Final Academic Internship Project (PFE)

Robotics & Embedded Systems
2020 - 2021

Development of a task-centered grasp analysis

Technical specifications for in-hand and dexterous manipulation

---

Confidentiality Notice
Confidential Report

Author:
Ricardo RICO URIBE

Promotion:
2021

Host Organism Tutor:
Mathieu GROSSARD

ENSTA Paris Tutor:
Thibault TORALBA

April 12 2021 - October 22 2021

CEA List
2 Boulevard Gobert
91120 Palaiseau - France

Confidentiality Notice

Acknowledgments

I had a hard time adapting to the work environment of this internship, I was burned-out because of 6 years of non-stop studying.

I was ready to quit, but in the end i didn't and I ended up having a great time working on this project. I wholeheartedly thank the CEA team and specially my boss/tutor Mathieu GROSSARD for allowing me to work remotely, He also helped me all grasping related questions, his knowledge and mathematical proficiency where fundamental for my understanding of the problem and development. I would like to extend my gratitude to Florian GOSSELIN for his analysis and for motivating me to improve my work.

I'm also grateful with my *enseignant referent* Thibault TORALBA for his help, his energy and helpfulness.

I cant forget my family and friends who where there for me and gave me the mental strength that I needed to work and make progress.

Thanks to everyone that accompanied me in this journey that hopefully will allow me to obtain my engineering and masters degree.

Abstract

This report provides you with the full analysis of the required force for the TRACEBOT manipulator. You will learn about grasping, how it was coded and how to use the code itself, to transform grasping information into real life force requirements. You can determine how to grab and object and realize in which direction you can resist the biggest perturbation force. I hope in the end you will understand why I am recommending 15 N and 25 N as the required force for the TRACEBOT manipulators, and how i got those results.

Key Words: Grasping, Force, TRACEBOT

Ce rapport vous fournit l'analyse complète de la force requise pour le manipulateur TRACEBOT. Vous apprendrez ce qu'est le grasping, comment il a été codé et comment utiliser le code lui-même, pour transformer les informations de préhension en exigences de force réelles. Vous pourrez déterminer comment saisir un objet et réaliser dans quelle direction vous pouvez résister à la plus grande force de perturbation. J'espère qu'à la fin, vous comprendrez pourquoi je recommande 15 N et 25 N comme force requise pour les manipulateurs du projet TRACEBOT, et comment j'ai obtenu ces résultats.

Mots Cles: Grasping, Force, TRACEBOT

Contents

Introduction

1.1 Context

The code and tools I developed in this internship were made with the intention of working for every application where grasping or task-analysis is needed. But the idea for the internship didn't appear from thin air, it was the TRACEBOT Robot that needed this analysis and for which I made the analysis of the tasks that will be presented in this report.

Because the focus of the results and analysis is the TRACEBOT project we need to give its context. Paraphrasing from the website [1], TRACEBOT is a European union project with the intent to create a traceable robotic handling of sterile medical products, a robot composed of two arms and a vision system capable of manipulating different objects, the objective is to lower costs and reduce time of the production of sterile medical products.

The size and reach of the TRACEBOT project is so big that the development was divided and subdivided in teams all across Europe, I was part of the Atomic Energy Commissary (CEA) team, and my role was to provide the force needed to perform tasks. For instance the medical procedure is to take a bottle and hold it in the air, my job was to find the force necessary for the bottle to not slip and fall to the ground, this force is the force at the finger tips holding the bottle, that later will be used to find the torque necessary for the motors moving the fingers.

1.2 Structure

This report presents my work in two parts, the first one is the creation of a general code that allows for grasping and force analysis that the Robotic Architecture Laboratory at CEA can use for any project, and the results and analysis of the information produced by the code for the design of the manipulators the CEA is in charge for the overall TRACEBOT project.

Because of the nature of the internship, almost all the time was dedicated to software development, for this you will also have at your disposal the description and examples on how to run the code.

The most important remark is that **task-centered** means that the analysis is focused in the robot ability to perform different (**static**) tasks, but for this type of analysis there isn't a robot architecture, and in some aspects, there is no need for objects, the information allows to find the technical needs to design a robotic hand.

You will find the simplified theoretical background needed to understand *grasping* (how a robot grasps/holds and object), my development of the tools for the analysis of the grasping and the task-oriented results found for the TRACEBOT project.

2 Theory: Grasping

This section is dedicated to our main mathematical tool, the grasp matrix G and the greater Grasping Context, I relied heavily on the *Springer Handbook of Robotics*[4], mainly the chapter of *Grasping*. But for the reader I have summarized and ordered the knowledge needed to understand Grasping, feel free to expand on the information by using the chapter directly. The notation used in this section and the project in general is the one used in the chapter.

