 \rangle^2$



(a) (b) (c) (d)

Figure 3.8: Possible configurations of line-segments with *partially bound proximity*

Fully bound proximity

If neither approach yields such result, proximity of line-segments s_i, s_ij is *fully bound* and δ_{ij} is the minimal distance between two of their end-points:

$$\delta_{ij} = min(\|\boldsymbol{A}_j - \boldsymbol{A}_i\|, \|\boldsymbol{B}_j - \boldsymbol{A}_i\|, \|\boldsymbol{A}_j - \boldsymbol{B}_i\|, \|\boldsymbol{B}_j - \boldsymbol{B}_i\|) \tag{3.5.13}$$

with $\boldsymbol{d}_{ij}$ spanned between the two and scalars $[\,\xi_i\ \xi_j\,] \in \{0,1\}^2$. Vector $\boldsymbol{n}_{ij}$ can be then determined as:

$$\boldsymbol{n}_{ij} = \frac{1}{\delta_{ij}}\boldsymbol{d}_{ij} \tag{3.5.14}$$

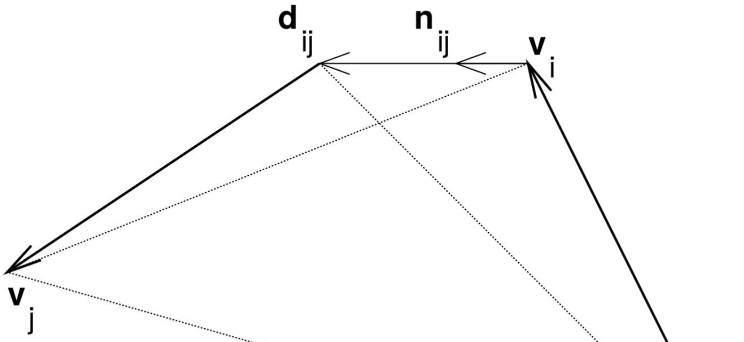


Figure 3.9: Example of *fully bound proximity* ($\delta_{ij} = \|\boldsymbol{A}_j - \boldsymbol{B}_i\|$)

Chapter 4

Application

The simulation model was created in *Matlab* and can be found in its entirety in the digital attachment along with trajectories, console write-outs and videos of performed simulations.

4.1 Program structure

In its entirety the simulation model consists of 41 purpose written `.m` functions their detailed description being beyond the extent of the thesis. Therefore in this section we will overview two functions: `CONTROL.m` and `skew_model.m` of the simulation model and functions called inside them providing a rough outline of the model's structure.

4.1.1 CONTROL.m

The main function of the program is CONTROL.m it first loads the setup of the manipulator from manipulator_setup.m and desired trajectory in the form of an array $\mathtt{POS} = \begin{bmatrix}\boldsymbol{X}_1 & \dots & \boldsymbol{X}_s\end{bmatrix}$ where $s \in \mathbb{N}$ is the number of discrete steps into which the trajectory is divided.

In a for i=1:s cycle POS(:,i) along with the manipulator set-up is fed into the function skew_model.m which returns matrices $\boldsymbol{E}$ (eqn. 3.4.5), $\boldsymbol{J}$ (eqn. 3.2.12) and $\boldsymbol{A}$ (eqn. 3.3.10) as well as arrays necessary for the visualization of the manipulator at each step s. Consequently *collision detection* and *singularity detection* are performed in real-time by functions collision_detection.m and singularity_detection.m.

```
...
for i = 1:size(POS,2)
        %tic
        pos = POS(:,i);
        % Skew model
                [ CS_uu, CS_LAMB, UNI_AA, UNI_vv, UNI_ny, UNI_delta,
                ↪  UNI_epsilon, UNI_n, UNI_Xi, PS_SEG, A, J ] =
                ↪  skew_model ( AA, cs_rho, ps_rho, pc_rho, rr,
                ↪  u_23, c_l, pos );
        % Collision detection
                [ epsilon_min, UNI_index, pair, pair_index ] =
                ↪  collision_detection ( i, UNI_epsilon, UNI_ny );
        % Singularity detection
                [ sig_J, sig_FC ] = singularity_detection ( i, A, J,
                ↪  1e-3, 1e-3 );
...
```

Listing 4.1: exert from function CONTROL.m

The arrays for visualization are stacked along an additional dimension into _s arrays inside the cycle. After the trajectory is checked for singularities and self-collision the _s arrays are loaded into the function plot_manipulation which plots the manipulator along the trajectory POS.

