POLITECNICO DI TORINO

Master of Science in Mechatronics engineering

Master of science thesis

Robot manipulation models to exploit environmental constraints

October 2019

Supervised by:

Prof. Paolo Prinetto

Prof. Domenico Prattichizzo

João Bimbo PhD

Author:

Mehrdad Tavassoli

To My family who always support me throughout my life and my friends who inspire me

Special thanks to João Bimbo who taught me a lot in my thesis as a friend and mentor

Abstract

In this thesis, I focus on modelling the grasp by replicating human's maneuvers to grasp and manipulate the objects that are not easily graspable using the conventional moves like top grasp. This can be done by exploiting the environmental constraints around the robot. A robot might push an object against a wall (Wall-grasp) to facilitate lifting it, or drag it to the edge of a table (Edge-grasp) to be able to grasp it from the side or topple to reorient the object (Toppling) to have a better grasp. In this thesis wall-grasp and Toppling are studied.

A novel approach is used that includes the environmental constrains as part of grasp that is included in grasp matrix. Based on permissible movements of the object in each scenario, the grasp matrix applied to take the constrains to model the grasp in to account. Gravity is also included by modeling its effect with another finger that pulls the object down which contrasts with usual methods like gravity compensator. The constraints are then fed it the optimizer that minimizes the contact forces that is required to lift the object up to fulfil the desired pose. Finally, after being tested by dynamic simulator, the data are ready to be applied to the soft robot to verify the simulation and model that has been acquired. Withstanding the noticeable outcomes, there exist a long way to the goal. Eventually, the method is evaluated and some solutions are proposed to elevate the dexterity of the system.

Table of Contents

Table of Figures

Table of tables

1. Introduction

1.1. Problem to be solved and purpose

Grasping and manipulation, ever since its development, emerged to be an autonomous subset of robotic studies. Although it appears trivial, grasping and manipulation, especially multi-finger grasping, is still challenging for the roboticists. As it involves contacts between two objects, it has non-linear nature which makes it very hard to model.

The essence of the robotic grasping and manipulation is to have a generic approach for the robot to handle arbitrary objects in an unstructured environment that is subjected to disturbances. In other words, the grasp need to be robust and generic. It is desired that the system to be as close as possible to its smarter biological counterpart (human arm which is very complicated. Moravec's paradox clearly depicts the difficulty involved: "It is comparatively easy to make computers exhibit adult level performance on intelligence tests or playing checkers, and difficult to give them the skills of a one-year old when it comes to perception and dexterity" [2].

To achieve higher dexterity, the biological structure of human hand (**compliance)** is used along with the strategies he uses to handle objects that are not approachable in conventional modeling. In this scenario, obstacle exploitation is seeming to be a breakthrough. This not only makes the robot hand less sensitive in comparison with a fully actuated robotic hand but also gives a descent accuracy. It is worthwhile mentioning that it is computationally far easier to have an under-actuated soft hand.

To fulfill the model for the grasp that possesses the abovementioned criteria, a conventional physics simulator is used that capture maps the twists and wrenches from the object center of mass (**COM**) to the contact point. Due to hysteresis property of compliance models that makes the contact behavior unpredictable, a rigid contact model is applied even though it induces static indeterminism to the problem.

In this work, a convectional physics simulator is favored than the new machine-learning approaches because: first their need to have a huge database to get a reliable result (around 100 **KUKA® No.7** that are in the same *condition* and doing the same task) that even if possible very difficult to obtain. Besides there are found some data that are not coherent to what is needed that makes machine learning techniques like deep learning inapplicable to our cause.

1.2. Thesis Organization

In this thesis, **chapter 2** is dedicated to an introduction to Method and theory in which the theoretical background of what is going to be explained is presented in full detail.

Chapter 3 is the cornerstone of this thesis that explains the method that is used along with the consideration and assumption that has be made.

In **Chapter 4**, result and conclusion is presented that contains both the simulation results obtained by **Matlab®** and what is obtained in practice acquired by force-torque sensor.

Lastly, **in Chapter 5**, conclusion of what has be done and the possibilities for improvement in the future is presented.

To become familiar with the experimental setup please refer to the appendix at the end of this thesis after the references.

2. Related works and the state of the art

Initially, researchers who were studying dexterous manipulation, start their analysis based on an acquired grasp. Okada (1982) controlled the robotic hand to screw a nut or Kobayashi's robot that draw a plain figure using a pencil (1984). In 1984 Kerr, developed the differential equation for the dexterous manipulation. In all the mentioned researches, the contacts are rolling on one another. As this modulation requires force/position control, this potentially limits the maneuvers and dexterity of the manipulation [3].

Cole (1989) [4] drive the kinematic and control of the multi-fingered hand of two arbitrary shape objects having contact with each other which is a generalized version of computed torque method of robot manipulator.

Unlike what is presented by Cole, Trinkle and Paul (1990) [3] studied the planning for dexterous manipulation and grasp considering the sliding contacts between the hand, the object and the support (the obstacles). A graphical approach is presented. Based on what Trinkle and Paul did, the object perimeters divided in to regions including **squeeze region** in which the object is squeezed without any movement, translate that the object slides over the contact, **breaking regions** to the supports (obstacles) in which the object is partially lifted and finally **liftability regions** that explains the part of the objects perimeter where if contact force of the hand is applied. The mentioned regions are subjected to changes based on the objects geometrical features as well as the position where the contact forces are applied, the reaction forces from the support and the gravity factor.

Erdmann (1991) [5], constructed a generic cone by imbedding the conventional friction constraints from real space into the bodies' configuration space. The mentioned presentation is applied to obtain a simple computational method for getting all the permissible movement of the object subjecting to the applied wrench to the body.

Buss, Hashimoto and Moore (1996) [6] has transferred the friction cone constraints into positive-definite constraints in form of matrixes. Their work followed by Li Han, Jeff C Trinkle and Zexiang Li (2001) [7] that put the friction constraints into **Linear Matrix Inequalities (LMIs)** in a **convex optimization problem.**

D. Prattichizzo (2013) [8] studied the total number of synergies is needed to control a compliance hand based on the task and the hand's structure to have a more precise and stable grasp.

Dafle et al (2014) [9] studied the possibility of re-grasp like in-hand-manipulation and proved its possibility even for a very simple gripper. 12 different strategies have been developed by applying it them to 3 different type of object. that is backed by 1200 trial for sake of efficiency improvement. In the long run, it is expected to have repositories out of the mentioned strategies to evolve them into a general-purpose approach in hand manipulation.

Based on what Bicchi et al (2015) [10] did, for planning the soft hand (Pisa/IIT hand), the object can be decomposed into basic geometries, also known as **Minimum Volume Bounding Boxes (MVBB)**. Soft hands can easily confirm itself into the mentioned bounding boxes, this in turn, leads into a set of hand posture. If these posse are exposed to the corresponding dynamic simulator of the hand, it is possible to evaluate whether it is a good grasp or not.

Maria Pozzi (2016) [11] evaluated the quality of the under actuated, synergy-driven hand has been measured using the same criterion that has been applied to the fully actuated hand.

To better implement the soft, under actuated hand like Pisa/IIT hand, human grasp behavior is taken as the role model to exploit the compliance contact between the hand and the ambient to compensate for the uncertainty involves in the process of grasping is investigated by Fabian Heinemann et al (2016) [12].

The compliant hand has been modeled to capture the way the hand (Pisa/IIT hand) closes and opens by measuring the motion of appropriate reference points on the fingers. This gives the preferred grasp direction of a hand under consideration that is **called closure signature** (**CS**) (Maria Pozzi, Gionata Salvietti, João Bimbo Monica Malvezzi, and Domenico Prattichizzo ,2018) [13].

One of the kind of manoeuvre the robot can replicate is edge grasp. M.Ghazaei (2019) [14] Studied the possible motion of a flat object generated by frictional contacts under quasi-static analysis that predicts the motion of a flat object applying a hybrid dynamical system. The model can be used for object reorientation to having a secure grip for the edge grasp. The other manoeuvres are toppling and wall-grasp. Mason et al (2019) [15] sued a hybrid control approach to control the object orientation and contact maintenance. A velocity controller is used to harness the position and orientation of the object whilst, a comprehensively less rigorous force controller that is perpendicular to the previous one is there just to be sure of contact maintenance between the fingertip and the object. To model the grasp, he applied the jacobian of holonomic constrains.

Kroemer [16] studied the possibility of machine learning in grasping and manipulation but it proved to be incompatible for modelling the grasp. Finally, in Tossing bot (2019)

[17] applied a hybrid approach based on conventional physic simulator and deep reinforced approach to pick and throw objects of different shapes into some a set of boxes one for learning and another to validate what is learned by the robot. They use a formulation deep networks are used to put the control parameters on top of what is already obtained by a mentioned physical simulator. The whole process is overseeing by visual observation (bin). It has been proven that the hybrid approach has the accuracy around 85%.

3. Theory and Methodology

3.1. Introduction

To replicate the dexterity of human hand as a superior manipulator, it is of great importance that the robot has the same characteristics of human hand. In this regard, there are several approaches in which a human may take to grasp an object.

Toppling and wall-grasp are among the strategies human uses to grasp objects with specific characteristics. Basically, if the height of the object is relatively low, applying conventional approaches is not easy if not impossible.

Toppling makes the object to be flipped to reorient the object for a more secure grasp. In this case, a force is applied against the support the object laid on (e.g. floor, table). Based on the friction between and force robot applies, the object flips over one edge. The forces applied by the finger tips and the reaction table applies to the object result in a force in the **COM**[1] which in turn makes the object turn around one edge.

In wall-grasp, the object is pushed to the wall or any other similar constraint. The contact wrenches and the reaction, the object receives from the constraint applies a force at the COM of the object that lift the object up. Depending on the friction between surfaces, direction of the contact force from the fingertips of the robots and its magnitude, the object may either slides over the constraints or tip over the contact point.

A mathematical model is used to foresee the hand and object interactions in terms of the forces they exchange under different loading conditions. The most desirable property to be considered during the grasping and manipulation is grasp maintenance. There are found some disturbances, mainly related to the inertia that are appreciable during high speed manipulation that may cause an unsecure grasp.

The studied manipulation tasks are subjected to the quasi-static loading condition thus quasi-static is considered in this thesis as a governing basis.

[1] Center of mass

3.2. Theoretical basis and background information:

To have a better understanding of the method that has been used in this thesis, it is important to understand the spirit of it which is the theories applied in this thesis. To obtain the mathematical model a grasp matrix G and hand Jacobian J are is used to map the wrenches and twists from COM to the contact points and map the joint loads τ_i from the joint space to the contact points in the same order.

As the evolution of position state over the course of time is very slow quasi static loading is adopted in the study of the grasp.

The whole problem is restricted under two type of constrains which are general constrains and task-imposed constraints that are tailored to the task under consideration. The mentioned restrictors are all applied to a linear optimizer as we are dealing with a linear problem to render the forces that need to be applied to the object to reorient the object to the desired pose. For this purpose, the permissible movement of the object is considered that plays a pivotal role in the modeling. Figure 1 illustrates the main quantities and materials for modeling the grasping:

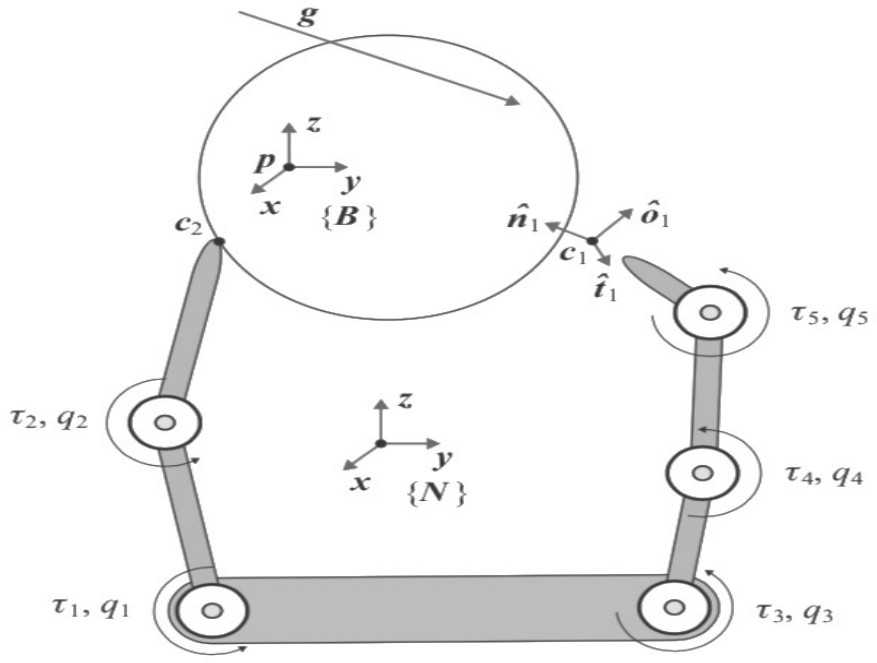

Figure 1- Main quantities of the grasp matrix [18]

Assume the components making the hands including links and joints are of rigid materials. In **Figure 1, {ℕ}** is the inertial frame that is congenitally chosen. **{𝔹}** is the body frame that is, again, arbitrary chosen. However, it is better to be in the bodies'

center of mass for simplicity. Lastly, the final reference frame that must be considered is the local frame at each contact points $\{\mathbb{C}_i\} \in \mathbb{R}^{3\times3}$ comprising from three unit vectors of $\{\hat{\mathbf{n}}_i, \hat{\mathbf{t}}_i, \hat{\mathbf{o}}_i\}$ that are mutually perpendicular to one another. The body frame $\{\mathbb{B}\} \in \mathbb{R}^{3\times3}$ is defined with respect to the inertial frame $\{\mathbb{N}\}$ through the vector $\boldsymbol{p} \in \mathbb{R}^{3\times3}$. c_i serves for the same purpose for the $\{\mathbb{C} \in \mathbb{R}^{3\times3}\}$. Position and orientation of $\{\mathbb{B}\}$ is represented by vector $\mathbf{u} \in \mathbb{R}^{n_u}$,again with respect to $\{\mathbb{N}\}$. For planar case $\mathbf{n_u = 3}$ but for 3D it can be **3** or **4** (in case of Euler angle representation it is 3 whiles there must another element be added to the other three elements for orientation for unit quaternion).

On the joint side, joints are labeled from palm to the tip from the 1st finger to the last from **1** to $\mathbf{n_q}$. The general coordinates which is the displacement of the joint (angle for the rotational joints and displacement for the prismatic one or both in case of hybrid joints like spherical joints) denote by $\boldsymbol{q} = [q_1, \ldots, q_{nq}]^T \in \mathbb{R}^{nq}$. The torques for revolute joints and force for prismatic joints are considered as joint loads that is denoted by $\boldsymbol{\tau} = [\tau_1, \ldots, \tau_{nq}]^T \in \mathbb{R}^{nq}$. The mentioned loads can be resulted various origin from the joints actions, inertia forces to even the interaction between the object and the finger tips at the contact points. However, it is convenient the joint loads to be separated into two contributors; those of contact in origin and the all the other contributors. In this thesis only non-contact joint loads are represented by $\boldsymbol{\tau}$ [18].

Twist and wrench of the object are represented by $\boldsymbol{\nu} = [\boldsymbol{v}^T \ \boldsymbol{\omega}^T]^T \in \mathbb{R}^{nv}$ and $\mathbf{g} = [\mathbf{f}^T \ \mathbf{m}^T]^T \in \mathbb{R}^{nv}$ respectively are with respect to the inertial frame $\{\mathbb{N}\}$ formed from translational component $\boldsymbol{v}$ of the center of mass (denoted by $\boldsymbol{p}$) and the angular component $\boldsymbol{\omega}$ [18]. Either twist and wrench can be presented in any other reference frame based on what is required in the situation. For this purpose, the action line of the force should be translated until it is reaches the origin of the new frame. Finally, due to force translation, a moment should be added.

Likewise, joint loads $\boldsymbol{\tau}$, $\mathbf{g}$ only stands for the non-contact wrenches. It is of great importance to note that the derivation of the object position and orientation $\mathbf{u}$ doesn't provides us the twist (i.e. $\dot{\boldsymbol{u}} \neq \boldsymbol{\nu}$) but the two mentioned variables are related to one another by **Error! Reference source not found.** In planar case, the twist is comprising rom translational velocity $\boldsymbol{v} \in \mathbb{R}^2$ and an angular velocity $\boldsymbol{\omega} \in \mathbb{R}$. In 3D space, however, $\boldsymbol{v} \in \mathbb{R}^3$ and an angular velocity $\boldsymbol{\omega} \in \mathbb{R}^3$. Same thing applies for the wrench of the object for either planar or 3-D spaces.

$$\dot{u} = V\nu \tag{1}$$

Where $\mathbf{V} \in \mathbb{R}^{n_u \times n_v}$ is relating the object's position and orientation derivation to object's twist. $\mathbf{V}$ is orthogonal that means $\boldsymbol{V^T V = I}$ which $\mathbf{I}$ is the identity matrix. Note that in planar systems $\mathbf{V} \in \mathbb{R}^{3\times3}$. [[18](38.11)].

The main notation of what is explained before is summed up in the following table:

Table 1 - Main notation of the grasp analysis – inspired by [18]

Notation	Definition
$\mathbf{n_c}$	Number of the contacts
$\mathbf{n_q}$	Number of Joints in the robotic hand
$\mathbf{n}_\nu$	Number of DOFs[1] of the object
$\mathbf{n}_\lambda$	Number of contact wrench component
$q \in \mathbf{R^{n_q}}$	Joint displacements
$\dot{\mathbf{q}} \in \mathbf{R^{n_q}}$	Joint velocities
$\tau \in \mathbf{R^{n_q}}$	Non-contact joint loads
$u \in \mathbf{R^{n_u}}$	Position and orientation of the object
$\nu \in \mathbf{R^{n_\nu}}$	Twist of the object
$g \in \mathbf{R^{n_\nu}}$	Non-contact object wrenches
$\lambda \in \mathbf{R^{n_\lambda}}$	Transmitted contact wrenches
$\nu_{\mathbf{cc}} \in \mathbb{R}^{\mathbf{n_\lambda}}$	Transmitted contact twists
$\{\mathbb{B}\}$	Frame fixed in the object (object frame)
$\{\mathbb{C}\}_\mathbf{i}$	Frame at the i^{th} contact
$\{\mathbb{N}\}$	Inertial frame

3.2.1. Velocity kinematics

To study the velocity kinematics of the grasp and grasp analysis in general two matrices are of great importance **G** and **J** that respectively stands for the grasp matrix and hand Jacobian.