2.1 Definition

In short, the concept of grasping refers to the action of grasp (take hold) of an object,

If we want to perform tasks, grasping can help us find the necessary forces and later the necessary torques via the grasping Jacobian. But we are getting ahead of ourselves.

Grasping is represented trough the grasp matrix G.

What is G?

Assume we have a rigid body *(object)* in a world with a frame $\{N\}$ described by $\{Ow, X, Y, Z\}$, the object has its own frame $\{B\}$: $\{p, x, y, z\}$ and the object is being grasp at various points throughout its surface, at each contact point $\{Ci\}_{i:1...nc}$ there is a tangent plane, that allows us to describe each contact point with its own frame, the normal vector to the tangent plane, points to the interior of the object and it's the X equivalent of each C_i, which leaves us with $\{c_i, n_i, o_i, t_i\}$ as the frame of each contact point. (The contact points are described in $\{N\}$).

At last, consider g as the sum of all external efforts applied to the object and v the velocity of the object at point p both described in $\{N\}$. Figure 1 shows this configuration.

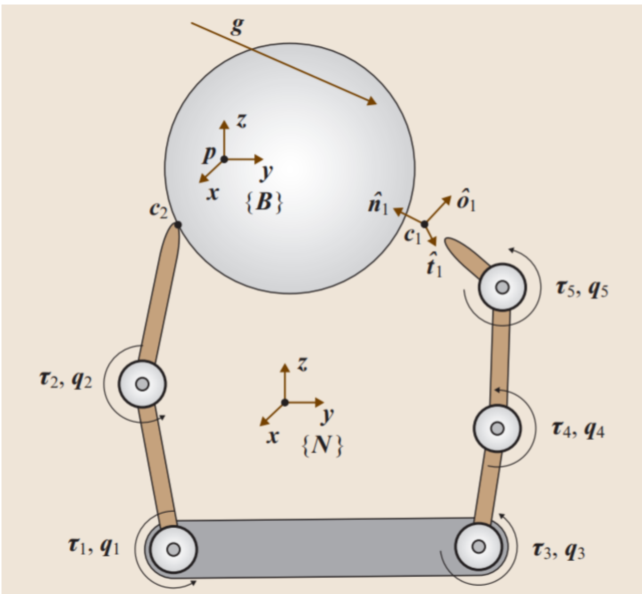


Figure 1: Main Notions for Grasping

In these conditions, the *complete* grasp matrix $\widetilde{G}$ links the velocities of each contact point expressed in its own frame to the velocity of the object expressed in the world frame, in the following way:

$$v_{contact} = \widetilde{G}^T v \tag{1}$$

Where $v_{contact} = [v_{contact_1,...,contact_{nc}}]$ is the concatenation of all contact points velocity vectors, expressed in the corresponding contact frame.

$\widetilde{G}^T$ is obtained by the following calculations:

$$\widetilde{G}^T = \begin{pmatrix} \widetilde{G}_1^T \\ \vdots \\ \widetilde{G}_{nc}^T \end{pmatrix} \tag{2}$$

$$\widetilde{G}_i^T = \overline{R}_i^T P_i^T \tag{3}$$

Where: $\overline{R}_i = Blockdiag(R_i, R_i) = \begin{pmatrix} R_i & 0 \\ 0 & R_i \end{pmatrix}$, $P_i = \begin{pmatrix} I_{3x3} & 0 \\ S(ci-p) & I_{3x3} \end{pmatrix}$, $S(r) = \begin{pmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{pmatrix}$, and, R_i is the rotation of the contact frame $\{C_i\}$ in respect to $\{N\}$.

Previously we indicated that $\widetilde{G}$ is the *complete* grasp matrix, this is because $\widetilde{G}$ maps all the 6 contact velocities (3 linear and 3 angular) of each C_i to the object velocity, in practice the transmitted velocities heavily rely on the contact type, topic that will be addressed further down. For now lets indicate that:

$$G^T = H\widetilde{G}^T \tag{4}$$

where H is the component selection matrix, obtained with: $H = Blockdiag(H_1, \ldots, H_{nc})$ and $Hi = \begin{pmatrix} H_{iF} & 0 \\ 0 & H_{iM} \end{pmatrix}_{i:1\ldots nc}$

2.2 Contact Types

There are three main contact types in grasping, Point with-out Friction (PwoF),Hard Finger (HF),Soft Finger (SF). Each one more complex than the previous but with more components transmitted.