4.1.2 `skew_model.m`

We may divide the function `skew_model.m` into three distinct parts. In the first part homogeneous transformation matrices $\boldsymbol{T}_{12}$ (eqn. 3.1.1) and $\boldsymbol{T}_{13}$ (eqn. 3.1.2) are calculated as well as vectors $\boldsymbol{u}_i$, $i \in \langle 1, 8\rangle$ described in section *mechanical model* (3.1.1). This is done simultaneously with the calculation of vectors necessary for the creation of the *collision model* 3.1.2.

In the second part of the function matrices $\boldsymbol{J}$ (eqn. 3.2.12) and $\boldsymbol{A}$ (eqn. 3.3.10) are calculated using the function `struct_n_Jacob_matrices.m` for which the vectors $\boldsymbol{b}_i$, $i \in \langle 1, 8\rangle$ are created by function `manipulator_vects.m`.

The third part contains functions: `cable2cable_prox.m`, `cable2platforms_prox.m`, `cable2columns_prox.m` where the *line segment proximity algorithm* (section 3.5) is applied through the function `segment_prox.m`, creating matrices $\boldsymbol{E}_I$, $\boldsymbol{E}_{II}$, $\boldsymbol{E}_{III}$ as well as arrays containing scalars ξ_i, ξ_j and vectors $\boldsymbol{n}_{ij}$, $\boldsymbol{d}_{ij}$ which will be used in `plot_manipulation.m`.

4.2 Performed simulations

For the simulations we used an arbitrary manipulator set-up (listing 4.2).

```
function [ AA, cs_rho, ps_rho, pc_rho, rr, u_23, c_l ] =
↪   manipulator_setup
% Frame anchoring points
        AA = [          10        0         8;
                        10        10        8;
                        0         10        8;
                        0         0         8;
                        10        0         4;
                        10        10        4;
                        0         10        4;
                        0         0         4        ]';
% Manipulator anchoring points' vectors
        rr = [           3        -5        0;
                         3         5        0;
                        -3         5        0;
                        -3        -5        0;
                         5        -3        0;
                         5         3        0;
                        -5         3        0;
                        -5        -3        0        ]' / 4;
% Prismatic joint
        u_23 = [0;0;1]; % axis
        c_l = 4.5; % max lenght
% Capsule radii
        cs_rho = 0.1; % cable capsule radius
        ps_rho = 0.2; % platform segments capsule radius
        pc_rho = 0.5; % platfrom columns capusle radius
```

Listing 4.2: function `manipulator_setup.m`

4.2.1 Translation along z axis

The main advantage of the proposed manipulator in comparison to other RRPM is its capability to operate below the plane delimited by the lower four cable exit points.

Two simulations a) and b) of 200 steps were carried out to validate this claim. In both of the simulations we start from identical configurations $\boldsymbol{X}_{1_a} = \boldsymbol{X}_{1_b}$ (figure 4.1):

$$\boldsymbol{X}_{1_a} = \boldsymbol{X}_{1_b} = \begin{bmatrix} 5 & 5 & 4.5 & 0 & 0 & 0 & 2 \end{bmatrix}^T \tag{4.2.1}$$

and start descending towards the desired configurations:

$$\boldsymbol{X}_{201a} = \begin{bmatrix} 5 & 5 & 0.5 & 0 & 0 & 0 & 4 \end{bmatrix}^T \tag{4.2.2}$$

$$\boldsymbol{X}_{201b} = \begin{bmatrix} 5 & 5 & 0.5 & 0 & 0 & 0 & 2 \end{bmatrix}^T \tag{4.2.3}$$