[1] Degree of Freedom

As it was previously mentioned, the mentioned matrices define the relative kinematic and force transmission between the hand and the object at the contact points [18].

In the analysis, each contact point should be considered as two overlapping points one on the object and the other on the finger of the end effector. While hand Jacobian **J** transpose the joint velocities to the contact point, the transpose of the grasp matrix $\mathbf{G^T}$ maps the object twist at the body frame to the contact points [18].

3.2.1.1. Grasp matrix G [18]:

To drive grasp matrix G, consider the following:

$$\begin{pmatrix} \mathcal{v}^{N}_{i,obj} \\ \omega^{N}_{i,obj} \end{pmatrix} = P_i^T \nu \tag{2}$$

In **Eq. (2)**, the left side is the twist of the object at the $\{\mathbb{C}_i\}$ with respect to the inertial frame $\{\mathbb{N}\}$ comprising the angular and the linear velocities and the right side of the mentioned equation, P_i is:

$$P_i = \left(\begin{bmatrix} I_{3\times3} & 0 \\ \mathbf{S}(c_i - p) & I_{3\times3} \end{bmatrix} \right) \tag{3}$$

In **Eq. (3)**, $\mathbf{I_{3\times3} \in \mathbb{R}^{3\times3}}$ is the identity matrix, and $\mathbf{S(c_i - p)}$ is the cross-product of the distance between the $\mathbf{i^{th}}$ contact point $\mathbf{c_i}$ to the location of the origin of the body frame which is the centre of the mass for the mentioned reason here in this paper.

$$S(c_i - p) = \begin{pmatrix} 0 & -(c_i - p)_z & (c_i - p)_y \\ (c_i - p)_y & 0 & -(c_i - p)_x \\ -(c_i - p)_y & (c_i - p)_x & 0 \end{pmatrix} \tag{4}$$

To represent the object twist in the local frame of the $\mathbf{i^{th}}$ contact $\mathbf{\{\mathbb{N}\}}$, it is enough to multiply the left-hand side of **Eq. (2)** to rotational matrix $\mathbf{R = [\hat{n}_i, \hat{t}_i, \hat{o}_i] \in \mathbb{R}^{3\times3}}$ that its columns are the unit vectors of the $\mathbf{\{\mathbb{C}_i\}}$ with respect to $\mathbf{\{\mathbb{N}\}}$. Consequently, the twist of the object in the local frame $\mathbf{\{\mathbb{C}_i\}}$ would be the following:

$$\nu^{C_i}{}_{i,obj} = \overline{R}_i^T \begin{pmatrix} v^{N}_{i,obj} \\ \omega^{N}_{obj} \end{pmatrix} \tag{5}$$

As $\mathbf{R_i}$ must be multiplied to each, $\mathbf{\overline{R}_i^T}$ is defined like the following:

$$\overline{R}_i^T = \begin{bmatrix} R_i & 0 \\ 0 & R_i \end{bmatrix} \in \mathbb{R}^{6\times6}$$

Partial grasp matrix $\mathbf{\widetilde{G}_i^T}$ can be obtained by replacing the objects twist with respect to

$\{\mathbb{N}\}$ from **Eq. (2)** to the **Eq. (5)**. This matrix maps the objects twist to the local frame $\{\mathbb{C}_i\}$:

$$v_{i,obj}^{C_i} = \widetilde{G}_i^T \nu \tag{6}$$

where:

$$\widetilde{G}_i^T = \overline{R}_i^T \mathbf{P}_i^T \tag{7}$$

However, based on the kind of contact we have between the finger tip of the hand and the hand a certain component are transmitted that it will be explained in this chapter.

3.2.1.2. Hand jacobian J [18]:

The same procedure as grasp matrix applies to obtain the hand jacobian **J**:

$$\begin{pmatrix} v_{i,hand}^{N} \\ \omega_{i,hand}^{N} \end{pmatrix} = Z_i \dot{\mathbf{q}} \tag{8}$$

In **Eq. (8)**, $\boldsymbol{v}_{i,hand}^{N}$ is the translational velocity of the contact **i** which is located on the hand of the robot and $\boldsymbol{\omega}_{i,hand}^{N}$ is the angular velocity of the link that touches the $\mathbf{i^{th}}$ contact point.

It is worthwhile mentioning that the hands twist elements in **Eq. (8)** are with respect to the inertial frame $\{\mathbb{N}\}$. Either the velocity is related to velocities of the joints through matrix $Z_i \in \mathbb{R}^{6\times nq}$ that its columns are the Plücker coordinates of the axis of the joints defined in **Eq. (9)** [19].

3.2.2. Plücker coordinates:

Plücker coordinates represents an axis is defined by its direction $\hat{\ell}$ and a point p that moves along it like $(\hat{\ell}, m)$ where $m = p \times \ell$ is the moment vector. Any point like q lies in the line if $q \times \hat{\ell} = m$. For sufficiency, $(q - r) \times \hat{\ell} = m - m = 0$ where r is another point in the other point located on the same line as q does, hence, vector $q - r$ is collinear with the line ℓ [20].

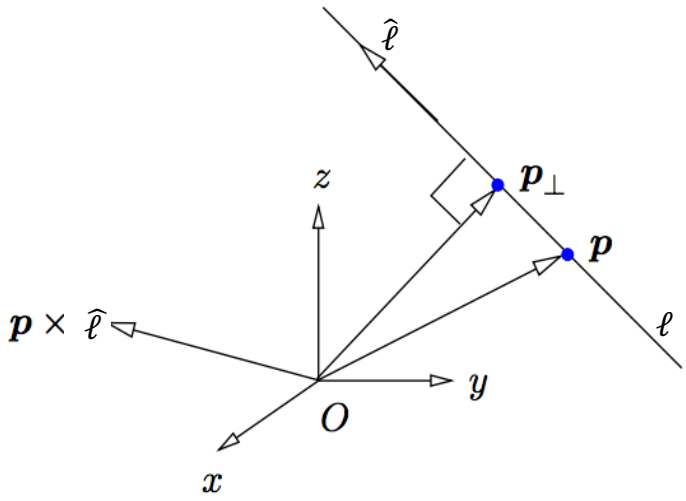

Figure 2 - Plücker coordinates ($\hat{\ell}$, $\hat{p} \times \hat{\ell}$) of a line – inspired by [20]

The Plücker coordinates and its component is shown in **Figure 2**.In Plücker coordinate, $\hat{\boldsymbol{\ell}}$ is the unit vector and $\| \boldsymbol{m} \| / \| \hat{\boldsymbol{\ell}} \|$ gives the distance of the line to the origin. It is evident if $\boldsymbol{m} = \mathbf{0}$ means that the point is located on the line that reach at its smallest value $\mathcal{P}\perp$ where [20]:

$$\mathcal{P}\perp = p - (\hat{\ell}.p)\,\hat{\ell} = (\hat{\ell}.\hat{\ell})p - (\hat{\ell}.p)\,\hat{\ell} = \hat{\ell}\times(p\times\hat{\ell}) = \hat{\ell}\times m$$

Rigid body is represented with **6** coordinates in space, however, in Plücker coordinate there are only **4** coordinates. This is because of the kind of constraints shrinks it in to 4. The first constraint is $\|\hat{\boldsymbol{\ell}}\| = \mathbf{1}$ and the second constraint $\hat{\boldsymbol{\ell}}.\boldsymbol{m} = \hat{\boldsymbol{\ell}}.\left(\boldsymbol{p}\times\hat{\boldsymbol{\ell}}\right) = \mathbf{0}$ [21].

$$Z_i = \begin{bmatrix} d_{i,1} & \dots & d_{i,n_q} \\ k_{i,1} & \dots & k_{i,n_q} \end{bmatrix} \tag{9}$$

vectors $\boldsymbol{d}_{i,j}$, $\mathcal{K}_{i,j} \in \mathbb{R}^3$ are defined in the following way [18]:

$$d_{i,j} = \begin{cases} 0_{3\times 1} & \text{if the } i^{th} \text{contact force doenst have any impact on the } j^{th} \text{joint} \\ \hat{z}_j & \text{if the j is prismatic} \\ \mathbf{S}\left(c_i - \zeta_j\right)^T \hat{z}_j & \text{if the j is revolute} \end{cases}$$

$$k_{i,j} = \begin{cases} 0_{3\times1} & \text{if the } i^{th} \text{contact force} \\ & \text{doenst have any} \\ & \text{impact on the } j^{th} \text{joint} \\ 0_{3\times1} & \text{if the j is prismatic} \\ \hat{z}_j & \text{if the j is revolute} \end{cases}$$

Based on **Denavit-Hartenberg** [22] convention, $\boldsymbol{\zeta_j}$ is chosen which is the origin of the coordinate frame related to the $\mathbf{j^{th}}$ join and $\hat{\boldsymbol{z}}_\mathbf{j}$ is the unit vector aligned with the z-axis of the mentioned frame which is the rotational axis for the revolute joint and the direction of translation for the prismatic joints. It is worthwhile mentioning that any other method can be applied instead of **Denavit-Hartenberg** [22]. However, for sake of simplicity this method is advised.

Ultimately, the hand twist in local frame at the contact points $\{\mathbb{C}_i\}$ is obtained by [18]:

$$\nu^{C_i}_{i,hand} = \bar{R}_i^T \begin{pmatrix} v^N_{i,hand} \\ \omega^N_{hand} \end{pmatrix} \tag{10}$$

Combination of **Eq. (8)** and **Eq. (10)** culminates in to the following equation for the **partial hand Jacobian:**

$$\tilde{J}_i = \bar{R}_i^T Z_i \tag{11}$$

Partial hand Jacobian matrix $\tilde{\mathbf{J}}_\mathbf{i}$ will map the joint velocities to the contact twist on the hand:

$$v^{C_i}_{i,obj} = \tilde{G}_i^T \dot{q} \tag{12}$$

All the twist in both hand and object can be compactly represented in the following way:

$$\nu^C_{hand} = \left(\nu_1^T{}_{,hand} \cdots \nu_{n_c}^T{}_{,hand}\right)^T \tag{13}$$

$$\nu^C_{obj} = \left(\nu_1^T{}_{,obj} \cdots \nu_{n_c}^T{}_{,obj}\right)^T \tag{14}$$

Now, by stacking the partial grasp matrix of every contact point, the complete grasp matrix $\tilde{\mathbf{G}} \in \mathbb{R}^{6\times 6n_q}$ and complete hand Jacobian $\tilde{\mathbf{J}} \in \mathbb{R}^{\mathbf{6n_c\times n_q}}$ are shown respectively:

$$\tilde{G} = \begin{bmatrix} \tilde{G}_1^T \\ \vdots \\ \tilde{G}_{n_c}^T \end{bmatrix} \tag{15}$$

$$\tilde{J} = \begin{bmatrix} \tilde{J}_1^T \\ \vdots \\ \tilde{J}_{n_c}^T \end{bmatrix} \tag{16}$$

consequently:

$$\nu_{c,obj}^{C} = \widetilde{G}^{T}\nu \tag{17}$$

$$\nu_{c,hand}^{C} = \tilde{J}^{T}\dot{q} \tag{18}$$

It is of great importance to mention that in either the complete grasp matrix $\widetilde{\mathbf{G}}$ and complete hand Jacobian matrix $\tilde{\mathbf{J}}$, all the $\mathbf{6n_c}$ components are mapped. However, based on the type of contact between the hand and the object, due to constraints of the contact, only certain components of twist and/or wrench are transmitted. That will be detailed later.

3.2.3. Contact modelling [23]:

A contact model defines the wrenches transmitted between the contacts as well as the relative motion between the two engaging objects which are the hand and the object to be manipulated. Both the geometry and the material of the contacts are influence the mentioned interaction. Based on the application, the model that is used can be vary from rigid-body model to compliant model.

3.2.3.1. Compliant modelling of the contacts:

Unlike the rigid-body model, the contact forces are driven from compliant or stiffness of the object involved. Notwithstanding being more complicate with respect to its rigid-body counterpart, compliant models overcome the statically indeterminable problem of the rigid-body model.

To reduce the complex nature of this modulation, a reduction in the order of compliance is required that has a limited number of variables [23].

3.2.3.2. Rigid-body modelling of the contacts [18]:

In this way of modelling, no deformations are allowed at the contact points. The contact forces arise from, first, the constraints regarding the incompressibility and impenetrability of the contacting objects and, second, surface frictional forces.

Although computationally efficient and good to describe the qualitative issues, for instance graspability[1] of the fixture, rigid-body modulation is not capable of describing the full range contact phenomena. As a case in point, rigid-body modulation is incapable of predict the contact forces individually in multi-contact fixture (static indeterminacy problem) [24]. Apart from the mentioned problem, in realistic situation deformation is probable that makes rigid-body less accurate. Lastly, Coulomb friction

[1] This characteristic of the grasp will be explained in detail

model that comes with the rigid-body modulation can render mechanic problem that either has no solution or many possible solution [25].

There exist **3** main different class of grasp based on **rigid-body** modulation that includes **point-contact-without-friction(PwoF), hard finger (SF)** and **soft-finger (SF)** [26]. These models define which wrenches or twist can be transmitted to the contact point either from either hand that is done by equating a subset of the object or the hand twist at each contact point.

In application of the rigid-body modulation should take caution to the large deformation in either finger or the object as it may render inaccuracies.

3.2.3.2.1. point-contact-without-friction (PwoF) [18]:

In this model, contact patch is very small and the surface between the hand and object is slippery.

Based on what is mentioned, according to the virtual work principle, normal component of the translational velocity $\boldsymbol{v}$ is transmitted to the contact point. The same applies for the wrenches.

3.2.3.2.2. Hard-finger (HF) [18]:

In presence of an appreciable contact-friction, there are 3 translational elements of twist are transmitted (i.e. $\boldsymbol{v}$) to the object. However, the patch is still too small to have considerable enough amount of friction moment. Consequently, there will be no angular component of the twist is transmitted. In this case HF is applicable. It is worthwhile mentioning that the same applies to the wrench of the object

3.2.3.2.3. Soft-finger (SF) [18]:

In presence of frictions and large enough contact patch, there exist both friction forces and friction moments. Similar to the hard finger case, 3 translational velocity $\boldsymbol{v}$ or force $\boldsymbol{f}$ components are transmitted, however, angular friction moment imposes an angular velocity or moment around the contact normal.

For all the above-mentioned models for contacts, relative twist between the contact **i** as follows:

$$[\tilde{J}_i \quad -\tilde{G}_i]\begin{bmatrix}\dot{q}\\ \nu\end{bmatrix} = \nu_{i,hand}^{c_i} - \nu_{i,obj}^{c_i} \tag{19}$$

Through a matrix known as selection matrix $\mathbf{H_i} \in \mathbb{R}^{\mathbf{n_{\lambda_i}}\times\mathbf{6}}$, a peculiar model is defined that picks $\mathbf{n_{\lambda_i}}$ components, known to be the **transmitted degree of freedom,** of relative twist and set them to zero [18]**:**

$$\mathrm{H_i}\left(\nu_{\mathrm{i,hand}}^{\mathrm{c_i}} - \nu_{\mathrm{i,obj}}^{\mathrm{c_i}}\right) = 0 \tag{20}$$

where*:*

$$H_i = \begin{bmatrix} H_{iF} & 0 \\ 0 & H_{iM} \end{bmatrix} \tag{21}$$

In **matrix 21**, $\mathbf{H_{iF}}$ and $\mathbf{H_{iM}}$ are the translational and rotational selection matrix respectively. The values of the selection matrixes are based on the selected model.

After choosing the appropriate contact model for each contact, there will be the following equations for the **kinematic contact constraint,** like **Eq. (20)**, for all the contact $\mathbf{n_c}$ in the compact way:

$$H\left(v^c_{c,hand} - v^c_{c,obj}\right) = 0 \tag{22}$$

here*:*

$$H = \begin{bmatrix} H_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & H_{n_c} \end{bmatrix} \in \mathbb{R}^{n_\lambda \times 6n_c}$$

It is worth noting that **kinematic contact constraint (Eq. (22))** holds only if contact forces satisfies the friction constraints.

The total number of $\mathbf{n_\lambda}$ that is transmitted through total number of contact $\mathbf{n_c}$ is given by:

$$n_\lambda = \sum_{i=1}^{n_c} n_{\lambda_i} \tag{23}$$

Table 2 - Selection matrix for three types of contact model – Adapted from [18]

Model	$\mathbf{n_{\lambda_i}}$	H_{iF}	H_{iM}	λ_i
PwoF	1	[1 0 0]	Void	$[f_{in}]$
HF	3	$I_{3\times3}$[1]	Void	$[f_{in}\ f_{it}\ f_{i0}]$
SF	4	$I_{3\times3}$	[1 0 0]	$[f_{in}\ f_{it}\ f_{i0}\ m_{in}]$

By substituting the **Eq. (17)** and **(18)** into **Eq. (22),** we have the following:

[1] unity matrix with 3 rows and column

$$[J \quad -G^T]\begin{bmatrix}\dot{q}\\ \nu\end{bmatrix} = 0 \tag{24}$$

$$G^T\nu = \nu_{cc,hand} = \nu_{cc,obj} = J \tag{25}$$

In **Eq. (25)**, $\boldsymbol{\nu_{cc,hand}}$ and $\boldsymbol{\nu_{cc,obj}}$ contains solely the components of the twist that are transmitted through the contacts.

Eventually, the transpose of grasp matrix and hand Jacobian ($\mathbf{G^T}$and $\mathbf{J}$ in the same order) can be defined:

$$G^T = H\tilde{G}^T \in \mathbb{R}^{n_\lambda \times 6} \tag{26}$$

$$J = H\tilde{J} \in \mathbb{R}^{n_\lambda \times n_q} \tag{27}$$

If the condition that underlined in **Eq. (25)** is satisfied for all the course of time is defined as **grasp maintenance** [18].