Point with-out Friction

This contact type its based on the assumption that the contact is produced in a infinitely small single point so there is no friction between the surface of the object and the finger. Where only the normal component (aligned with $\{n_i\}$) is transmitted.

Hard Finger

This contact type is also produced with the assumption of a single contact location, but this point has some tangential friction μ,so the three linear components are passed $\{n_i, o_i, t_i\}$

Soft Finger

This contact is the same as Hard Finger with the addition of torsional friction γ allowing for the transmission of momentum along the contact normal. We use

the following table to reference and construct H.

Model	li	H_{iF}	H_{iM}
PwoF	1	(1 0 0)	none
HF	3	I_{3x3}	none
SF	4	I_{3x3}	(1 0 0)

Table 1: Selection Matrices for Contact Models

Even-though some contact types use friction as a medium to transfer forces, at this point the grasp matrix doesn't use a friction model (and it never will, in the sense that the effects of friction don't appear inside of G), instead the friction appears when the concept of forces appear (explained in the quality metrics section).

Now we have a comprehensive grasp matrix $G \in R^{6xl}$,where $l = \sum_{i=1}^{nc} li$. Which allows us to transmit the selected velocities and momentums at each contact point. The grasp matrix will always have 6 rows, representing the 6 degrees of freedom of the object but even if we chose contact types that don't transfer momentum directly we can indirectly control them with a smart contact point location.
As a rule we always need $l >= \textit{degrees of freedom} + 1$ for the grasp to have the desired properties (to be "*valid*").
To know if the grasp that our matrix describes is valid we need to classify it, then, we can rank the grasps between them to find the best performing grasp, topics discussed in the following sections.

2.3 Classification

The chapter of grasping presents 2 different classifications for a grasp system that depend only on G.

Indeterminate

A grasping system is indeterminate if $N(G^T)$ is nonTrivial.
If a system is indeterminate it means that for an object with 6DoF there are some velocities and momentums that cannot be imprinted by the manipulator (there are internal object *twists* that cannot be controlled).

Graspable

A grasping system is graspable if $N(G)$ is nonTrivial.
All internal object *forces* are controlled, this means that we can augment the contact forces and the object will not move but instead improve the grasp if it is based on friction.

At this stage we know we want a grasp that is not indeterminate and that is also graspable.

2.4 Quality Metrics

We need to determine how good is G, that is why we have a quality metric that test for Frictional Form Closure (FFC) this metric is change in the basic Form Closure.

Form Closure can exist with any contact type, but in its most basic form, it uses PwoF and uses one contact to restrict each DoF of the object plus an additional contact to obtain full form closure. Now as this approach requires a lot of contacts and uses a very primitive contact model, that is why FFC was developed.

Frictional Form Closure

FFC reduces the amount of contacts by using the friction to restrict some of the DoF of the object. But this approach encompasses more challenges, now it is necessary a friction model, the book recommends the Coulomb friction Cone, that needs to be faceted fig.2 to work with a Linear Programming (LP) Problem as shown below.

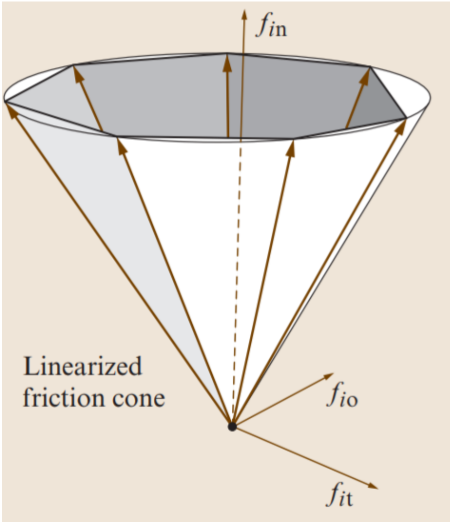


Figure 2: Faceted Friction Cone

$$
\begin{array}{lrl}
\max & d & \\
\text{s.t.} & G\lambda = & 0 \\
& F\lambda - \ Id \geq & 0 \\
& d \geq & 0 \\
& e\lambda \leq & nc
\end{array} \tag{5}
$$

The LP shown in (5) is the problem of FFC where, we are trying to *maximize* d, a distance that can be interpreted as how far is the grasp from losing FFC, and rank the grasps by which one has a bigger d.