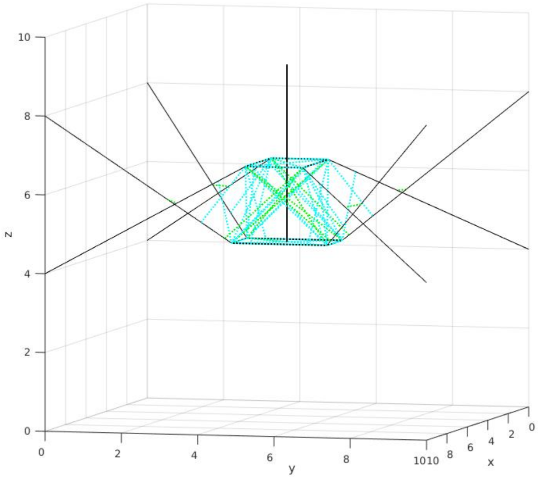


Figure 4.1: Configuration $\boldsymbol{X}_{1_a} = \boldsymbol{X}_{1_b}$

After configurations $\boldsymbol{X}_{101_a} = \boldsymbol{X}_{101_b}$ (figure 4.2):

$$\boldsymbol{X}_{101_a} = \boldsymbol{X}_{101_b} = \begin{bmatrix} 5 & 5 & 2.5 & 0 & 0 & 0 & 2 \end{bmatrix}^T \tag{4.2.4}$$

the two simulations start to differ as in simulation a) we start extending the platform at the rate of the descend while in simulation b) the geometry of the platform remains constant.