3.2.4. Dynamic equilibrium [18]:

Dynamic equation of the system can be obtained by Lagrange-Euler equation or any other approach in the following way that are confined by kinematic equations mentioned in **Eq. (24)**. With application of the kinematic constraints.in the meanwhile, as it is previously mentioned, quasi-static assumption is applied that makes the inertia terms in **Eq. (28)** be zero leading to **Eq. (29)**. It is apparent, that **Eq. (29)** of the dynamic equation is closely related to that of (kinematic equation)

$$\begin{cases} M_{hand}(q)\ddot{q} + b_{hand}(q,\dot{q}) + J^T\lambda = \tau_{app} \\ M_{obj}(u)\dot{\nu} + b_{obj}(u,\nu) - G\lambda = g_{app} \end{cases} \tag{28}$$

The inertia matrix $M_{hand}(.)$ and $M_{obj}(.)$ are symmetric and positive definite[1], b_{hand}, b_{obj} are velocity-product terms like Coriolis effect, g_{app} and τ_{app} are the external wrench including the gravity and the vector of external loads and actuators action. Finally, $\mathbf{G\lambda}$ is the vector of total wrench applied to the object by hand. As it is formerly mentioned, λ_s are the contact forces and moments that are transmitted to the object in the following manner:

$$\lambda = \left[\lambda_1^T \dots \lambda_{n_c}^T\right]^T$$

[1] $q^T M_{n\times n} q > 0$ for all $q \in \mathbb{R}^n$

where:

$$\lambda_i = H_i[f_{in}\ f_{it}\ f_{io}\ m_{in}\ m_{it}\ m_{io}]^T$$

Based on the type of contact that has been defined, the conveyed contact force can be defined at the fourth column of Table 2.

$$\begin{bmatrix} J^T \\ -G \end{bmatrix}[\lambda] = \begin{bmatrix} \tau \\ g \end{bmatrix} \tag{29}$$

where:

$$\begin{cases} \tau = \tau_{app} - M_{hand}(q)\ddot{q} - b_{hand}(q, \dot{q}) \\ g = g_{app} - M_{obj}(u)\dot{v} - b_{obj}(u, v) \end{cases} \tag{30}$$

Eq. (29) gives another view of the grasp matrix **G** and hand Jacobian **J**. In this case, G can be considered as the transmitted contact wrenches to the one that can be applied by the joints of the hand, whilst $\boldsymbol{J^T}$ is transpose of hand Jacobian that maps the contact wrenches to the vector of the joint loads τ [18].

The dynamic equation in **Eq. (29)** is clearly closely to the velocity kinematics in **Eq. (24)**. Thanks to the quasi-static assumption that has been made, the inertial terms Mhand(.)., Mobj(.) can be omitted, which makes it even closer to that of kinematic equation. Consequently, we the following equation can be obtained:

$$\begin{cases} \tau = \tau_{app} \\ g = g_{app} \end{cases} \tag{31}$$

As it is clearly seen in **Eq. (31)**, the dynamic equation is not dependent on the joints and object velocities, hence in quasi static grasp, kinematic equation, **Eq. (17)**, and the dynamic equation, **Eq. (31)**, have duality. In another those equations can be solved independently to acquire **λ, $\dot{\mathbf{q}}$** and **ν**. It is of great importance to note that the duality is applied only in presence of slow evolution of the system states otherwise, if the inertia is considerable, there will be no possibility to have it.

3.2.5. Controllable Twists and wrenches:

Regarding the planning of the grasp, it is very important to be aware of the kind of twists that are led from a set of finger and links of the grasp in general and, reversely, the condition under which hand can block all the possibilities of the objects to move. The dual view is to know the set of wrenches that fingers along with the criteria under which the mentioned wrenches transmitted to the object at the contact points [18]. For this purpose, subspaces of hand Jacobian **J**, grasp matrix **G** and other associated matrixes (i.e. transposes of the mentioned matrixes $\mathbf{J^T}$and $\mathbf{G^T}$) need to be studied [27] that will be explained in detail later in this chapter. The mentioned spaces are the null space and range space ($\mathcal{N}(.)$ and $\mathcal{R}(.)$ in the same order) of both hand Jacobian **J** and

grasp matrix $\mathbf{G}$ and their transposes ($\mathbf{J^T}$and $\mathbf{G^T}$). It is necessary to have a brief introduction on range space and null space in linear algebra.

3.2.5.1. Null space $\mathcal{N}(.)$ and range space $\mathcal{R}(.)$ in linear algebra:

Also known as Kernel, Null space a linear map represented as a matrix $A_{m \times n}$ with coefficients in a field K and operating on column vectors x with n components over K is the set of solutions to the equation A x = 0, where 0 is understood as the zero vector. The dimension of the kernel of A is called the nullity of A [28].

we have:

$$\mathcal{N}(A) = \text{Null}(A) = \text{Ker}(A) = \{x \in K^n | \, Ax = 0 \}$$

in which:

$$Ax = 0 \Leftrightarrow \begin{cases} a_{11}x_1 + \cdots + a_{1n}x_n = 0 \\ \vdots \\ a_{m1}x_1 + \cdots + a_{mn}x_n = 0 \end{cases}$$

The null space of $\mathbf{A}$ is the solution of the above-mentioned system of linear equation.

In linear algebra, the column space (also called the range or image) of a matrix $\mathbf{A}$ is the span (set of all possible linear combinations) of its column vectors. The column space of a matrix is the image or range of the corresponding matrix transformation. Let K be a field of scalars. Let $\mathbf{A}_{m \times n}$ be a matrix, with column vectors v_1, v_2, ..., v_n. a linear combination of these vectors is any vector of the following form: $\mathbf{c_1 v1} + \cdots \mathbf{c_n v_n}$
Any set of linear combinations of column vector of $\mathbf{A}$ can be written in the following way as a product of A with a column vector [29]:

$$A\begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}\begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} = c_1\begin{bmatrix} a_{11} \\ \vdots \\ a_{m1} \end{bmatrix} + \cdots + c_n\begin{bmatrix} a_{1n} \\ \vdots \\ a_{mn} \end{bmatrix} = \mathbf{c_1 v1} + \cdots \mathbf{c_n v_n}$$

Figure 3 illustrate the subspaces of hand Jacobian and grasp matrix. As it is clearly seen, general velocity $\dot{q} \in \mathbb{R}^{n_q}$ can be decomposed into two orthogonal vectors ($\mathcal{R}(J^T)$ and $\mathcal{N}(J)$) on the left and on the right side, there seen, contact twists $\mathbf{vcc} \in \mathbb{R}^{\mathbf{n_\lambda}}$ can be obtained by multiplication of range space of grasp matrix $\boldsymbol{\mathcal{R}(G)}$ into the pseudoinverse of it (i.e. $\mathbf{G^+}$). Same applies for the wrenches.

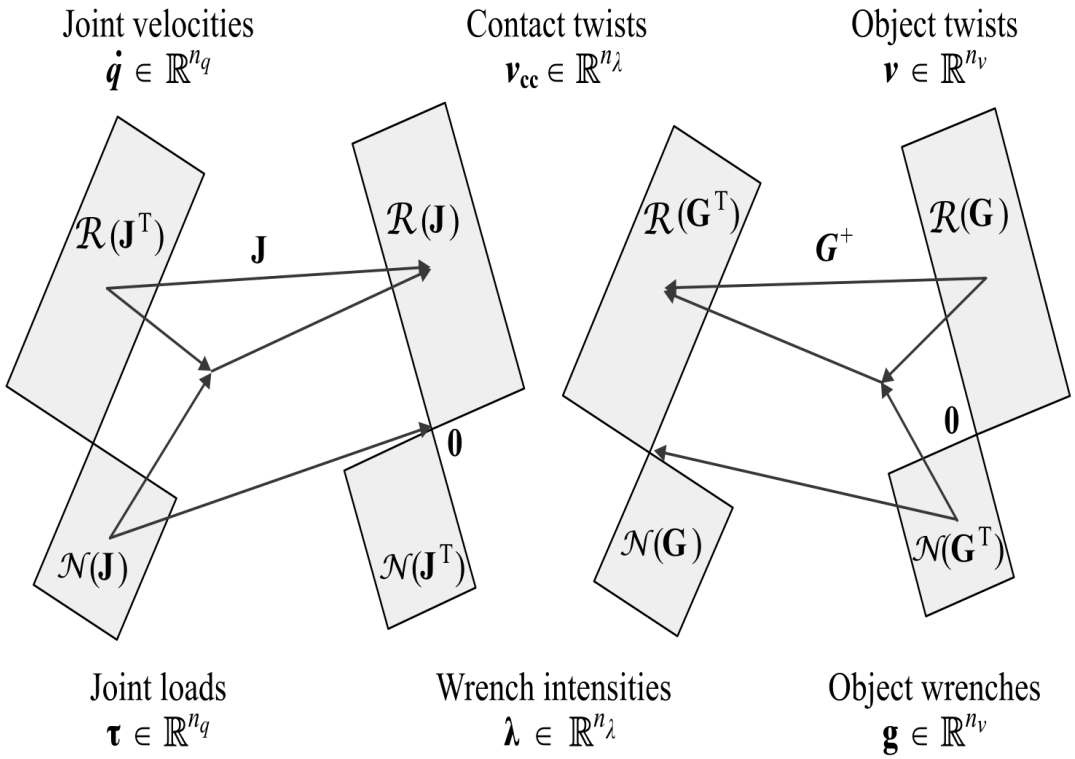

Figure 3 Linear maps of grasping system twists and wrenches - Adapted from [18]

3.2.6. Grasp classification:

In Figure 3, there are four null spaces that drive the different classes of grasp. It is assumed that the **dynamic equation** of the system (**Eq. (29)**) at least has a solution. The following equations gives a better understanding of underlying physiques behind the mentioned subspaces of G, G^T, J and J^T [18].

$$\dot{q} = J^{+1}\nu_{cc} + \mathcal{N}(J)\eta \tag{32}$$

$$\nu = (G^T)^+\nu_{cc} + \mathcal{N}(G^T)\eta \tag{33}$$

$$\lambda = -G^+ g + \mathcal{N}(G)\eta \tag{34}$$

$$\lambda = (J^T)^+\tau + \mathcal{N}(J^T)\eta \tag{35}$$

Table 3 introduces the basic classes of grasping based on their null space. The null spaces, there are many contribution of on vector space to only vector space.

[1] Here, + denotes the generalized inverse of a matrix which is not usually square. It is also known as pseudo-inverse of a matrix

Table 3 - basic classes of the grasping- inspired by [18]

Condition	*Classification*	*Many-to-one*
$\mathcal{N}(J)\neq 0$	*Redundant*	$\dot{q} \rightarrow \nu_{cc} \& \tau \rightarrow \lambda$
$\mathcal{N}(G^T)\neq 0$	*Indeterminate*	$\nu \rightarrow \nu_{cc}$ & $g \rightarrow \lambda$
$\mathcal{N}(G)\neq 0$	*Graspable*	$\lambda \rightarrow g$ & $\nu_{cc} \rightarrow \nu$
$\mathcal{N}(J^T) \neq 0$	Defective	$\lambda \rightarrow \tau \& \nu_{cc} \rightarrow \dot{q}$

3.2.6.1. Redundant grasp

The grasp is said to be redundant if $\mathcal{N}(J)$ is non-trivial [18]. In this situation, there is at least a set of joint velocities of the hand $\dot{q}$ that has no contribution to the objects movements. $\dot{q}$ in $\mathcal{N}(J)$ are known as internal velocities as there will be no movement in the fingers tips. In quasi-static motion, internal motions have no impact on the objects movement and vice versa.

3.2.6.2. Indeterminate grasp

If the null space of the transpose of the grasp matrix $\mathcal{N}(G^T)$ is non-trivial, the grasp is indeterminate. In this case, the twists of the object ν cannot makes the object move in the direction of the constraints the movements are defined. The mentioned twist in $\mathcal{N}(G^T)$ is known as internal twist of the object [18].

3.2.6.3. Defective grasp

There are found some contact forces belonging to $\mathcal{N}(J^T)$ that have no influence on the load applied to the contact points based on dynamic equation given by **Eq. (28).** In this case, null space of the hand Jacobian's matrix is non-trivial [18].

3.2.6.4. Graspable grasp

Wrench intensities λ that belongs to the null space of the grasp matrix $\mathcal{N}(G)$ are internal object forces as they have zero contribution in objects movement. It is worthwhile mentioning that these internal forces be the squeezing forces makes the overall forces stays in friction cone. In other words, the normal forces to the object surface are seen to belong to the $\mathcal{N}(G)$. In force closure grasps and Soft-finger and hard finger contacts where friction plays a crucial role, graspability seen to be a desirable criterion that guarantee the object be maintained in the grasp [18]. To have a graspable grasp, null space of grasp $\mathcal{N}(G)$ should be non-trivial.

3.3. Methodology

3.3.1. Introduction:

To simulate and model the grasp, contact information that exchanges between the hand and an object, like twists and wrenches, is obtained by hand Jacobian and Grasp matrix (J and G respectively). The Grasp matrix is formed from the contact points plus a point at the object's **COM.** Depending on the nature of the contacts, some constraints are imposed including the non-penetration conditions, contact maintenance[1] and friction cone constraints. Based on the scenario under consideration that can be toppling or wall-grasp, another set of constraints acting on contact velocities and forces are added that govern the permissible movement of the objects. In short, using the grasp matrix **G,** the following constraints can be included in a linear optimization:

- General constraints
  - Friction cone constraints
  - Contact maintenance
  - Non-penetration condition
- Task constraints

Some contacts must slip and some must be maintained. These are included in the task constraints, while the general task that is applicable to all the situations and scenarios are listed in general constraints. The full description of the constraints will be explained in greater detail later in this chapter. To include the gravity, include an imaginary finger at the center of the mass **COM** pointing the direction of the gravity imposing a force equivalent to the mass multiplied by the acceleration of the gravity. The gravity effect is fed to the grasp matrix by a zero matrix $\mathbb{R}^{2(n_c+1)\times 2}$**.** The mentioned finger should be added to form the grasp matrix by a null matrix $\mathbb{R}^{2(n_c+1)\times 2}$ **where** n_c denotes the number of contact points**.**

Finally, based on the permissible movement of the object in different scenarios, the optimization renders the optimal solution for the contact forces on using the minimization of the cost function which is **Peshkin's minimal energy** that is the multiplication of the external forces acting on the i^{th} contact point to its velocity. This cost function is especially tailored for the quasi-static cases [1].

3.3.2. Scenarios:

The main concept is not to avoid the obstacles in robot's vicinity, but exploit them to replicate a human arm which is the best manipulator. Analogous to human, there are

[1] The two bodies maintain their contacts over the course of time

three different maneuvers that a robotic arm can perform that are different in general but share some properties:

- *Toppling*
- *Wall-grasp*
- *Edge-grasp*

The toppling and wall-grasp approaches are studied in this thesis. More information on the 3rd approach is provided by **G. Ardakani et al. (2018)** [14].

3.3.2.1. Toppling:

During toppling, the robot aims to reorient the object by imposing force(s) on the upper edge(s) of the object making it rotate around a pivot point/edge. This makes it easy for the robotic hand to have a secure grip. For toppling, the force that is intended to perform it must point outside of the edge that is leaning against the table or ground. As it is depicted in Figure 4, $\mathbf{F_1}$ and F_2 can rotate the object while $\mathbf{F_3}$ squeezes it. Further studies on the graphical studies of grasping and manipulation can be found in Trinke and Paul (1990) [3]. Besides, the applied force on the object should impose a twist at its **Centre of Mass (COM)** such that the object rotates. The constraints related to the friction cone and pivot point must also be satisfied.

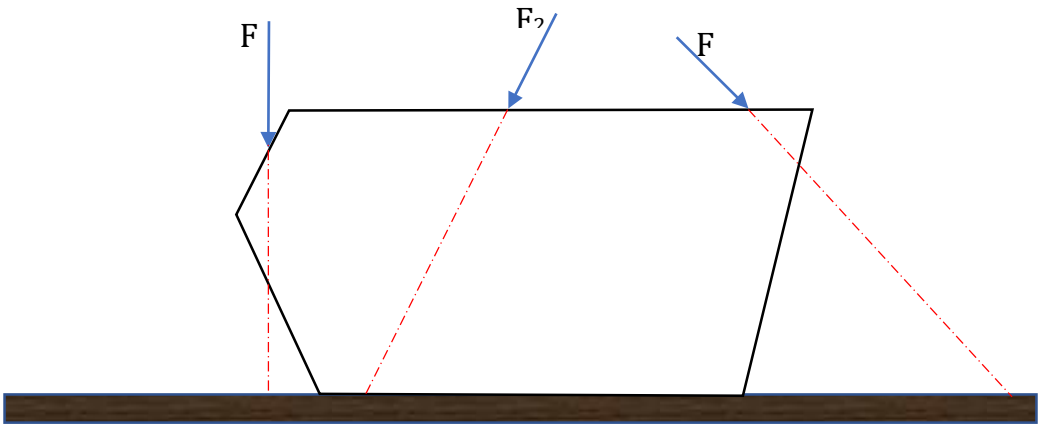

Figure 4 - Geometrical representation of position where the force can flip the object up

The total wrench that the hand causes at the **COM** can be obtained in the local frame:

$$g = G\lambda \tag{36}$$

Under the assumption of a quasi-static condition and the grasp is studied in two dimensions (2-D), the forces at the **COM** is aligned with the same direction as the object velocity v_{obj}, hence:

$$v_{obj} = \alpha g \tag{37}$$

where **α** is the scaling factor. With substitution of **Eq. (36)** into **Eq. (37)**, the mentioned wrench **g** at the contact points is**:**

$$\nu_{obj} = \alpha G \lambda \tag{38}$$

To impose the above-mentioned constraint, the velocities of the contact points are needed. As it is previously mentioned, $\mathbf{G^T}$ maps the object wrenches and twists of the object at the **COM** into the local contact frame **{C_i}**:

$$\nu_{cc} = \alpha G^T G \lambda \tag{39}$$

Eq. (39) is in the local reference frame at the contact point. To bring it to the global frame **{N},** the rotation matrix $\overline{\mathbf{R}}$ is multiplied to **Eq. (39)**:

$$\nu_{cc}^{\{N\}} = \alpha \overline{R} G^T G \lambda \tag{40}$$

Constraints

General constraints

To formulate the problem, it is necessary that the contact forces lie in for sliding at the edge of the friction cone. The friction equation for each contact point follows **Eq. (41)**:

$$\begin{cases} -\mu_i \lambda_{i,n} \leq \lambda_{i,t} \\ \lambda_{i,t} \leq \mu_i \lambda_{i,n} \end{cases} \quad i = 1, \dots, n_c \tag{41}$$

where $\mathbf{n_c}$ is the number of fingers that includes the imaginary finger for the gravity.