To solve 5 we are missing some definitions.

$$F = Blockdiag(F_1, \dots, F_{nc})_{i:1,\dots,nc}$$

$$F_i = \begin{pmatrix} S_i[col_1] \times S_i[col_2] \\ \vdots \\ S_i[col_k] \times S_i[col_{k+1}] \\ \vdots \\ S_i[col_{ng}] \times S_i[col_1] \end{pmatrix} \qquad S_i = \begin{pmatrix} \dots & 1 & \dots \\ \dots & \mu_i \cos(2k\pi/ng) & \dots \\ \dots & \mu_i \sin(2k\pi/ng) & \dots \end{pmatrix}$$

S_i is the friction cone generator matrix for contact point i, F_i is the faceted friction cone and F is the recollection of all friction cones for the nc points.

Lastly, $e = (e_1, \dots, e_{nc})$, with e_i being the first row of H_i.

The definition of this quality metric was important for the development of a more useful metric, the following metric was proposed by ***Mathieu GROSSARD*** by combining metrics appearing in [3] and [5]. The metric was first used in[2], and now we are going to use it.

Task-Oriented Quality Metric (Alpha Quality Metric)

As previously stated g is the sum of all external efforts, that means that g can be described as $g = \alpha * d_{W_{ext}}$, with this form of g we can analyze any desired direction to obtain the maximum magnitude the grasp can handle before slipping (failing).

$\alpha$is a number with Newton (N) as units, but g has to have units of Force and Moment, $g \in R^{6x1}$ *with units [N N N Nm Nm Nm]* the author of [2] helped me with the definition for $d_{W_{ext}}$ to allow the LP to find a correct α. The steps to define $d_{W_{ext}}$ before the problem are:

1. Place a 1 in the position for the component where you wish to analyse the direction. remember [X Y Z mX mY mZ], this directions can be negatives and combined with each other to find the$\alpha$for the combined direction.

2. Normalize the first 3 and last 3 components independently (force separated from moment).

3. multiply the last 3 components (moment) by the object's characteristic length.

What is and object characteristic length ?

Is whatever dimension or sub-dimension of the object you want, in my case I choose the worst possible scenario, where the length was the maximal distance between the center of mass of the object and any other point. I choose this because the characteristic length is used to calculate moments, so i wanted to know the α needed to counteract a torque with maximum arm.

We now should have a $d_{W_{ext}} \in R^{6x1}$ *with units [1 1 1 m m m]*, and all other necessary elements to find α (and the contact forces associated with that α)

The resulting LP problem is the following:

$$
\begin{array}{llll}
\max_{(\alpha, f_c)} & \alpha & & \\
\text{s.t} & G f_c + \alpha d_{W_{ext}} = 0 & & \text{Static Equilibrium} \\
 & F f_c \geq 0 & & \text{Friction Cone} \\
\hline
\text{bounds} & \alpha \geq 0 & & \\
 & 0 \leq f_{c_n} \leq f_{c_{max}} & & \text{Actuator Limitations (HF)}
\end{array} \tag{6}
$$

The static equilibrium refers to the ability to maintain the grasp without any movement from the object, the friction cone is the constraint that ensures that the forces are related between them via the friction coefficient (the tangential forces are a small part of the normal force); and finally the actuator limitations refers to a real life limitation, if not for this bound the problem would suggest an infinite normal force, also the normal force is bound to be positive because if negative the finger or contact point would un-attach itself from the surface of the object

In general, α is used when the perturbation is unknown, because α needs a direction of analysis and maximal possible force, we can find α for all possible directions and $f_{max} = 1$, this will give us a table that indicates for each grasp, with maximal contact force of 1 N what is the maximal force resisted before slipping, and because of the linear nature of the problem, if we increase the maximal force by a factor of 2, then, α will increase by the same factor.

This approach of unknown perturbation but all possible directions was used to determine the force needed for a known perturbation (explained in the results section)

2.5 Jacobian

Throughout this entire section we have neglected the jacobian, if you are familiar with robotics you would know this is a very common concept, and is tightly linked to a robot architecture description. The jacobian matrix links each robot joint velocity to the end-effector velocities.

But in the case of Grasping we have a modified Jacobian, because we are not analysing the end-effector in it-self but rather and object being contacted by the end-effector, and more importantly, we have more than one end-effector, because for the manipulation we have to consider multiple fingers of a single manipulator.

Because of this we have the following mathematical description for the Grasp Jacobian, called inside this report just as jacobian $J \in R^{6 \times nq}$.

As interesting as the Jacobian is, my results had no need for it. This caused a duality, where I fully developed the code needed for the Grasp Jacobian as I already have the knowledge needed but it I didn't used because our approach didn't have a specific architecture for the robot or a joint configuration.