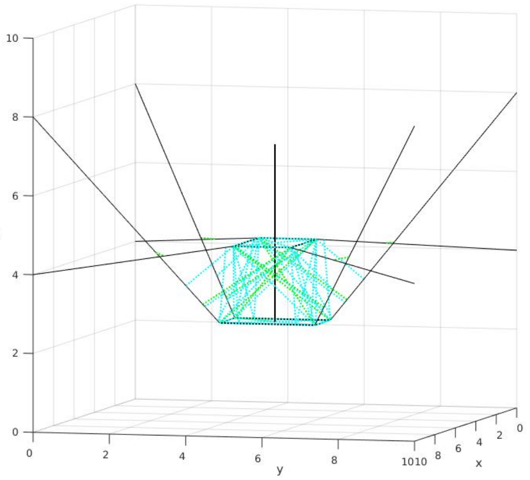


Figure 4.2: Configuration $\boldsymbol{X}_{101_a} = \boldsymbol{X}_{101_b}$

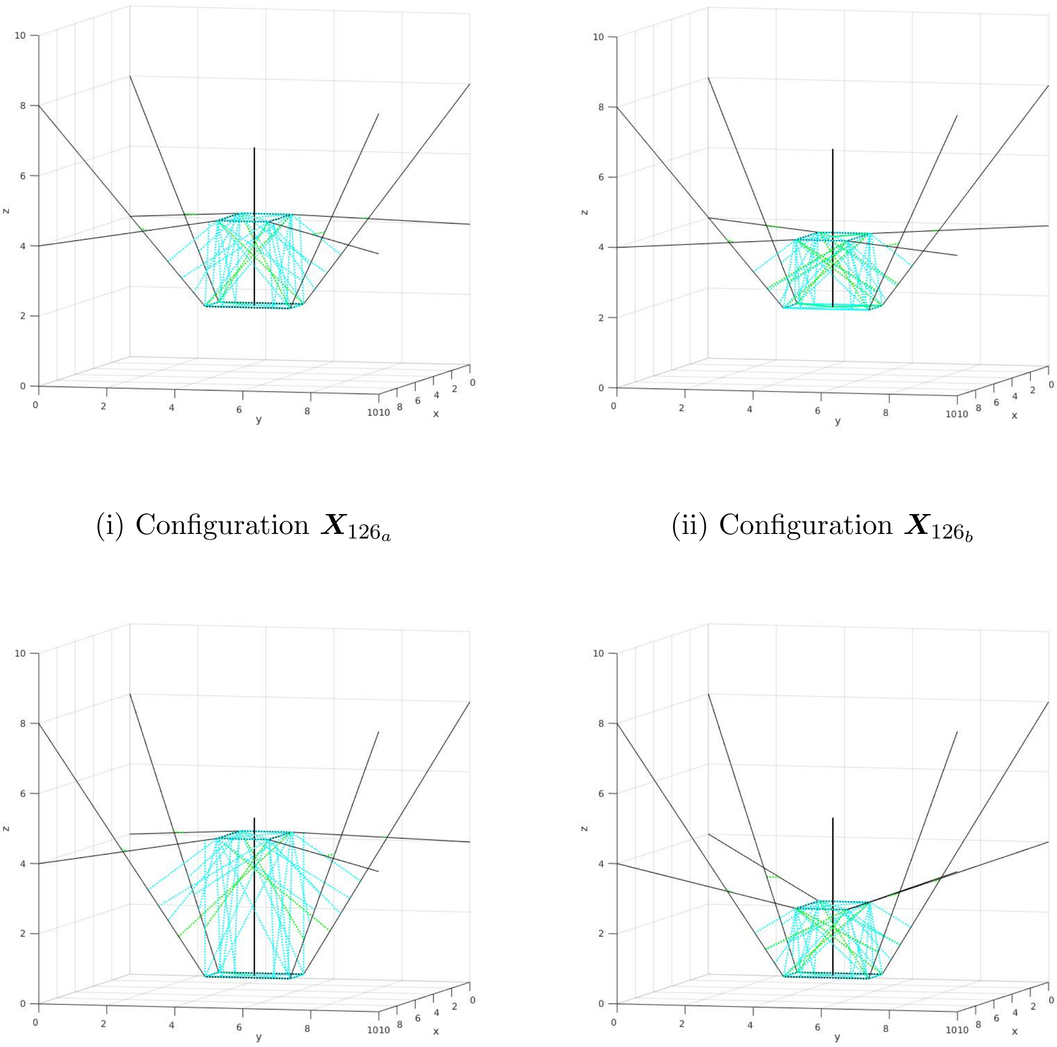
Inevitably in the simulation b) configurations from $\boldsymbol{X}_{126_b}$ (figure 4.3ii), where the lower four cables lie in one plane:

$$\boldsymbol{X}_{126_b} = \begin{bmatrix} 5 & 5 & 2 & 0 & 0 & 0 & 2 \end{bmatrix}^T \tag{4.2.5}$$

to $\boldsymbol{X}_{201_b}$ (figure 4.3iv) are *force-closure singular* while simulation a) reaches the desired configuration $\boldsymbol{X}_{201_a}$ (figure 4.3iii):

$$\boldsymbol{X}_{201_a} = \begin{bmatrix} 5 & 5 & 0.5 & 0 & 0 & 0 & 4 \end{bmatrix}^T \tag{4.2.6}$$

without the occurrence of singularities or self-collisions proving the claim.



(i) Configuration $\boldsymbol{X}_{126_a}$

(ii) Configuration $\boldsymbol{X}_{126_b}$

(iii) Configuration $\boldsymbol{X}_{201_a}$

(iv) Configuration $\boldsymbol{X}_{201_b}$

Figure 4.3: Configurations of simulations a) and b)

4.2.2 Rotation around x axis

It could be argued that the configuration $\boldsymbol{X}_{201_a}$ (figure 4.3iii) of the previous pair of simulations could be achieved by a manipulator with constant platform geometry equivalent to the geometry of the proposed manipulator when $s_{23} = 4$. While this is true, configurations in which the platform is retracted are advantageous when rotating along the x and y axis and could theoretically decrease strain on the platform when resisting certain wrenches.