To maintain the contact, the normal force on the contact point should be positive, **Eq. (42)**, that should be taken in to account.

$$\lambda_{i,n} > 0 \tag{42}$$

Eq. (41) and **(42)** in matrix form has the following structure. It is worthwhile mentioning that the corresponding value for the gravity is included as well by two zero column vector at the end of the mentioned matrix based on where one defines the gravity as a finger, for convenience, like what has been done in this thesis, the last finger is accounted for the gravity effect). In the **Eq. (39)**, it is illustrated in red.

$$Gen_{const,i} = \begin{bmatrix} -\mu_i & 1 \\ -\mu_i & -1 \\ -1 & 0 \end{bmatrix} \quad , i = 1, \dots, n_c \tag{43}$$

$$Gen_{const} = \begin{bmatrix} Gen_{const,1} & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & Gen_{const,nc} & 0 & 0 \end{bmatrix}_{3n_c \times (n_\lambda + 2)} \tag{44}$$

Non-penetrating constraints are assigned to confine the object in accordance with the environment it is located. In other words, it prevents the object to go inside the other objects as the bodies (object and its surroundings like table) are considered rigid. To take the non-penetrating constraints into account, the normal components of the velocity at the contact points on the object and the constraints should be non-negative (**Eq. (45)**). It is worthwhile mentioning that, the contacts velocities here are measured with respect to the local frame **{C_i}.** As it is clearly seen, it is the same as **Eq. (39)** just for the contact points of the object to the environment.

$$\nu_{cc} = \alpha G^T G\lambda \;\; \forall\, i \in i_{oc} \tag{45}$$

where i_{oc} is the subset of the contact points that are in contact with the obstacle including the wall and the table.

Task constraints

In toppling, based on the task, contact point on one end should not move in any direction. As a result, **Eq. (43)** is applied:

$$v^{\{N\}}{}_{cc,p} = 0 \tag{46}$$

In view of **Eq. (40)**, **Eq. (46)** means the velocity of the contact point at the pivot point should be identically zero or at least very small. Here $\boldsymbol{v}^{\{N\}}{}_{\boldsymbol{cc,p}}$ is the contact velocity at the pivot point (comprising both normal and tangential component) where the object rotates around in the global frame **(Figure 5).**

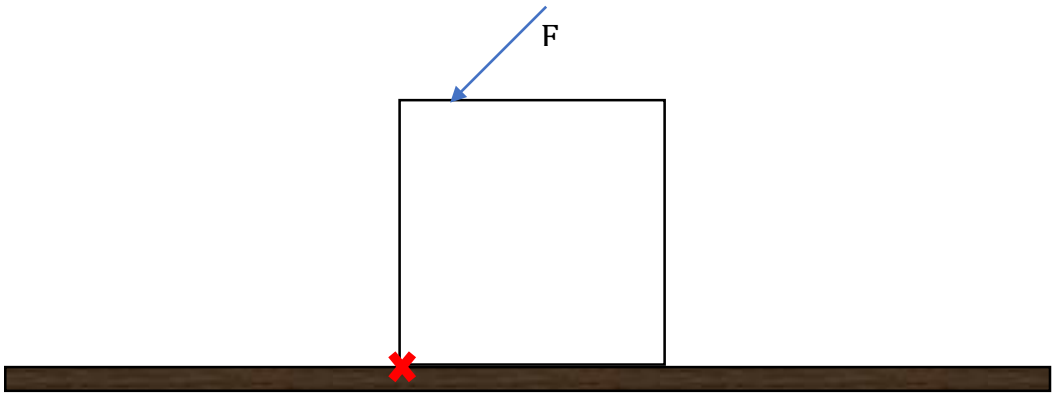

Figure 5 Toppling scenario: the pivot point is highlighted by a red marker

3.3.2.2. Wall-grasp:

Similar to the toppling scenario, wall-grasp strategy aims to change the pose of the object in order to have a better and more secure grip. In this scenario, the object is pushed all the way to a wall-like obstacle. Based on the force that is applied to the object and its reactions from the constraints, it can slide along the obstacle's surface or pivot around the contact point on to the wall.

Apart from the magnitude of the wrench at the contact point, it is equally important to consider both the direction motion and the position of the finger on the object. Like what is mentioned in **toppling scenario,** once the first finger is set, the objects perimeter can be divided to sliding, lifting and squeeze points. For more information about the graphical study of the grasp, see Trinke and Paul (1990) [3]. Instead, a mathematical modeling approach is taken in this thesis.

Constraints

General constraints

General constraints are almost the same as that of **toppling (Eq. (43) to (45)).** However, there are differences in small details like the number of contact points between the object and the environment. Task constraints, on the other hand, are completely different.

Task constraints

Based on the permissible movements two movement can be considered for the object. Either the upper contact point on the wall slides down toward the table or it pivots around it. In this thesis, the former scenario is studied where in addition a neighboring vertex on the ground slides away from the wall **(Figure 6)**. To include the mentioned constraints into a solver, the normal component of the mentioned contact points should be both zero with respect to the global frame **{N}** and the tangential velocity for the one leaning on the wall should be positive with respect to the local frame **{C_i}** and negative for the contact point on the table with respect to the same reference frame.

$$v_{cc,n_{icon}}^{\{N\}} = 0 \tag{47}$$

where $\boldsymbol{n_{icon}}$ is the normal component of the i^{th} contact point to the wall and table.

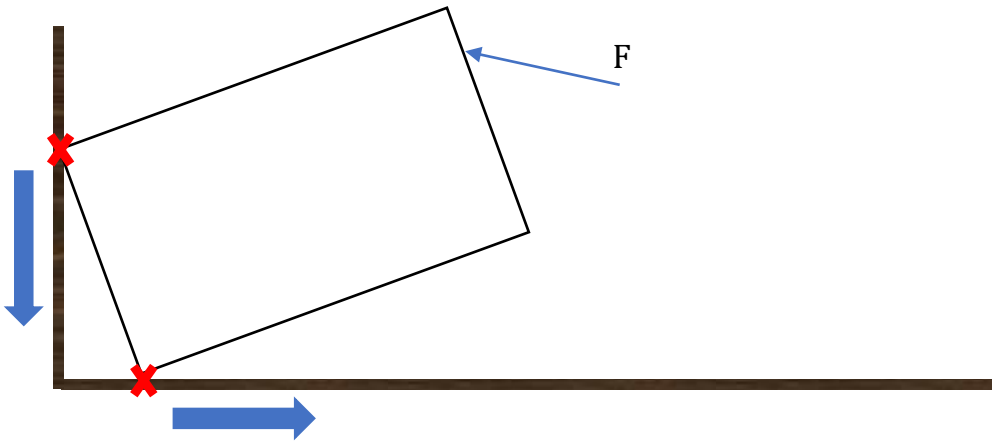

Figure 6 - Wall-grasp permissible movement

3.3.2.3. Optimization

By approximation on the friction cone modeling and quasi-static assumption, the nonlinearities are reduced and what is left is a linear problem that can be solved by linear programming. In this case, linear optimizers in **Matlab®** like **linprog** can be utilized.

An optimizer, based on the constraint that has been defined and the cost function that has been assigned, returns a set of contact forces making the object move in the permissible move set by minimizing the cost function that has been assigned which is Peshkin's minimum power energy [1].

According to Peshkin's principle [1], at each sampling time system picks the easiest motion regarding the energy it spends as it confronts the obstacles. This principle is applicable only to quasi-static motion that is subjected to normal forces acting as confiding forces, Coulomb friction forces and the forces that are independent of velocity. The power is defined as:

$$P = \sum_{i} f_{exi} v_i \tag{48}$$

where Pdenotes the power, $\mathbf{f_{exi}}$ is the sum of all the external forces excluding the normal forces, and $\mathbf{v_i}$ is the velocity of the $\mathbf{i^{th}}$ point where the external forces are applied.

4. Results and examples

4.1. Introduction

In this chapter, the simulation and the experimental results of either scenarios, including wall-grasp and toppling, are presented and studied. The values used for weight and friction coefficients correspond to what is dealt by the robot (KUKA® iiwa **LBR No. 7** light robotic arm). The contact forces obtained by the model in simulation should be fed to the robot to be validated the model. The contact forces that are applied to the object cab be monitored by a very sensitive force-torque sensor that has been mounted on the robot's end-effector.

In this chapter, it is also explained how to obtain the friction coefficients which comparing from two different approaches. Friction coefficient between the object (made of wood) and the obstacle (both wall and table made of MDF[1]) is obtained by a ramp and the one related to the probe mounted on the robot which is made of condense latex and object is obtained by the force-torque sensor.

4.2. Data and measurements

Here are the data that robot is confronted during the experiments and measurement that has been applied in the simulation.

4.2.1. Friction coefficient

For the friction coefficient between the object and its surrounding, a ramp structure is used Figure 7. In this approach, the slope is gradually increased until the object initiate sliding. The corresponding angle gives the static friction coefficient.

[1] Medium-Density Fiberboard

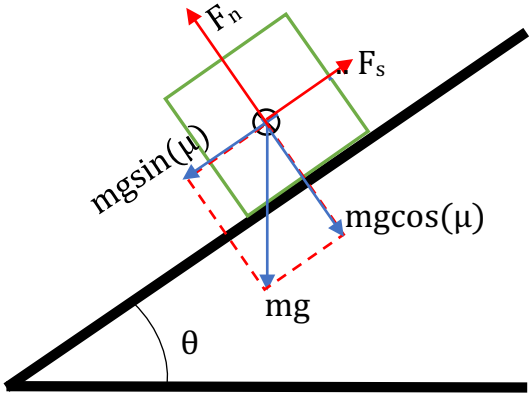

Figure 7 Ramp structure to calculate static friction coefficient

Here is the calculation regard to it:

$$F_s = mgsin(\theta)$$

$$F_n = mgcos(\theta) \implies \mu_s = {mgsin(\theta)}/{mgcos(\theta)} = {sin(\theta)}/{cos(\theta)} = tan(\theta)$$

$$\mu_s = {F_s}/{F_n}$$

Based on the observation, the object start sliding at angle around **18°**. According the previously mentioned calculation static friction coefficient would be:

$$\mu_s = tan(18^\circ) \cong 0.325$$

Regarding the static friction coefficient between the probe that is made of dense latex and the wooden object an experiment with the robot itself is used. Thanks to force-torque sensor, the force that has been measured in three axes x, y and z are measured. The ratio of their mean value (i.e. z and y direction that's stand for normal and tangential measured forces in the same order) render the static friction coefficient μ_s.

In **Figure 8**, the position of the probe with respect to the base is illustrated in the upper section and the bottom section depicts the forces in all the three axils (i.e. **x, y** and **z**). The mean value of each measured forces is shown by $\bar{x}, \bar{y}$ *and* $\bar{y}$ in the same order. The dashed lines stand for the average value of the measured forces ($|\bar{z}|$ and $|\bar{y}|$).

The friction coefficient is:

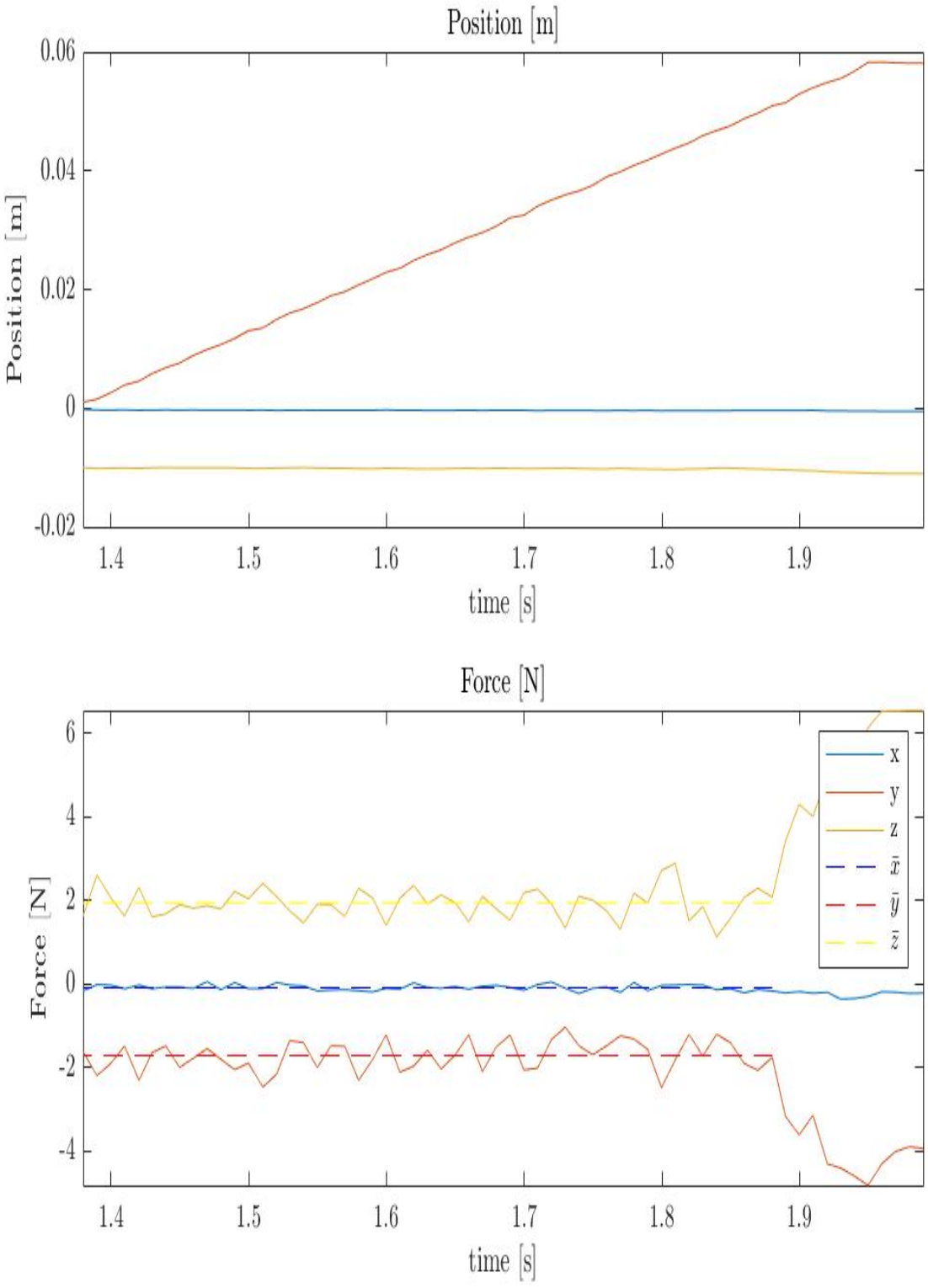

Figure 8 - position and force measured by force-torque sensor to obtain static friction coefficient μ_s

In short, the table below illustrates the values that has been used in the experiments:

Table 4 - The values of the parameter in the experiments

	Toppling		***Wall-grasp***	
Weight [g]	*495*		*700*	
Dimensions [cm] in 2-D	*6.7× 6.7*		*14.5× 4.8*	
Static friction coefficient μ_s	*Object/object*	*0.325*	*Object/object*	*0.325*
	Object/hand	0.8812	*Object/hand*	0.8812

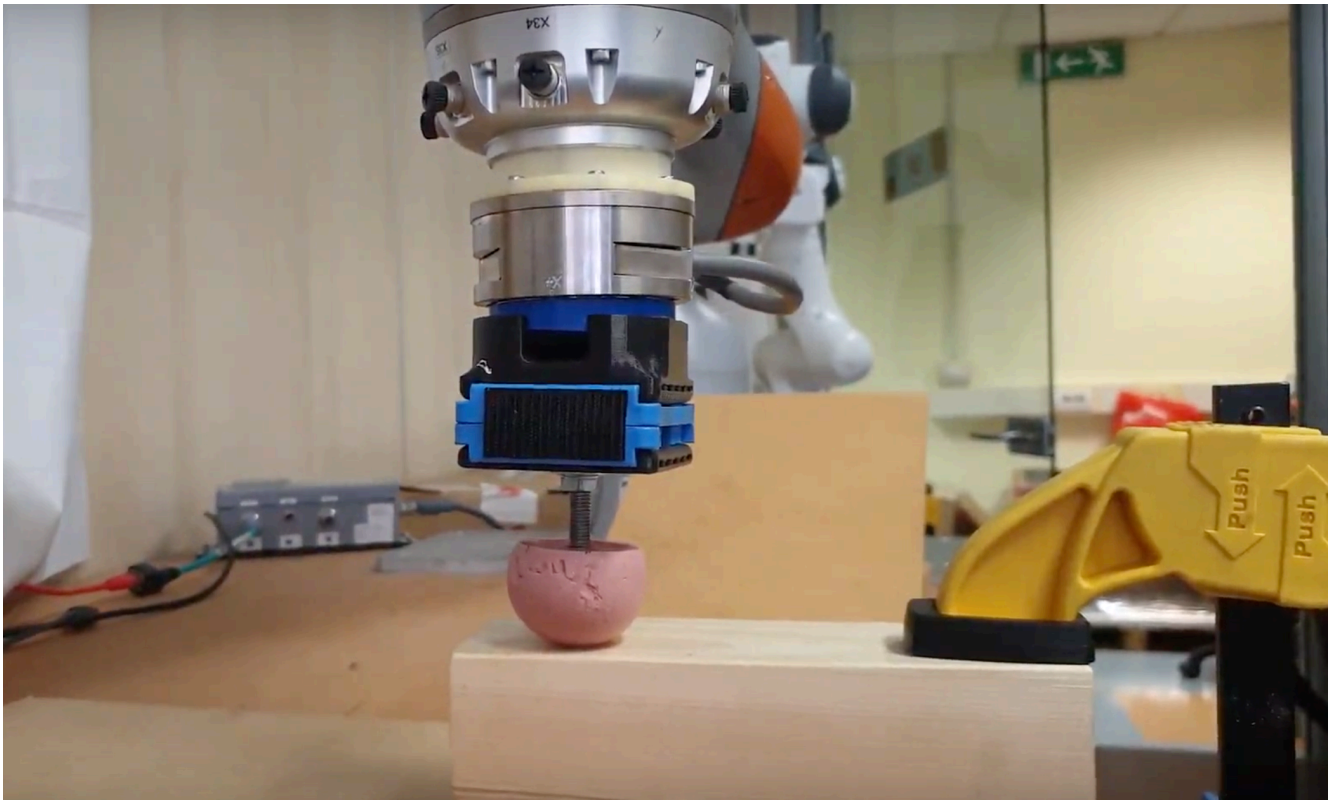

Figure 9 - Obtaining the friction coefficient between the probe and the object

4.3. Simulation and experimental results

The modulation based on what is explained previously. Here, there are several simulations based on the location of the fingertip (probe).