3 Work Done

This section presents all the material, results and code created during the internship period. You will find the structure of the provided elements, the order in which they were produced, then an in-depth on the code, the workflow to generate data and finally the results and their analysis.

3.1 Structure

The main folder, that contains all of the project is called *Task Oriented Grasp Analysis* this folder (repository) is found on the gitlab server of the laboratory. in there you will have access to the code, the excel presenting the data, the stl objects, the slides created throughout the duration of the internship and this report.

```
TASK ORIENTED GRASP ANALYSIS
├─ code
├─ excel
│   ├─ excel file
│   └─ images
├─ report
├─ slides
└─ stl
```

```
code
├─ config_files
│   ├─ examples
│   ├─ update_config
│   ├─ object description
│   └─ config
├─ excel_data_generation
│   └─ generation files
└─ grasp
    ├─ examples
    ├─ class_stl
    ├─ class_grasp
    ├─ class_jacobian
    ├─ data_types
    ├─ grasp_functions
    └─ quality_metrics
```

Inside the code folder you will find the Python code created, it contains three different types of code, the most important being *grasp*, it has all that is needed to solve the previously mention grasping theory. The other two types are essential for the data generation pertaining the TRACEBOT Project, the *config_files* transforms the object description file into an usable input for the final type of code, the *excel_data_generation*. It uses the object description as input and produces Excel tables as output.

The other folders are self explanatory, the excel folder contains the data generated to be analysed in a later section and *png* files showing each type of input data. The report and slides folder, as their name suggest contain this report and the slides respectively, and finally the stl folder contains the stl files created for the analysis of the TRACEBOT Objects

3.2 Code basis

Lets keep in mind the objective, to find the forces needed to resist the perturbation and produce the requirements for the grippers. To accomplish this we crafted several functions that access our output file (an excel) and connected all the other code between them to get and object, a grasp, a perturbation, determine their α for all the directions, and then with the known perturbation find the required force. This is part of the next section, which explains the workflow required to obtain the results, for know lets dive into the code.

We will use fig 3 as the focus point, and go from there.

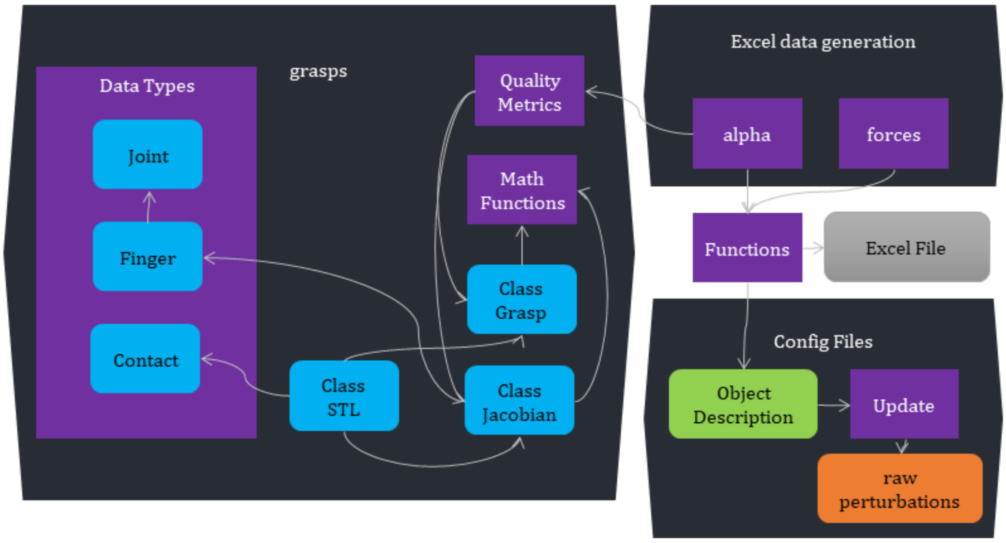


Figure 3: Relationships between the code

Blue represents classes (code written with Object Oriented Programming in mind), Purple is regular python scripts, and gray, green and orange represents excel, yaml and txt files respectively.

The choice from the different types is very simple, excel was chosen for the readability and data presentation, txt was chosen because of the simpleness in its formatting and yaml was chosen for its human-readability and ease in modification. *I'm certain there is a different solution but I'm are happy with our workflow and results*

This report is not meant to be the code documentation, that's why you wont find all the functions explained or code extracts, that information is found within the code itself ,nevertheless the next subsection can give you an inside look into the code.