With the current simulation model we can prove the former advantage by comparing two simulations c) and d) starting in configurations $\boldsymbol{X}_{1_c}$ (figure 4.4i) and $\boldsymbol{X}_{1_d}$ (figure 4.4ii):

$$\boldsymbol{X}_{1_c} = \begin{bmatrix} 5 & 5 & 5 & 0 & 0 & 0 & 2 \end{bmatrix}^T \tag{4.2.7}$$

$$\boldsymbol{X}_{1_d} = \begin{bmatrix} 5 & 5 & 5 & 0 & 0 & 0 & 4 \end{bmatrix}^T \tag{4.2.8}$$

and rotating along the x axis towards configurations $\boldsymbol{X}_{101_c}$ and $\boldsymbol{X}_{101_d}$:

$$\boldsymbol{X}_{101_c} = \begin{bmatrix} 5 & 5 & 5 & \pi/8 & 0 & 0 & 2 \end{bmatrix}^T \tag{4.2.9}$$

$$\boldsymbol{X}_{101_d} = \begin{bmatrix} 5 & 5 & 5 & \pi/8 & 0 & 0 & 4 \end{bmatrix}^T \tag{4.2.10}$$

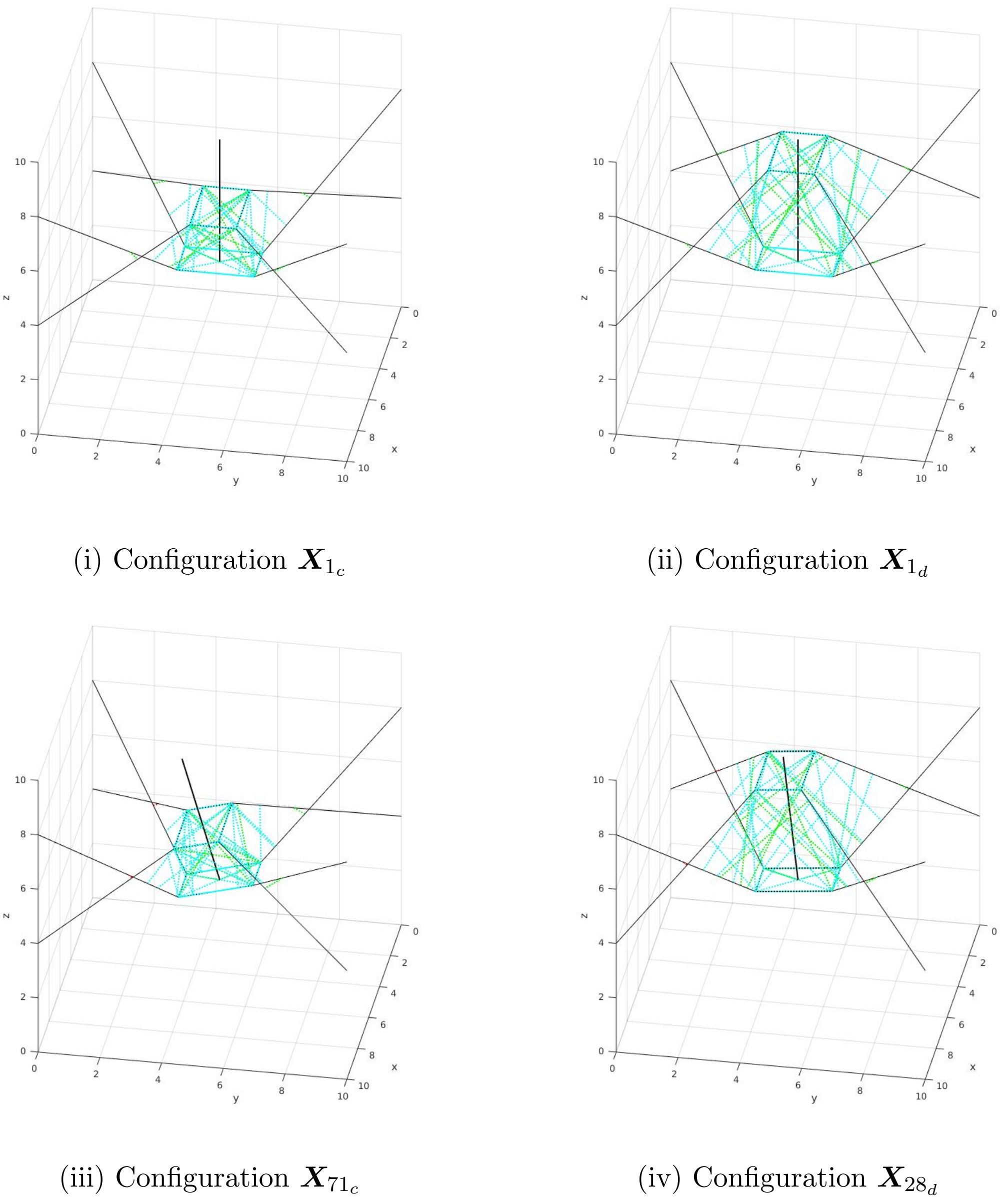
In simulation d) first self collision occurs at step num. 28 between cables 1&5 and 4&8 in configuration $\boldsymbol{X}_{28_d}$ (figure 4.4iv):