4.3.1. Toppling:

Based on what is experienced, **Figure** 10 *shows the sequence of the contact points in the* ***Matlab®*** *code. Take note to the fact that the gravity is imaginarily taken in to account as a finger acting on the* ***COM*** *(illustrated by a blue cross). The sequence is completely arbitrary chosen; however, it must be in correspondent with what is implemented in code.*

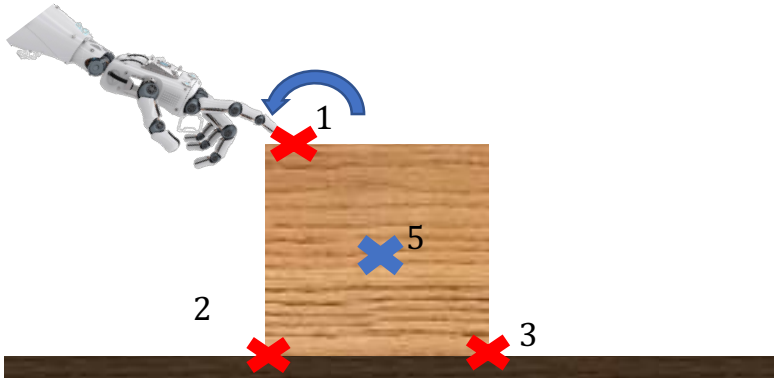

Figure 10 - Contact point location in toppling

4.3.1.1. P_1[1] = (-6, 6.7):

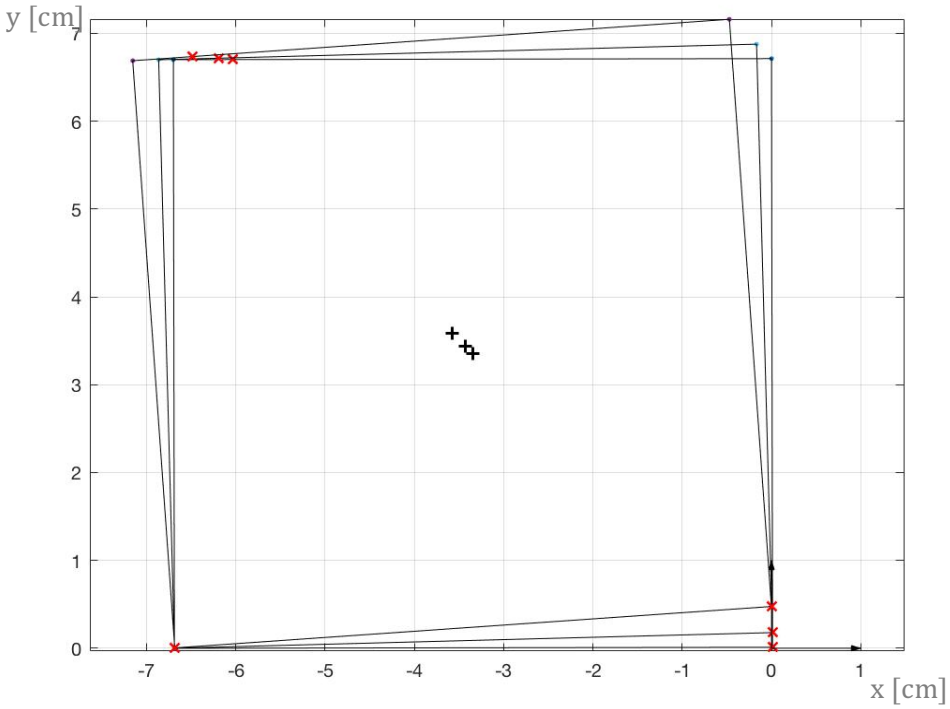

Figure 11 - Toppling P_1 = (-6, 6.7) [cm]

[1] P_i is the location of the i^{th} contact point with respect to global reference frame located (0,0)

$$G_1 = \begin{bmatrix} 0.0939 & 0.9956 & -0.0939 & -0.9956 & -0.0939 & -0.9956 & 1 & 0 \\ -0.9956 & 0.0939 & 0.9956 & -0.0939 & 0.9956 & -0.0939 & 0 & 1 \\ 2.6800 & -3.3500 & -3.3500 & -3.3500 & 3.3500 & -3.3500 & 0 & 0 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} 0.0318 & 0.9995 & -0.0318 & -0.9995 & -0.0318 & -0.9995 & 1 & 0 \\ -0.9995 & 0.0318 & 0.9995 & -0.0318 & 0.9995 & -0.0318 & 0 & 1 \\ 2.6800 & -3.3500 & -3.3500 & -3.3500 & 3.3500 & -3.3500 & 0 & 0 \end{bmatrix}$$

$$G_3 = \begin{bmatrix} 0.0939 & 0.9956 & -0.0939 & -0.9956 & -0.0939 & -0.9956 & 1 & 0 \\ -0.9956 & 0.0939 & 0.9956 & -0.0939 & 0.9956 & -0.0939 & 0 & 1 \\ 2.6800 & -3.3500 & -3.3500 & -3.3500 & 3.3500 & -3.3500 & 0 & 0 \end{bmatrix}$$

$$\mathcal{N}(G_1) = \begin{bmatrix} 0.2570 & 0.5020 & 0.2570 & -0.3137 & 0.4456 \\ 0.3241 & -0.1174 & 0.3241 & -0.4239 & -0.2183 \\ -0.2547 & 0.6481 & -0.2547 & -0.2116 & 0.1817 \\ 0.7298 & 0.0683 & -0.2702 & 0.1778 & -0.0238 \\ 0.3233 & 0.2657 & 0.3233 & -0.0289 & -0.4406 \\ -0.2702 & 0.0683 & 0.7298 & 0.1778 & -0.0238 \\ 0.1351 & 0.2548 & 0.1351 & 0.7797 & 0.1695 \\ 0.1887 & -0.4114 & 0.1887 & -0.0719 & 0.7048 \end{bmatrix}$$

$$\mathcal{N}(G_2) = \begin{bmatrix} 0.2465 & 0.5104 & 0.2475 & -0.3197 & 0.4426 \\ 0.3260 & -0.1063 & 0.3260 & -0.4218 & -0.2226 \\ -0.2633 & 0.6449 & -0.2633 & -0.2076 & 0.1731 \\ 0.7297 & 0.0713 & -0.2703 & 0.1778 & -0.0162 \\ 0.3239 & 0.2727 & 0.3239 & -0.0180 & -0.4360 \\ -0.2703 & 0.0713 & 0.7297 & 0.1778 & -0.0162 \\ 0.1273 & 0.2617 & 0.1273 & 0.7800 & 0.1676 \\ 0.1911 & -0.3991 & 0.1911 & -0.0693 & 0.7113 \end{bmatrix}$$

$$\mathcal{N}(G_3) = \begin{bmatrix} 0.2310 & 0.5241 & 0.2310 & -0.3304 & 0.4365 \\ 0.3286 & -0.0872 & 0.3286 & -0.4176 & -0.2307 \\ -0.2778 & 0.6388 & -0.2778 & -0.2003 & 0.1584 \\ 0.7294 & 0.0762 & -0.2706 & 0.1773 & -0.0030 \\ 0.3247 & 0.2848 & 0.3247 & 0.0010 & -0.4275 \\ -0.2706 & 0.0762 & 0.7294 & 0.1773 & -0.0030 \\ 0.1139 & 0.2728 & 0.1139 & 0.7805 & 0.1639 \\ 0.1945 & -0.3775 & 0.1945 & -0.0649 & 0.7221 \end{bmatrix}$$

$$\lambda = \begin{bmatrix} 6.0300 & -3.4300 & 10.9726 & -3.0368 & 0.0000 & -0.0000 & 0 & -4.8510 \\ 6.0300 & -3.4300 & 11.0412 & -3.1117 & 0.0000 & -0.0000 & 0 & -4.8510 \\ 6.0300 & -3.4300 & 11.1572 & -3.2393 & 0.0000 & 0.0000 & 0 & -4.8510 \end{bmatrix}^T$$

$$v_{cc}^{\{\mathbb{N}\}} = \begin{bmatrix} -0.2017 & 0.0198 & -0.0 & -0.0 & -0.0004 & 0.2016 & -0.1010 & 0.1006 \\ -0.3589 & 0.0244 & -0.0 & 0.0 & -0.0114 & 0.3578 & -0.1846 & 0.1732 \\ -0.6495 & 0.0094 & -0.0 & -0.0 & -0.0550 & 0.6440 & -0.3495 & 0.2945 \end{bmatrix}^T$$

Here, **λ** is the contact force where the **1st** column is related to the first flip. The same applies for the rest of the remaining columns. The first two elements of every column are the force the hand applies to the object and the last two are related to gravity force acting at the **COM** (The $\mathbf{5}^{\boldsymbol{rd}}$ finger).

As it is clearly seen, $\boldsymbol{\mathcal{N}(G_i) \neq 0, \forall\, i = 1,2,3}$ (null space of the grasp matrix for the $\mathbf{i_{th}}$ time step) makes the grasp be graspable. For more information on graspability (please refer to **Method and theory** chapter – **grasp classification**).

In **Figure 12**, the wrenches that has been measured by force-torque sensor is presented*:*

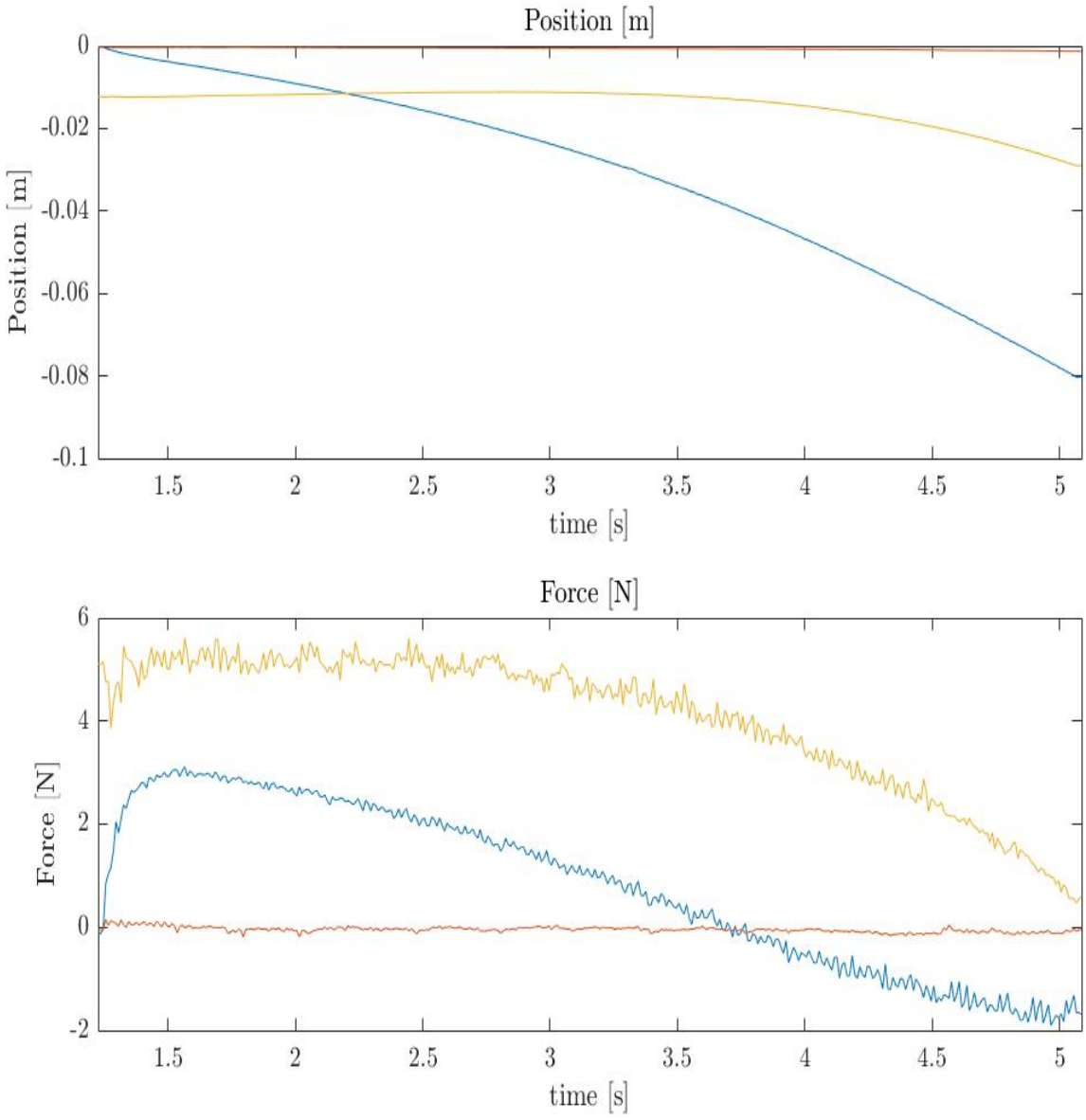

Figure 12 - Toppling P_1 = (-6, 6.7) [cm] wrenches applied to the system measured by force-torque sensor

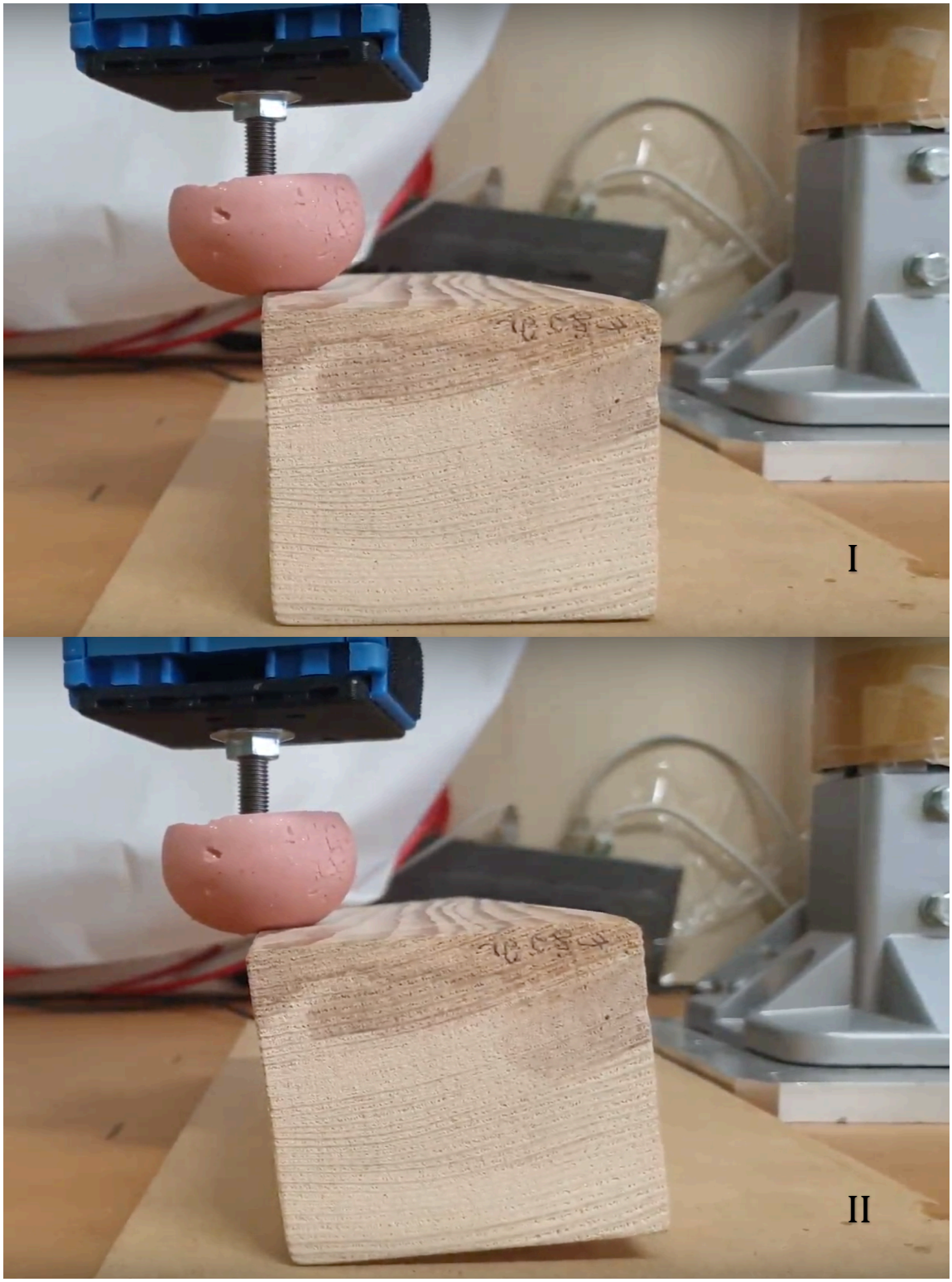

Figure 13 -Toppling experiment P1 = (-6, 6.7) [cm]

Based on what is obtained from optimization**,** null space of transpose of grasp matrix is a null matrix ($\boldsymbol{\mathcal{N}\left(G_i^T\right) = 0, \forall\, i = 1, 2, 3}$). based on Error! Reference source not found.**,** the grasp is determinable**.**

Based on the **task-imposed constraints**, the $\mathbf{3^{rd}}$ *and* $\mathbf{4^{th}}$, which are corresponding to the pivot point (**second contact point** in this thesis), components of $\boldsymbol{v}_{cc}^{\{\mathbb{N}\}}$ (acquired from **Eq. (39)**) are almost zero ($\boldsymbol{v}_{cc}^{\{\mathbb{N}\}}(\mathbf{3},\mathbf{4}) \cong \mathbf{0}$).