3.3 Work Flow for the data generation

In here will be a full example on how the force information was produced, we will have no variations to simplify the exercise, and a TRACEBOT object will be used to give you a sense into the results and their later analysis.

The object of the exercise will be a **Petri dish**, the grasp will be the **C12**, the direction of analysis will be **-Z**, and the perturbation will be its own **weight**.

The fig 4 shows the general flow to generate force data.

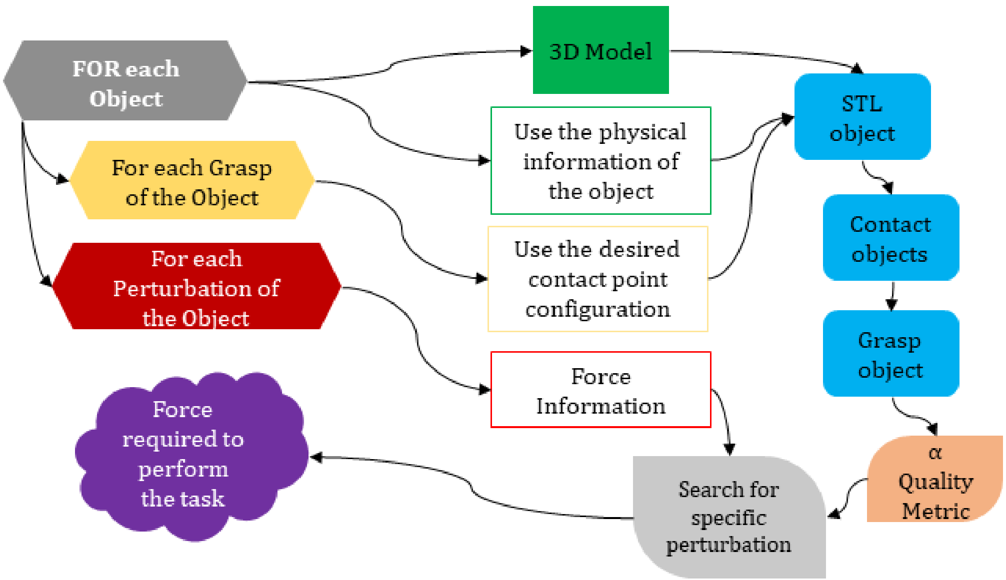


Figure 4: Flow to generate data

3.3.1 3D Model

First we create a 3D model of the object in a modeling software and export it as an stl file (shown in fig5).

Then we can use the stl class python file to use the stl object as a python object. The class provides you with a function that allows to view the object inside of python. Shown in the code below and the resulting image in figure 6.

```
mesh = STL("PATH TO STL FILE", [3D POSITION OF CENTER OF MASS])
mesh.view("FIGURE NAME")
```