$$\boldsymbol{X}_{28_d} = \begin{bmatrix} 5 & 5 & 5 & 0.10603 & 0 & 0 & 4 \end{bmatrix}^T \tag{4.2.11}$$

making further rotation impossible, while in the simulation c) self collision occurs at step num. 71 in configuration $\boldsymbol{X}_{71_c}$ (figure 4.4iii):

$$\boldsymbol{X}_{71_c} = \begin{bmatrix} 5 & 5 & 5 & 0.27489 & 0 & 0 & 2 \end{bmatrix}^T \tag{4.2.12}$$

Allowing for approximately further 10 degrees of rotation along the axis.



(i) Configuration $\boldsymbol{X}_{1_c}$

(ii) Configuration $\boldsymbol{X}_{1_d}$

(iii) Configuration $\boldsymbol{X}_{71_c}$

(iv) Configuration $\boldsymbol{X}_{28_d}$

Figure 4.4: Configurations of simulations c) and d)

Chapter 5

Conclusion

First part of the thesis is dedicated to the overview of kinematics and the static equilibrium of *CDPMs* as well as restrictions of their workspaces. Two types of singularities effecting *CDPMs* are described: *Jacobian singularities* and *force-closure singularities* derived from kinematic relations and equations of static equilibrium respectively. Further a suitable approach to collision detection in *CDPMs* is identified and the effect of joint limits is discussed.

The rest of the thesis is focused on a particular *CDPM* of novel design. Based on typical eight cable *RRPMs* the manipulator differs by possessing a platform with adjustable geometry. Such property of the platform introduces kinematic redundancy to the mechanism, manifesting in changes to both kinematic relations and equations of static equilibrium in comparison to typical *CDPMs*, requiring them to be derived independently. The purpose of such design is the extension of the manipulator's workspace which is confirmed in section 4.2. Beside the mechanical model a collision model had to be created for these purposes. It uses a novel algorithm for calculating the minimal distance between two line-segments in space which is in turn applied for the identification of interference between *capsules* forming the model.

The extension of the workspace as result of the adjustable platform geometry is confirmed through a series of simulations. We prove that by scaling the platform up the manipulator is able to operate in non-singular configurations outside of the typical workspace delimited by points through which its cables exit the workspace while scaling the platform down increases the range of rotation along certain axis. The two sets of simulations also demonstrate the model's capability of detecting singular configurations as well as self collision in a given pose.

A significant improvement to collision detection could be achieved by developing a method for identifying *tunnelling* a phenomenon during which a collision is not registered as the objects in the simulation interfere only for a period in between two steps of the manipulation. Progress on developing such method along with algorithms for collision detection between capsules and other objects such as cuboids has already been made in anticipation of further research into the topic.