4.3.1.2. $P_1 = (4.7,6.7)$:

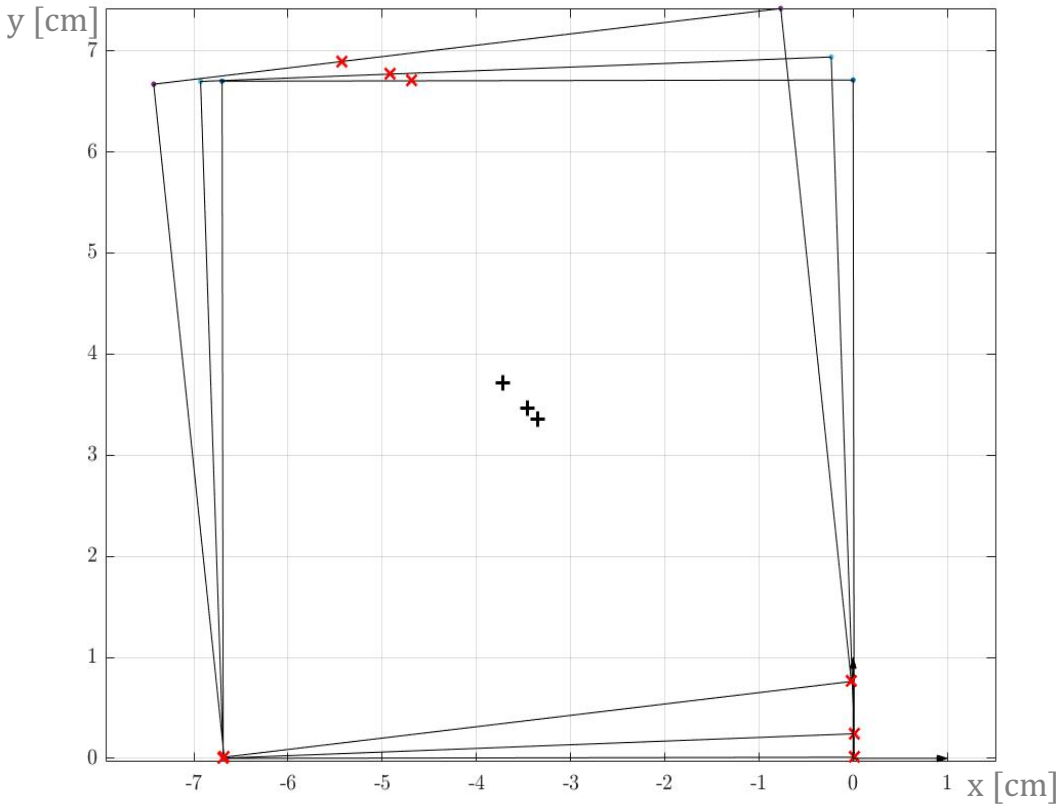

Figure 14 - Toppling - P_1 = (4.7,6.7) [cm]

$$G_1 = \begin{bmatrix} 0.0017 & 1.0000 & -0.0017 & -1.0000 & -0.0017 & -1.0000 & 1 & 0 \\ -1.0000 & 0.0017 & 1.0000 & -0.0017 & 1.0000 & -0.0017 & 0 & 1 \\ 1.3400 & -3.3500 & -3.3500 & -3.3500 & 3.3500 & -3.3500 & 0 & 0 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} 0.1118 & 0.9937 & -0.1118 & -0.9937 & -0.1118 & -0.9937 & 1 & 0 \\ -0.9937 & 0.1118 & 0.9937 & -0.1118 & 0.9937 & -0.1118 & 0 & 1 \\ 1.3400 & -3.3500 & -3.3500 & -3.3500 & 3.3500 & -3.3500 & 0 & 0 \end{bmatrix}$$

$$G_3 = \begin{bmatrix} 0.0939 & 0.9956 & -0.0939 & -0.9956 & -0.0939 & -0.9956 & 1 & 0 \\ -0.9956 & 0.0939 & 0.9956 & -0.0939 & 0.9956 & -0.0939 & 0 & 1 \\ 2.6800 & -3.3500 & -3.3500 & -3.3500 & 3.3500 & -3.3500 & 0 & 0 \end{bmatrix}$$

$$\mathcal{N}(\mathrm{G}_1) = \begin{bmatrix} 0.2082 & 0.5725 & 0.2082 & -0.3128 & 0.4682 \\ 0.3081 & -0.1257 & 0.3081 & -0.4124 & -0.2312 \\ -0.3178 & 0.5870 & -0.3178 & -0.1566 & -0.1125 \\ 0.7171 & 0.0565 & -0.2829 & 0.1887 & -0.0374 \\ 0.3413 & 0.3452 & 0.3413 & -0.0665 & -0.3808 \\ -0.2829 & 0.0565 & 0.7171 & 0.1887 & -0.0374 \\ 0.1259 & 0.2393 & 0.1259 & 0.7900 & 0.1551 \\ 0.1849 & -0.3592 & 0.1849 & -0.0883 & 0.7367 \end{bmatrix}$$

$$\mathcal{N}(\mathrm{G}_2) = \begin{bmatrix} 0.1969 & 0.5805 & 0.1969 & -0.3195 & 0.4636 \\ 0.3102 & -0.1142 & 0.3102 & -0.4094 & -0.2366 \\ -0.3256 & 0.5815 & -0.3256 & -0.1511 & -0.1032 \\ 0.7173 & 0.0592 & -0.2827 & 0.1890 & -0.0288 \\ 0.3404 & 0.3535 & 0.3404 & -0.0547 & -0.3765 \\ -0.2827 & 0.0592 & 0.7173 & 0.1890 & -0.0288 \\ 0.1177 & 0.2453 & 0.1177 & 0.7910 & 0.1523 \\ 0.1865 & -0.3459 & 0.1865 & -0.0852 & 0.7428 \end{bmatrix}$$

$$\mathcal{N}(\mathrm{G}_3) = \begin{bmatrix} 0.1729 & 0.5960 & 0.1729 & -0.3349 & 0.4521 \\ 0.3141 & -0.0897 & 0.3141 & -0.4020 & -0.2495 \\ -0.3417 & 0.5691 & -0.3417 & 0.1385 & 0.0842 \\ 0.7175 & 0.0650 & -0.2825 & 0.1886 & -0.0102 \\ 0.3381 & 0.3711 & 0.3381 & -0.0293 & -0.3665 \\ -0.2825 & 0.0650 & 0.7175 & 0.1886 & -0.0102 \\ 0.1004 & 0.2568 & 0.1004 & 0.7930 & 0.1456 \\ 0.1889 & -0.3174 & 0.1889 & -0.0789 & 0.7554 \end{bmatrix}$$

$$\lambda = \begin{bmatrix} 9.6100 & -4.8300 & 13.9821 & -4.5442 & 0.5836 & -0.1897 & 0 & -4.8510 \\ 10.098 & -5.1100 & 14.0582 & -4.5689 & 0.6309 & -0.2050 & 0 & -4.8510 \\ 10.098 & -5.1100 & 14.2155 & -4.6200 & 0.7301 & -0.2373 & 0 & -4.8510 \end{bmatrix}^T$$

$$v_{cc}^{\{\mathbb{N}\}} = \begin{bmatrix} -0.2303 & 0.0687 & -0.00 & -0.00 & -0.0004 & 0.2302 & -0.1153 & 0.1149 \\ -0.5138 & 0.1341 & -0.00 & 0.00 & -0.0184 & 0.5083 & -0.2633 & 0.2450 \\ -1.1638 & 0.2111 & 0.00 & 0.00 & -0.1266 & 1.1258 & -0.6262 & 0.4996 \end{bmatrix}^T$$

As it is clearly seen, for all the instances, null space of the grasp matrix **G** is not zero **($\mathcal{N}(\boldsymbol{G}_i) \neq \mathbf{0}$, $\forall$ i = 1,2,3)** which makes the grasp be graspable. Null space of the transpose of the grasp matrix is determinable. Again, the velocity of the object at the

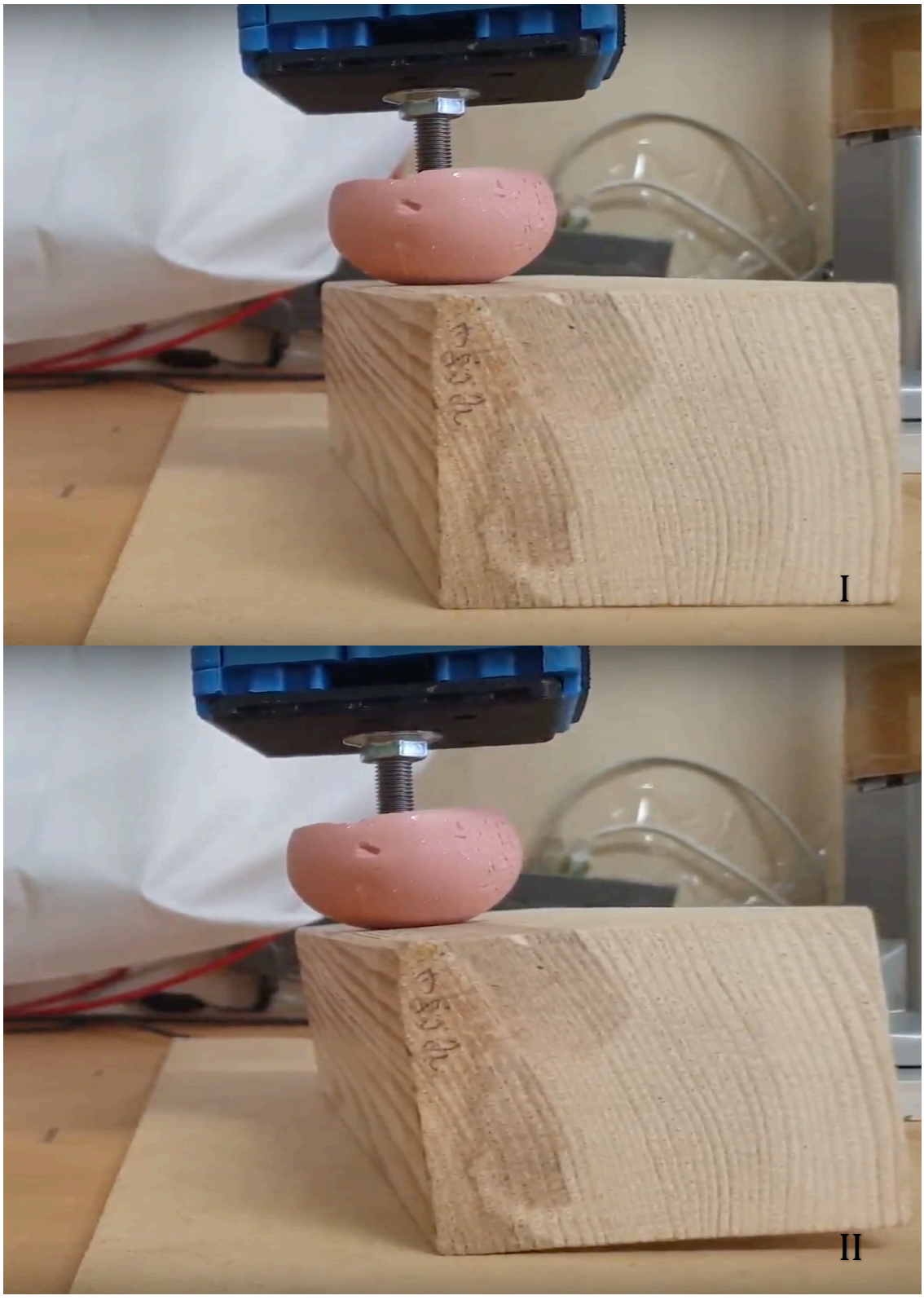

Figure 15 -Toppling experiment P1 = (-5, 6.7) [cm]

pivot point with respect to the inertial frame $\{\mathbb{N}\}$ (2^{nd} contact point) is almost zero which is what is required for the task.

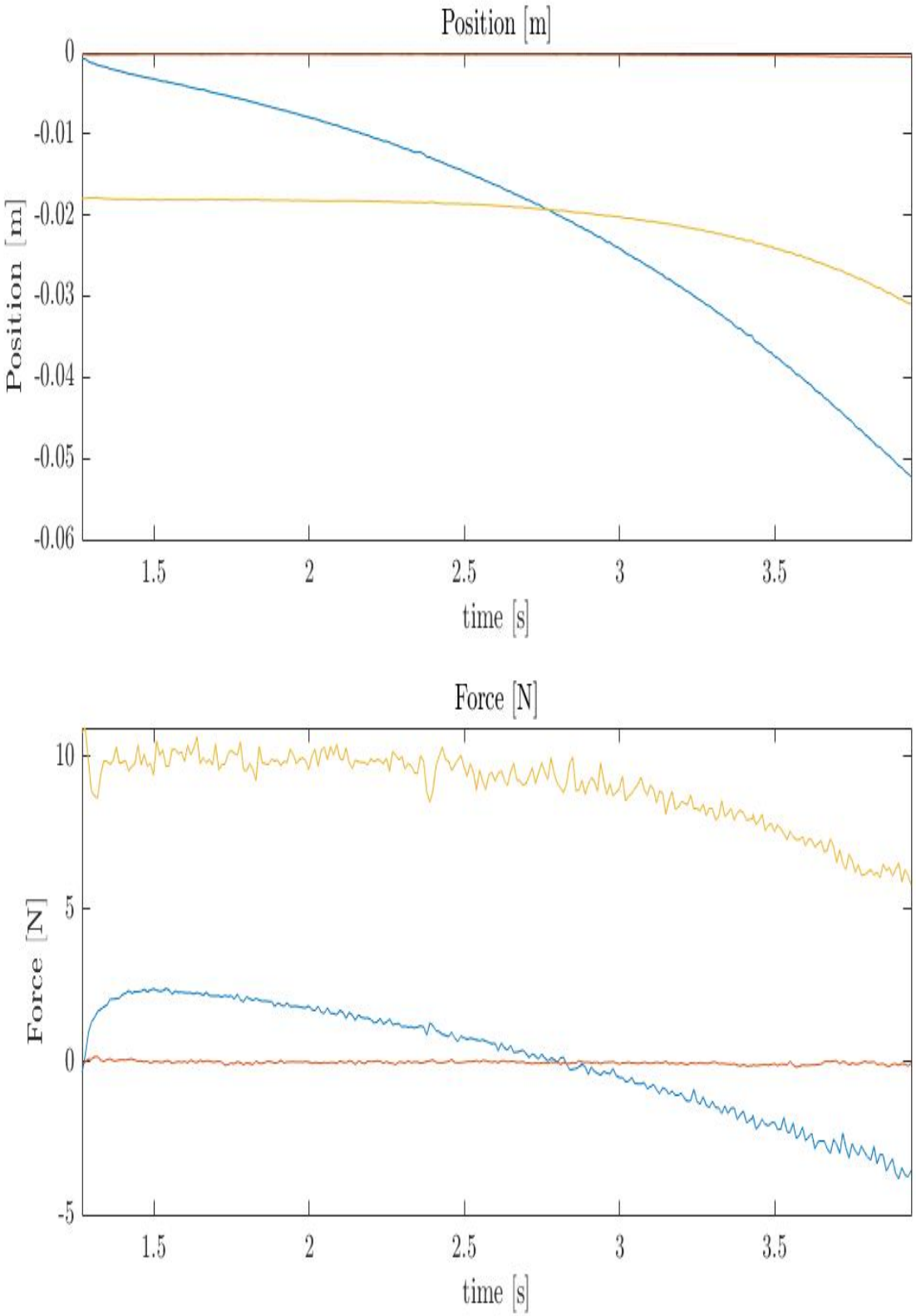

Figure 16 - Toppling P1 = (-5, 6.7) [cm] wrenches applied to the system measured by force-torque sensor

4.3.2. Wall-grasp:

Likewise toppling, same logic applied to contact numbering **Figure 17**. It is no harm to remind that, regardless of the sequencing used, it must correspond to what is typed in **Matlab®** code. In **Figure 17**, at the lower left corner of the object where it contacts the obstacle, **contact 3** and 4 are presented with one cross for sake of simplicity. Whilst the **3rd** point of the object is in contact with the wall, the **4th** has collision with the table.

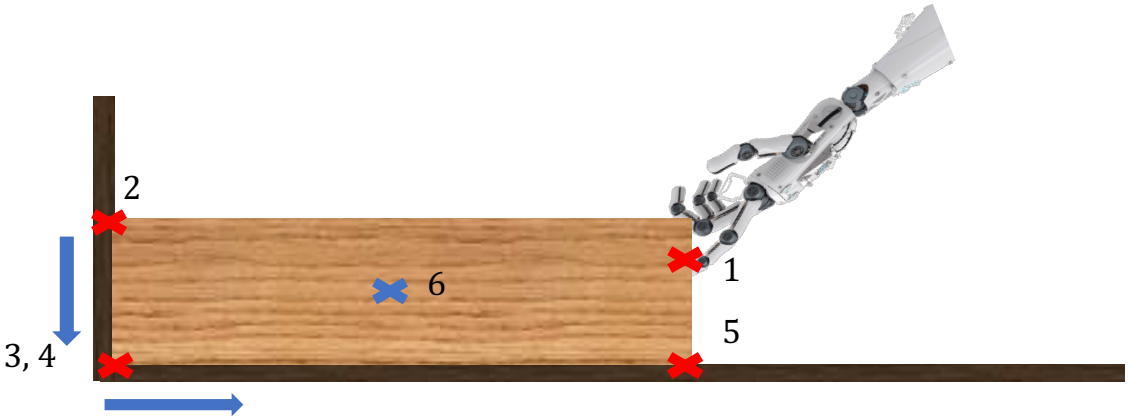

Figure 17 - Contact point location in wall-grasp

4.3.2.1. $P_1 = (0,4.345)$:

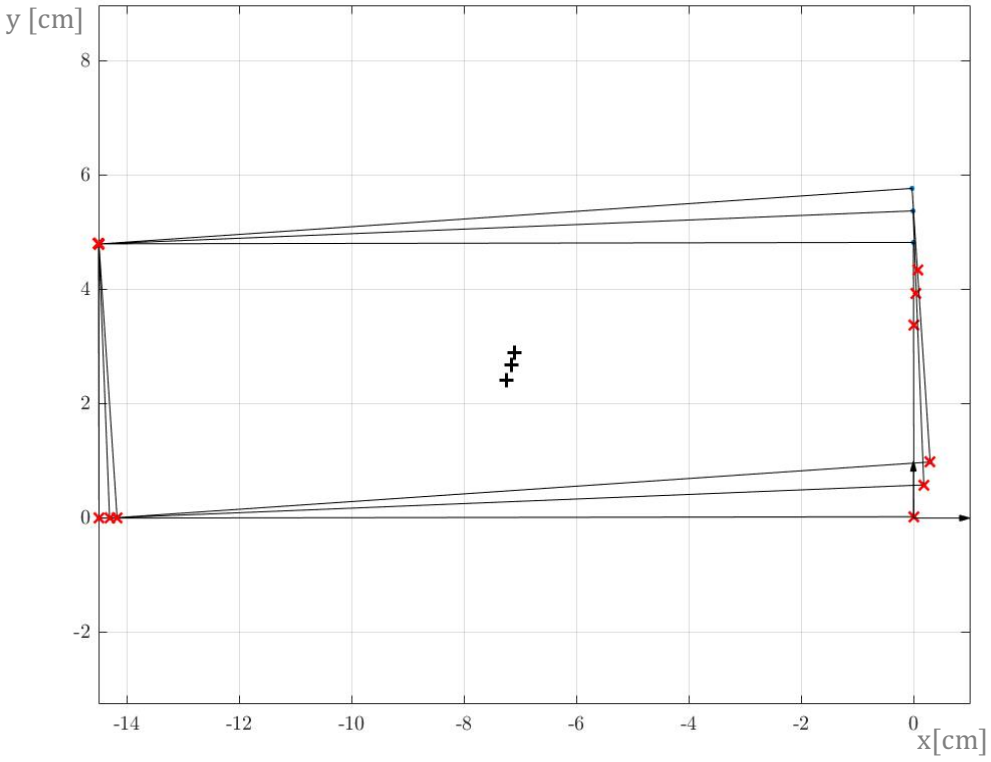

Figure 18 - wall-grasp - P_1 = (0,4.345) [cm]

$$\mathcal{N}(G_1) = \begin{bmatrix} -0.1192 & 0.4448 & -0.1192 & -0.1192 & -0.4448 & 0.0838 & -0.4448 \\ 0.1807 & 0.0703 & 0.1807 & 0.1807 & -0.0703 & 0.5691 & -0.0703 \\ -0.4737 & 0.0273 & -0.4737 & -0.4737 & -0.0273 & 0.3348 & -0.0273 \\ 0.7192 & 0.0500 & -0.2808 & -0.2808 & -0.0500 & -0.0035 & -0.0500 \\ 0.1047 & 0.8494 & 0.1047 & 0.1047 & 0.1506 & -0.0742 & 0.1506 \\ -0.2808 & 0.0500 & 0.7192 & -0.2808 & -0.0500 & -0.0035 & -0.0500 \\ -0.2808 & 0.0500 & -0.2808 & 0.7192 & -0.0500 & -0.0035 & -0.0500 \\ -0.1047 & 0.1506 & -0.1047 & -0.1047 & 0.8494 & 0.0742 & -0.1506 \\ 0.1090 & -0.0699 & 0.1090 & 0.1090 & 0.0699 & 0.7209 & 0.0699 \\ -0.1047 & 0.1506 & -0.1047 & -0.1047 & -0.1506 & 0.0742 & 0.8494 \\ 0.0404 & -0.1307 & 0.0404 & 0.0404 & 0.1307 & -0.0283 & 0.1307 \\ -0.0858 & -0.0102 & -0.0858 & -0.0858 & 0.0102 & -0.1413 & 0.0102 \end{bmatrix}$$

$$\cdots \begin{bmatrix} 0.4113 & -0.0170 \\ 0.0053 & 0.3749 \\ -0.1064 & -0.0696 \\ 0.0044 & -0.1421 \\ -0.1210 & 0.0151 \\ 0.0044 & -0.1421 \\ 0.0044 & -0.1421 \\ 0.1210 & -0.0151 \\ -0.0055 & -0.0850 \\ 0.1210 & -0.0151 \\ 0.8806 & 0.0058 \\ -0.0008 & 0.8864 \end{bmatrix}$$

$$\mathcal{N}(\mathrm{G}_2) = \begin{bmatrix} -0.1217 & 0.4444 & -0.1217 & -0.1217 & -0.4444 & 0.0810 & -0.4444 \\ 0.1797 & 0.0724 & 0.1797 & 0.1797 & -0.0724 & 0.5695 & -0.0724 \\ -0.4740 & 0.0257 & -0.4740 & -0.4740 & -0.0257 & 0.3334 & -0.0257 \\ 0.7190 & 0.0500 & -0.2810 & -0.2810 & -0.0500 & -0.0036 & -0.0500 \\ 0.1042 & 0.8498 & 0.1042 & 0.1042 & 0.1502 & -0.0745 & 0.1502 \\ -0.2810 & 0.0500 & 0.7190 & -0.2810 & -0.0500 & -0.0036 & -0.0500 \\ -0.2810 & 0.0500 & -0.2810 & 0.7190 & -0.0500 & -0.0036 & -0.0500 \\ -0.1042 & 0.1502 & -0.1042 & -0.1042 & 0.8498 & 0.0745 & -0.1502 \\ 0.1087 & -0.0685 & 0.1087 & 0.1087 & 0.0685 & 0.7215 & 0.0685 \\ -0.1042 & 0.1502 & -0.1042 & -0.1042 & -0.1502 & 0.0745 & 0.8498 \\ 0.0411 & -0.1304 & 0.0411 & 0.0411 & 0.1304 & -0.0266 & 0.1304 \\ -0.0855 & -0.0114 & -0.0855 & -0.0855 & 0.0114 & -0.1415 & 0.0114 \end{bmatrix}$$

$$\begin{matrix} & \begin{matrix} 0.4111 & -0.0135 \\ 0.0016 & 0.3747 \\ -0.1068 & -0.0721 \\ 0.0065 & -0.1422 \\ -0.1209 & 0.0128 \\ 0.0065 & -0.1422 \\ 0.0065 & -0.1422 \\ 0.1209 & -0.0128 \\ -0.0030 & -0.0850 \\ 0.1209 & -0.0128 \\ 0.8807 & 0.0053 \\ -0.0002 & 0.8865 \end{matrix} \\ \cdots & \end{matrix}$$

$$\mathcal{N}(G_3) = \begin{bmatrix} -0.1232 & 0.4441 & -0.1232 & -0.1232 & -0.4441 & 0.0793 & -0.4441 \\ 0.1790 & 0.0737 & 0.1790 & 0.1790 & -0.0737 & 0.5697 & -0.0737 \\ -0.4741 & 0.0247 & -0.4741 & -0.4741 & -0.0247 & 0.3326 & -0.0247 \\ 0.7190 & 0.0501 & -0.2810 & -0.2810 & -0.0501 & -0.0037 & -0.0501 \\ 0.1038 & 0.8500 & 0.1038 & 0.1038 & 0.1500 & -0.0747 & 0.1500 \\ -0.2810 & 0.0501 & 0.7190 & -0.2810 & -0.0501 & -0.0037 & -0.0501 \\ -0.2810 & 0.0501 & -0.2810 & 0.7190 & -0.0501 & -0.0037 & -0.0501 \\ -0.1038 & 0.1500 & -0.1038 & -0.1038 & 0.8500 & 0.0747 & -0.1500 \\ 0.1085 & -0.0677 & 0.1085 & 0.1085 & 0.0677 & 0.7218 & 0.0677 \\ -0.1038 & 0.1500 & -0.1038 & -0.1038 & -0.1500 & 0.0747 & 0.8500 \\ 0.0416 & -0.1303 & 0.0416 & 0.0416 & 0.1303 & -0.0256 & 0.1303 \\ -0.0853 & -0.0122 & 0.0416 & 0.0416 & 0.0122 & -0.1417 & 0.0122 \end{bmatrix}$$

$$\begin{matrix} & \begin{matrix} 0.4110 & -0.0114 \\ -0.0006 & 0.3745 \\ -0.1070 & -0.0735 \\ 0.0078 & -0.1422 \\ -0.1208 & 0.0115 \\ 0.0078 & -0.1422 \\ 0.0078 & -0.1422 \\ 0.1208 & -0.0115 \\ -0.0015 & -0.0849 \\ 0.1208 & -0.0115 \\ 0.8807 & 0.0049 \\ 0.0001 & 0.8865 \end{matrix} \\ \cdots & \end{matrix}$$

$$\lambda = \begin{bmatrix} 10.000 & -4.500 & 12.1102 & -3.9287 & 0.0074 & -0.0003 & 6.3850 & 2.0729 \\ 10.000 & -4.500 & 12.2094 & -3.9681 & 0.0000 & -0.0000 & 6.3864 & 2.0756 \dots \\ 10.000 & -4.5000 & 12.2654 & -3.9862 & 0.0001 & -0.0000 & 6.3864 & 2.0755 \end{bmatrix}$$

$$\begin{matrix} & \begin{matrix} 0.0023 & 0.0001 & -0.0000 & -6.8600 \\ 0.0000 & 0.0000 & -0.0000 & -6.8600 \\ 0.0000 & 0.0000 & 0.0000 & -6.8600 \end{matrix}^T \\ \dots & \end{matrix}$$

$$v_{cc}^{\{\mathbb{N}\}} = \begin{bmatrix} 0.0068 & 0.2162 & 0.00 & 0.0000 & 0.0716 & 0.0001 & 0.0716 & 0.0001 \\ 0.0021 & 0.1305 & -0.00 & -0.0000 & 0.0432 & 0.0007 & 0.0432 & 0.0007 \\ 0.0007 & 0.0934 & 0.00 & 0.0000 & 0.0309 & 0.0008 & 0.0309 & 0.0008 \end{bmatrix} \ldots$$

$$\ldots \begin{bmatrix} 0.0712 & 0.2163 & 0.0356 & 0.1082 \\ 0.0410 & 0.1311 & 0.0205 & 0.0656 \\ 0.0285 & 0.0941 & 0.0143 & 0.0471 \end{bmatrix}^T$$

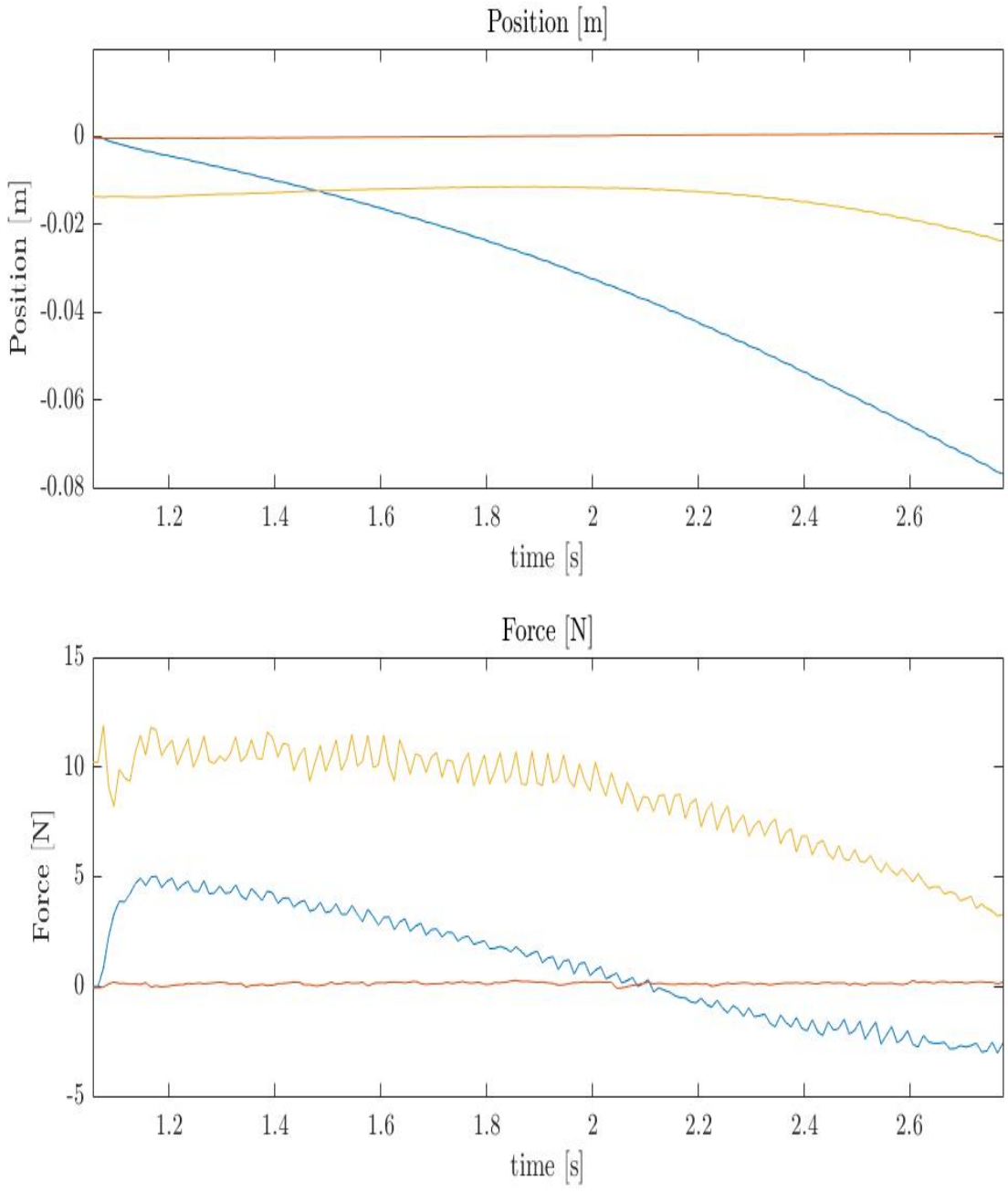

Figure 19 - wall-grasp - P_1 = (0,4.345) [cm] wrenches applied to the system measured by force-torque sensor

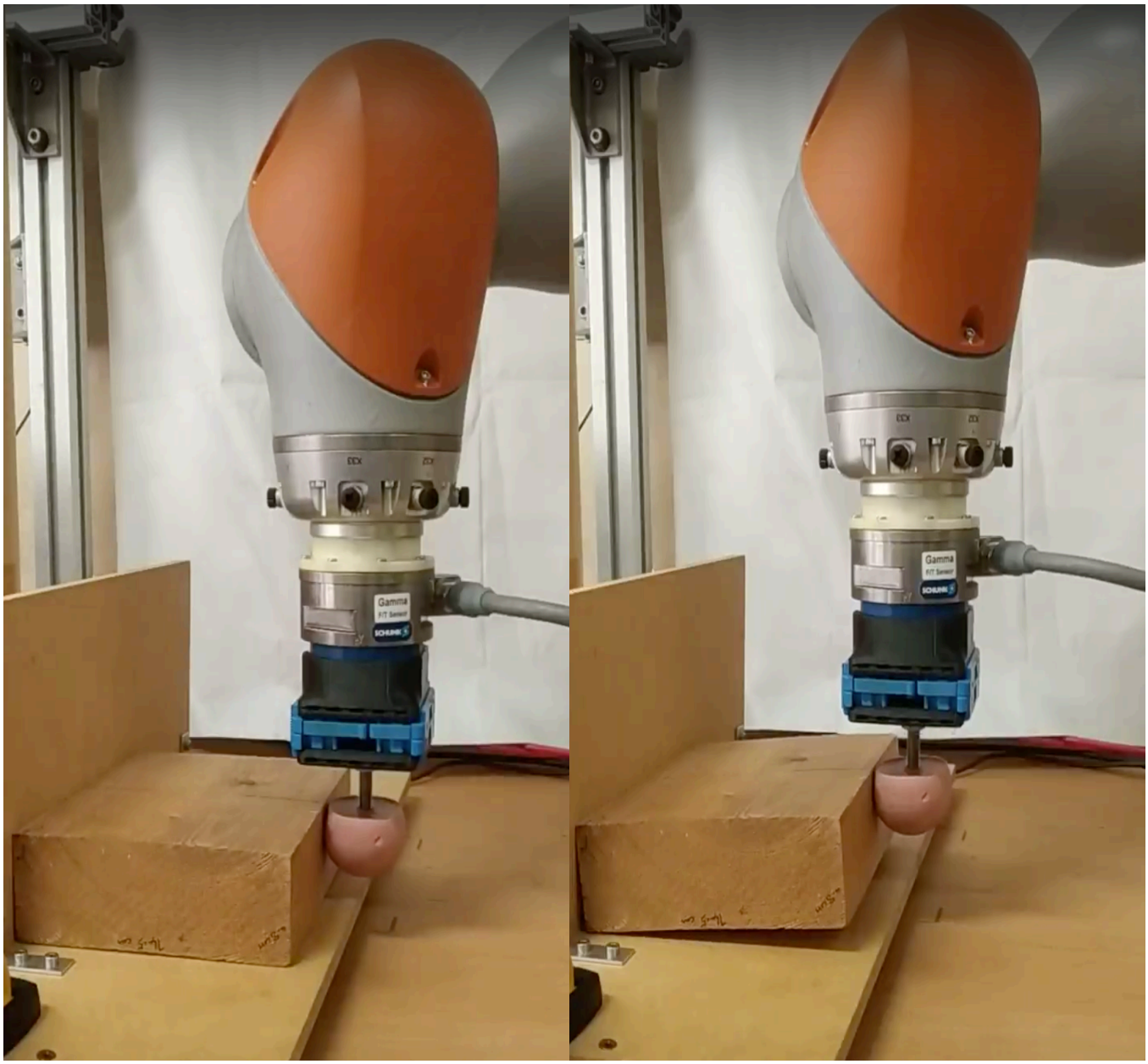

Figure 20 – wall-grasp experiment - P_1 = (0,4.345) [cm]

The **graspability** feature of the grasp is well-evident for all the steps as the null space of grasp matrix are all non-zero ($\boldsymbol{\mathcal{N}(G_i) \neq \mathbf{0}, \forall\, i = 1,2,3}$). This guarantee the grasp under consideration to maintain the grasp while object is moving. The null space of the transpose of the grasp matrix is null ($\boldsymbol{\mathcal{N}\left(G_i^T\right) \neq \mathbf{0}, \forall\, i = 1,2,3}$) that makes the grasp be **determinable.**

Here, it is apparent that the task-imposed constraints are all satisfied. Here, the normal components of the contact velocity **(the 3^{rd} and 4^{rd}** components of $\boldsymbol{v_{cc}^{\{N\}}}$**),** with respect **to** the global frame **{N}**, should be **zero** ($\boldsymbol{v_{cc}^{\{\mathbb{N}\}}(3,4) \cong 0}$**).** This guarantees that the object trajectory following the permissible movement**.**

5. Conclusion and future works

Grasping and manipulation's aim is to build, plan and control a robotic hand that can enable it to grasp objects in a delicate yet secure way. As the equations governing the interaction of the robotic hand with its environment have nonlinearities, the problem if grasping or manipulation are not trivial, however easy they might seem to us as humans. Exploiting environmental constraints makes it easier for the robotic hand, especially one with some level of compliance, to have a secure grip which is different in comparison with a conventional robotics that tries to avoid obstacle as much as possible.

In this thesis, out of three maneuvers to grasp and manipulate the object, toppling and wall-grasp are studied using a novel approach based on grasp matrix **G**. Another point is the way gravity is included in the formulation by considering a finger of the grasp pointing downward. This approach to include the gravity makes it computationally easier in comparison with its counterparts like gravity compensator as a case in point. The simulation for each of the mentioned approaches provide a verification of what is obtained. Lastly the simulation itself is confirmed using experiments with the robot.