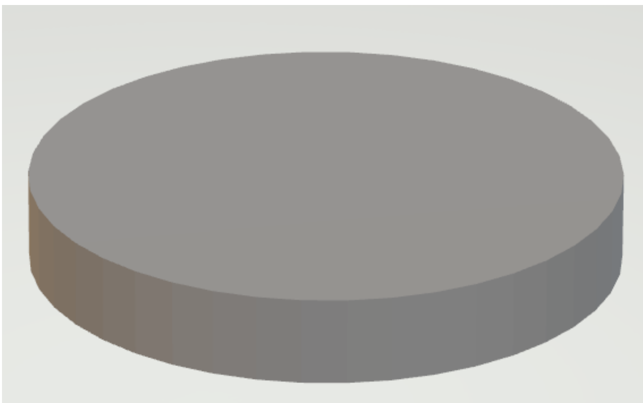


Figure 5: stl file of the Petri

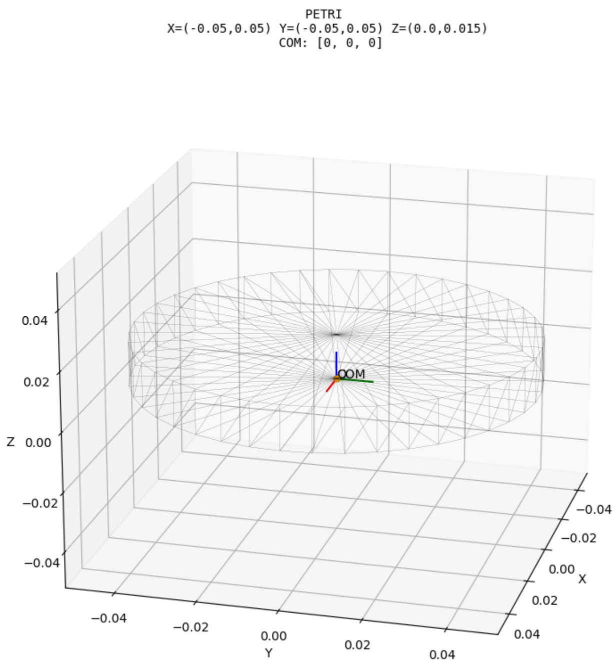


Figure 6: Mesh view of the object inside of Python

3.3.2 Grasp Declaration

Now we have to declare a Grasp for the object, remember that a grasp is a way to hold the object, so we took a human approach *(all grasps are from the TRACEBOT videos and were defined by an other intern)*. With the grasp information (fig7) we can then use a combination of the stl class and the yaml file to produce a object of the type Grasp Class, and then with the stl and grasp class view the contact points (fig8) and the grasp matrix (fig9) respectively.

Object	name	hand contacts	Reference photo
Petri	C12		

Figure 7: Human taxonomy of the C12 Grasp applied to the Petri

From the taxonomy we can build the contact points locations as a yaml file.
each contact location has a name according to the naming convention created by the other intern

```yaml
Petri:
    C12:
          1D:
            x: 0.05
            y: 0.0
            z: 0.01
          2D:
            x: -0.035
            y: 0.035
            z: 0.01
          3D:
            x: -0.05
            y: 0.015
            z: 0.01
          4D:
            x: -0.05
            y: -0.015
            z: 0.01
          5D:
            x: -0.035
            y: -0.035
            z: 0.01
```

Then we can use the stl class to generate the complete contact points to create the grasp object.

```python
C1 = mesh.contact_from_point([yaml C12 1D])
C2 = mesh.contact_from_point([yaml C12 2D])
C3 = mesh.contact_from_point([yaml C12 3D])
C4 = mesh.contact_from_point([yaml C12 4D])
```

```
C5 = mesh.contact_from_point([yaml C12 5D])
mesh.view(contacts=[C1, C2, C3, C4, C5]) #fig\ref{fig:petri-c12}
grasp = Grasp(p, C)
print(grasp.Gt)
grasp.get_classification(True) #fig\ref{fig:matrix}
```

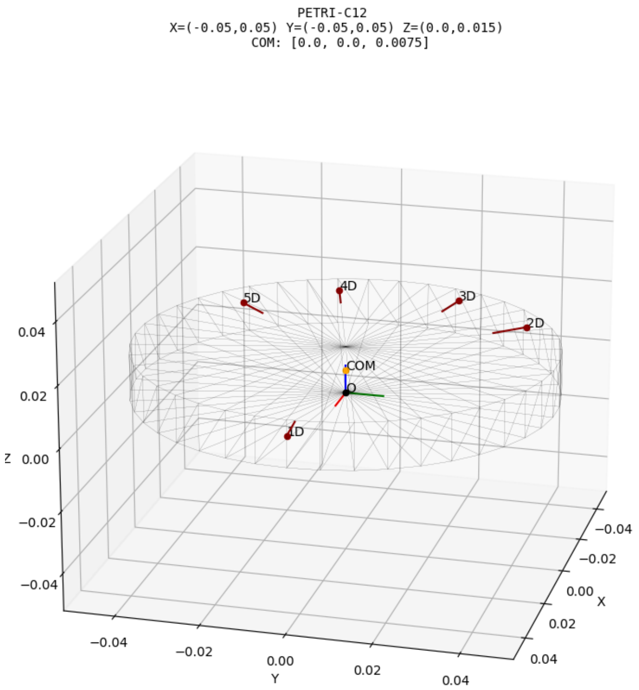


Figure 8: Mesh view of the Petri-C12 configuration

```
Gt
 [[-0.997 -0.075  0.     0.    -0.002 -0.004]
 [ 0.033 -0.445  0.895  0.001 -0.045 -0.022]
 [-0.067  0.893  0.446 -0.002 -0.022  0.045]
 [ 0.     0.     0.    -0.997 -0.075  0.   ]
 [ 0.68  -0.733  0.     0.002  0.002  0.002]
 [ 0.327  0.303  0.895  0.031  0.032 -0.022]
 [-0.656 -0.609  0.446  0.017  0.014  0.044]
 [ 0.     0.     0.     0.68  -0.733  0.   ]
 [ 0.975 -0.223  0.     0.001  0.002 -0.003]
 [ 0.099  0.435  0.895  0.012  0.045 -0.023]
 [-0.199 -0.873  0.446  0.009  0.022  0.047]
 [ 0.     0.     0.     0.975 -0.223  0.   ]
 [ 0.956  0.295  0.    -0.001  0.002 -0.   ]
 [ 0.     0.    -1.     0.015 -0.05   0.   ]
 [-0.295  0.956  0.    -0.002 -0.001 -0.052]
 [ 0.     0.     0.     0.956  0.295  0.   ]
 [ 0.68   0.733  0.    -0.002  0.002 -0.002]
 [-0.327  0.303  0.895 -0.032  0.031 -0.022]
 [ 0.656 -0.609  0.446 -0.014  0.017  0.044]
 [ 0.     0.     0.     0.68   0.733  0.   ]]
-------------------------
GRASP CLASSIFICATION:
Nullspace(Gt): trivial --> Not Indeterminate
Nullspace(G): not trivial --> Graspable
```