Naturally the next step towards creating a functional physical *CDPM* would be the inclusion of dynamics into the mechanical model as well as applying an approach to force distribution among the cables of the manipulator. Some factors which should be also considered in further modelling of the manipulator are the geometry of cable exit points, cable elasticity and slack.

Bibliography

[1] Valentin Lorenz Schmidt. *Modeling techniques and reliable real-time implementation of kinematics for cable-driven parallel robots using polymer fiber cables*. PhD thesis, 2017.

[2] Jean-Baptiste Izard, Marc Gouttefarde, Micaël Michelin, Olivier Tempier, and Cédric Baradat. *A Reconfigurable Robot for Cable-Driven Parallel Robotic and Industrial Scenario Proofing*, pages 135–148. 01 2013.

[3] Philipp Miermeister, Werner Kraus, Tian Lan, and Andreas Pott. *An Elastic Cable Model for Cable-Driven Parallel Robots Including Hysteresis Effects*, volume 32, pages 17–28. 08 2015.

[4] Jean-Pierre Merlet. *Parallel Robots*. Springer, 2 edition, 2006.

[5] A. Taghavi, S. Behzadipour, N. Khalilinasab, and H. Zohoor. Workspace improvement of two-link cable-driven mechanisms with spring cable. In A. Pott and A. Bruckmann, editors, *Cable-Driven Parallel Robots*. Springer International Publishing, 2013.

[6] Micaël Michelin, Cédric Baradat, Dinh Quan Nguyen, and Marc Gouttefarde. Simulation and control with xde and matlab/simulink of a cable-driven parallel robot (cogiro). *Mechanisms and Machine Science*, 32, 08 2015.

[7] Xiumin Diao and Ou Ma. A method of verifying force-closure condition for general cable manipulators with seven cables. *Mechanism and Machine Theory*, 42:1563–1576, 12 2007.

[8] R. Chellal, E. Laroche, and Gangolff J. Cuvillon, L. An identification methodology for 6-dof cable-driven parallel robots parameters application to the inca 6d robot. In A. Pott and A. Bruckmann, editors, *Cable-Driven Parallel Robots*. Springer International Publishing, 2013.

[9] Werner Kraus, Philipp Miermeister, and Andreas Pott. Investigation of the influence of elastic cables on the force distribution of a parallel cable-driven robot. In A. Pott and A. Bruckmann, editors, *Cable-Driven Parallel Robots*. Springer International Publishing, 2013.

[10] Hui li, Xinyu Zhang, Rui Yao, Jinghai Sun, Gaofeng Pan, and Wenbai Zhu. *Optimal Force Distribution Based on Slack Rope Model in the Incompletely Constrained Cable-Driven Parallel Mechanism of FAST Telescope*, pages 87–102. 01 2013.

[11] Ross MacAusland. The moore-penrose inverse and least squares. In *MATH 420: Advanced Topics in Linear Algebra*. University of Puget Sound.

[12] Jean-Pierre Merlet. Legs interference checking of parallel robots over a given workspace or trajectory, 05 2006.

[13] Xiumin Diao, Ou Ma, and Qi Lu. Singularity analysis of planar cable-driven parallel robots. pages 272 – 277, 10 2008.

[14] Richard M. Murray, Zexiang Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 03 1994.

[15] Andreas Pott, Hendrik Mütherich, Werner Kraus, Valentin Schmidt, Philipp Miermeister, and Alexander Verl. *IPAnema: A family of Cable-Driven Parallel Robots for Industrial Applications*, volume 12, pages 119–134. 01 2013.

[16] Rajnoha. Object positioning in 3d space using parallel cable-driven robot. Master's thesis, Brno University of Technology, Brno, 2016.

[17] Kraus. Redundantly actuated cable-driven manipulator. Master's thesis, Czech Technical University in Prague, Prague, 2016.

[18] Wikipedia contributors. Condition number — Wikipedia, the free encyclopedia, 2020. [Online; accessed 22-Feb-2020].