Although the results are promising, we are still far away from a model that can capture all the dimensions of the system in a way that leads us to replicate a smarter biologic manipulator like human hand. Due to the hysteresis behavior of the elastic materials and complicated order of the model, compliant contact models are not implemented. `instead a rigid body contact is used, hence, it imposes static indeterminacy to the problem that make the model not accurate as it should be. Lastly, nonlinear behavior of the system is not possible to sufficiently included in the model that reduces the desired outcome.

To improve the result, it is better to incorporate machine learning methods on the top of the current physic simulator to include the nonlinearities and sort of information that is very hard to include in the model directly. This not only makes the model more accurate and robust but also more generic to be able to apply for unstructured environment. Another step that can be taken to make the model be accurate is to include Model Predictive Control (**MPC**) algorithms especially Non-linear MPC (**NMPC**) in to account. This approach is good as the dynamic of the system is slow enough (thanks to quasi-static assumption that has been taken) to have the prediction of the factors affecting the quality of the grasp on-line.

References

[1] M. Peshkin and A. Sanderson, “Minimization of energy in quasistatic manipulation,” in *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, Philadelphia, PA, USA, USA, 1988.

[2] H. Moravec, The future of robot and human intelligence, boston massachusetts: Harvard University Press, 1998.

[3] J. Trinkle, “Planning for Dexterous Manipulation with Sliding Contac,” *IJRR,* vol. 9, no. 3, pp. 24-48, 1 June 1990.

[4] A. Cole, J. Hauser and S. Sastry, “Kinematics and control of multifingered hands with rolling contact,” in *IEEE International Conference on Robotics and Automation*, Philadelphia, PA, USA, 1989.

[5] M. Erdmann, “A configuration space friction cone,” in *IEEE/RSJ International Workshop on Intelligent Robots and Systems '91*, Osaka, Japan, Japan, 1991.

[6] M. Buss, H. Hashimoto and J. Moore, “Dextrous hand grasping force optimization,” *IEEE Transactions on Robotics and Automation ,* vol. 12, no. 3, pp. 406 - 418, 1996.

[7] L. Han, J. Trinkle and Z. Li, “Grasp analysis as linear matrix inequality problems,” *IEEE Transactions on Robotics and Automation,* vol. 16, no. 6, Dec 2000.

[8] D. Prattichizzo, M. Malvezzi, M. Gabiccini and A. Bicchi, “On Motion and Force Controllability of Precision Grasps with Hands Actuated by Soft Synergies,” *IEEE Transactions on Robotics ,* vol. 29 , no. 6, pp. 1440 - 1456, 2013.

[9] N. C. Dafle, A. Rodriguez, R. Paolini, B. Tang, S. S. Srinivasa, M. Erdmann, M. T. Mason, I. Lundberg and H. Staab, “Extrinsic dexterity: In-hand manipulation with external forces,” in *IEEE International Conference on Robotics & Automation (ICRA)*, Hong Kong, China, 2014.

[10] M. Bonilla, D. Resasco, M. Gabiccini and A. Bicchi, “Grasp Planning with Soft Hands using Bounding Box Object Decomposition,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, 2015 .

[11] M. Pozzi, A. M. Sundaram, M. Malvezzi, D. Prattichizzo and M. A. Roa, “Grasp quality evaluation in underactuated robotic hands,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, South Korea, 2016.

[12] F. Heinemann, S. Puhlmann, C. Eppner, J. Élvarez-Ruiz, M. Maertens and O. Brock, “A taxonomy of human grasping behavior suitable for transfer to robotic hands,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA, 2015.

[13] M. Pozzi, G. Salvietti, J. Bimbo, M. Malvezzi and D. Prattichizzo, "The Closure Signature: A Functional Approach to Model Underactuated Compliant Robotic Hands," *IEEE Robotics and Automation Letters* , vol. 3, no. 3, pp. 2206 - 2213, 2018.

[14] M. M. G. Ardakani, J. Bimbo and D. Prattichizzo, "Quasi-static Analysis of Planar Sliding Using Friction Patches," *arXiv:1904.06677,* vol. 1, 2019.

[15] Y. Hou and M. T. Mason, "Robust Execution of Contact-Rich Motion Plans by Hybrid Force-Velocity Control," *arXiv:1903.02715v1,* vol. 1, 2019.

[16] O. Kroemer, "Machine Learning for Robot Grasping and Manipulation," TU Darmstadt, Darmstadt, 2015.

[17] A. Zeng, S. Song, J. Lee, A. Rodriguez and T. Funkhouser, "TossingBot: Learning to Throw Arbitrary Objects with Residual Physics," *arXiv:1903.11239v2,* vol. 2, 3 July 2019.

[18] D. Prattichizzo and J. C. Trinkle, "Grasping," in *Handbook of robotics*, Springer, pp. 955 - 988.

[19] M. S. Ohwovoriole and B. Roth, "An Extension of Screw Theory," *Mech. Des,* pp. 725 - 735, 1981.

[20] Y.-B. Jia, "Plücker Coordinates for Lines in the Space," 29 August 2019. [Online]. Available: http://web.cs.iastate.edu/~cs577/handouts/plucker-coordinates.pdf.

[21] W. V. D. Hodge and D. Pedoe, Methods of Algebraic Geometry, vol. 1, Cambridge: Cambridge University Press, 1994.

[22] J. Denavit and R. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," *ASME Journal of Applied Mechanics,* vol. 22, pp. 215 - 221, 1951.

[23] I. Kao, K. M. Lynch and J. W. Burdick, "Contact Modeling and Manipulation," in *Handbook of robotics*, 2nd, Ed., Springer, pp. 931 - 954.

[24] P. Prattichizzo, M. Malvezz, M. Gabiccini and A. Bicchi, "On the problem of decomposing grasp and manipulation forces in multiple whole-limb manipulation," *Robotics and Autonomous Systems,* vol. 13, no. 2, pp. 127 - 147, 1994.

[25] Y.-T. Wang, V. Kumar and J. Abel, "Dynamics of rigid bodies undergoing multiple frictional contacts," in *Proceedings 1992 IEEE International Conference on Robotics and Automation*, Nice, France, France, 1992.

[26] J. Salisbury and B. Roth, " Kinematic and force analysis of articulated mechanical hands, Transm. Autom. Des. 105(1), 35–41 (1983)," *J. Mech., Trans., and Automation,* pp. 35 - 41, 1 March 1983 .

[27] J. Kerr and B. Roth, "Analysis of Multifingered Hands," *The International Journal of Robotics Research (IJRR),* vol. 4, no. 4, pp. 3 - 17, 1986 .

[28] M. H. (originator), “Encyclopedia of Mathematics,” Springer, [Online]. Available: http://www.encyclopediaofmath.org/index.php?title=Kernel_of_a_matrix&oldid=12040.

[29] “Kernel (linear algebra),” [Online]. Available: https://en.wikipedia.org/wiki/Kernel_(linear_algebra).

[30] KUKA Deutschland GmbH, "*Spez LBR iiwa V8," PB2535 datasheet, May* 2019.

[31] ATI Industrial Automation, Inc., "*Axia80," 9205-05-1026-05 datasheet, 2018 .*

[32] SCHUNK GmbH & Co. KG, "*Assembly and operating manual FT-AXIA 80 Ethernet," IM0024484 datasheet, Dec. 2018.*

Appendix

Experimental setup

In this thesis, after mathematical modeling and simulation mainly in **Matlab®**, it is fed to **Gazebo** which simulates the physics if it is applicable to the robot or not. **Moveit** is used to plane the robots path. Finally, the forces that has been obtained in linear optimization is applied to the robot (**KUKA® LBR iiwa** 7) for the final verification.

Hardware:

Robot:

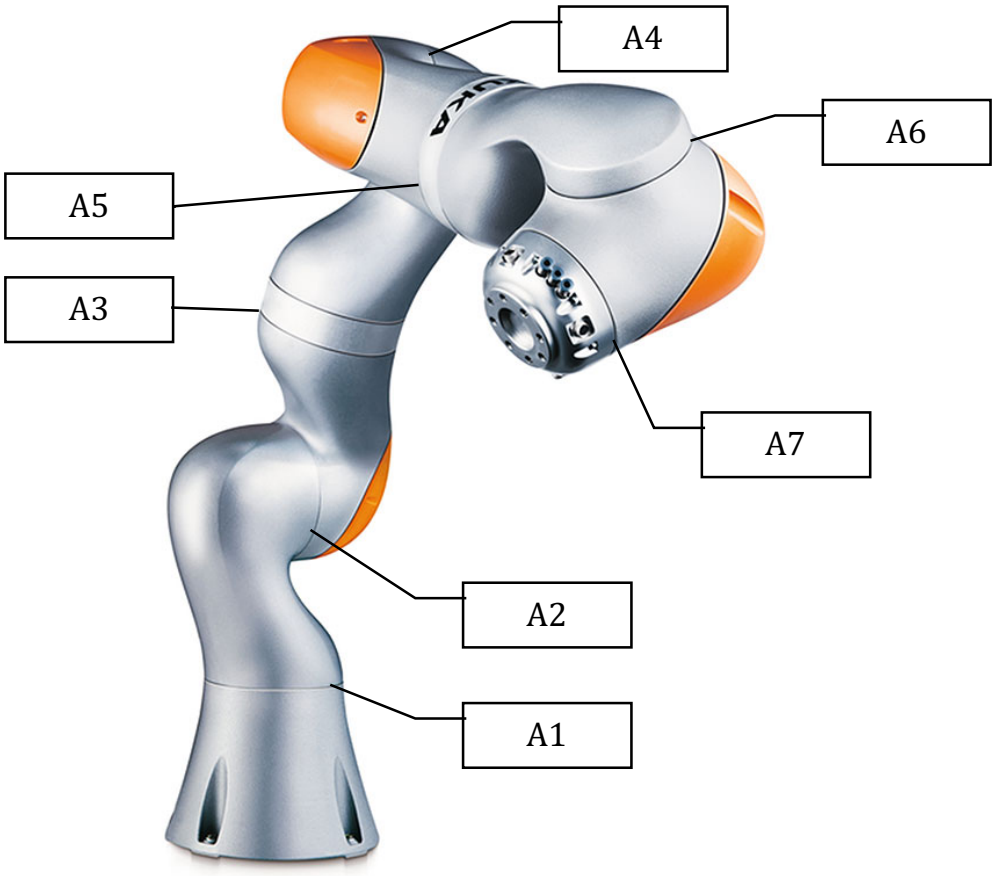

Figure 21 - KUKA LBR iiwa 7 - Adapted from: "https://robots.ieee.org/robots/lbriiwa/" belongs to KUKA GmbH

KUKA® LBR iiwa No.7 (Leichtbauroboter[1] intelligent industrial work assistants) is a light weight robot [30] comprising from **7** joints that has one degree of redundancy for

[1] German word for lightweight robot

maneuverability (in special space, a rigid body has 6 degree of freedom). Every joint is equipped with integrated torque-sensor enables the accuracy, safety and enable the sequence of robot application to be controlled without using any control device but gestures like tapping against the robot [30].

Figure 22 illustrates the work space of the robot and **Table 5** gives a brief information on the working dimension of the robot.

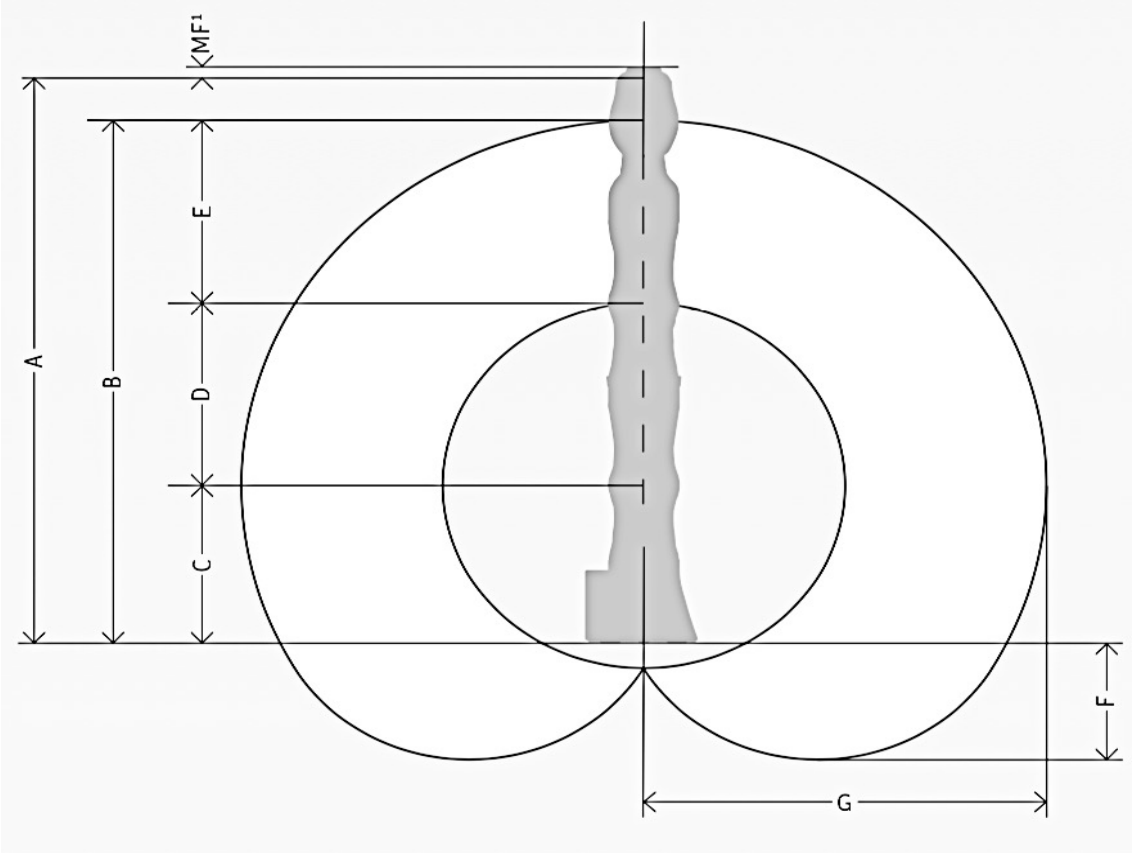

Figure 22 - working space of LBR iiwa 7 - Adapted from [30]

Table 5 - working space dimensions of LBR iiwa 7 [30]

Dim. A [mm]	Dim B [mm]	Dim. C [mm]	Dim. D [mm]	Dim. E [mm]	Dim. F [mm]	Dim. G [mm]	Volume [m^3]
1.266	1.140	340	400	400	260	800	1.7

In **Table 6** and **Table 7** iiwa LBR No.7 specifications are shown:

Table 6 - Axis data of LBR iiwa 7 [30]

	Range of motion	Maximum torque [Nm]	Maximum velocity [°/s]
Axis 1 (A1)	± 170°	176	98
Axis 2 (A2)	± 120°	176	98
Axis 3 (A3)	± 170°	110	100
Axis 4 (A4)	± 120°	110	130
Axis 5 (A5)	± 170°	110	140
Axis 6 (A6)	± 120°	40	180
Axis 7 (A7)	± 175°	40	180

The Axis corresponding to the values in **Table 6** are indicated in **Figure 21**.

Table 7 - LBR iiwa specification [30]

Rated payload	7 [kg]
Number of axis	7
Wrist variant	In-line wrist
Mounting flange A7	DIN ISO 9409-1-A50
Installation Position	Any
Positioning accuracy (ISO 9283)	± 0.1 [mm]
Axis-specific torque accuracy	± 2%
Weight	23.9 [Kg]
Protection rating	IP 54

Force/Torque sensor [31] [32]:

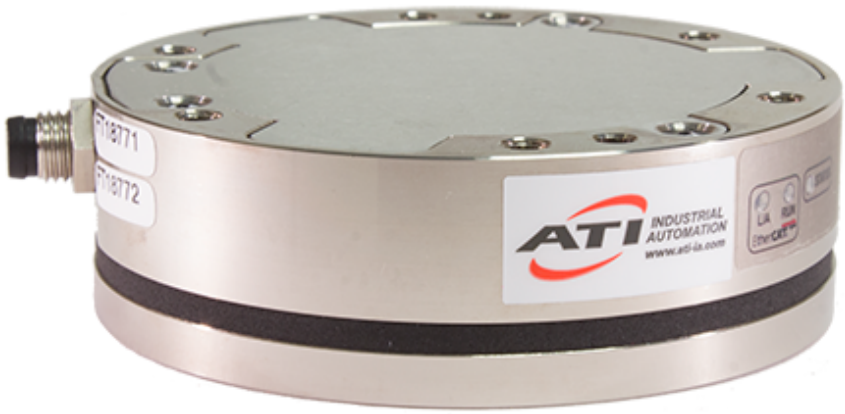

Figure 23 - Axia80 F/T Sensor - Adapted from:
"https://www.ati-ia.com/Company/NewsArticle2.aspx?id=1169833375"

Multi-Axial Axia80 Force/Torque (F/T) sensor is used to measure forces and torques in 6 axes as well as the position of the prob. The product possesses a very good accuracy with high level of precision (around **2%** full scale) for all the **6 axes**. The AD converter has a high resolution of 16-bit. The sampling frequency is **4 [kHz]**. The high **signal-to-noise (SNR)** ratio makes it very resilient to the background noises which in turn brings about robustness [31] [32].

The last but not the least is its low cost and durability. **Axia80** can endure much more than its measuring rang. In the following loading characteristic is shown. [31]

Table 8 - Axia80 loading characteristics – Adapted from [31] [32]

Loading characteristic	Fxy [N]	Fz [N]	Txy [Nm]	Tz [Nm]
Measurement range 0	±500	±900	±20	±20
Measurement range 1	±200	±360	±8	±8
Overloading rating	±2500	±4500	±100	±100
Effective resolution	0.1	0.1	0.005	0.005

Software:

Gazebo:

Gazebo is a 3-D dynamic simulator that simulate the robot in different environments for a high degree of details. Gazebo is best used in testing the robot's performance in a realistic environment.

Moveit:

Moveit basically render the trajectories for the robotic arm (in this thesis) that situate the end effector of the robot in the desired place. It is a non-trivial task due to the sequence of joints general coordinates (position) with respect to the other joints. In short, Moveit facilitate the plan that the joints of a robot have to follow to move the end effector from its current position to the desired one.

Moreover, Moveit can execute the plan that has been obtained previously to the Robot Operating system (ROS) control.