Figure 9: Grasp information of the Petri-C12 configuration

3.3.3 Alpha

With a grasp, we can continue to the calculation of α as presented in the quality metric section. Remember, α needs a direction of analysis an a maximum possible force. For our exercise we will chose **-Z** as the direction and **1 N** as maximum force available. We can launch the code as shown below and obtain the result shown in the figure 10.

```
alpha, forces = alpha_from_direction(grasp, DIRECTION, MAX FORCE)
print(alpha, forces[0::3])  #print only the normal
                            #component of the contact forces
```

```
UNKNOWN PERTURBATION:
for dWext=[ 0.  0. -1.  0.  0.  0.] with Fmax= 1N
  max magnitude resisted=0.698
 required Normal Contact Forces=[1.    0.814 0.06  0.    0.788]
```

Figure 10: Alpha Magnitude and Normal Forces required for the Petri-C12 Configuration when a perturbation is along the *-Z* direction

3.3.4 Known Perturbation

The α metric is useful when we don't know the perturbation, because it gives us an analysis in any direction we want.
But when we know the perturbation we can calculate alpha for that direction and determine the necessary relation ship between alpha and the magnitude of the perturbation. Remember, the quality metric is linear, so if we increment the maximum force by 2, α will increase by 2. By using this property, we can determine the force required if we divide the perturbation magnitude by the α found with 1 N as *Fmax*.

For the exercise lets use the perturbation of its own weight in **-Z** (the weight is 15 grams transformed into N), the figure 11 shows how to write the perturbation in the Excel file.

Object	Distinct Events	W	dir	pos
Petri	hold	-0.147	Z	com

Figure 11: Perturbation declaration on Excel

com signifies center of mass, each perturbation has a magnitude, a direction and a point of application, the creation of the perturbation vector is easy in the cases where the forces don't generate any momentum. That's why below you will find directly the perturbation vector, nevertheless, there is a code that automatically transforms the generated raw *txt* into the forces for the *yaml* file.

```
#raw txt
{"Petri-hold":[[-0.147099750429584,"Z","com"]]}

#vector form:
[0, 0, -0.147, 0, 0, 0]
```

We can verify the perturbation declaration with the image generated as shown in fig 12

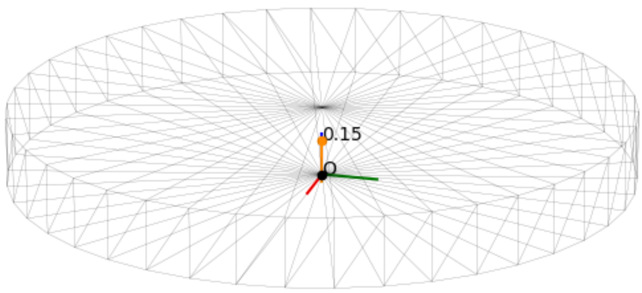


Figure 12: Mesh View of the perturbation

From this perturbation we can analyse α. Figure 10 shows that the magnitude resisted is 0.698 N and we only need to resist 0.147 N to resist the perturbation, that means we can decrease the maximal force to 0.211 N ($0.147/0.698 = 0.211$).

3.3.5 conclusion

We determine that, to grab the petri dish with a grasp C12 and resist its own weight in the -Z direction, we need a manipulator capable of applying 0.211 N at the finger tips normal to the surface.

3.4 TRACEBOT USE-CASE

Thanks to the previous section you understand what is required to produce the results, The workflow is the same and the number of variations is larger. Table 2 shows the variation amount of information analysed.

Objects	Grasps	Perturbations
Petri	C12, C6, C8, F28, T18, T2, T3, T4, T7, T8	HOLD, WRITE
Marker	C8, F26, F28, T10, T13, T16, T18, T9	HOLD, UNCAP, RECAP, WRITE
Marker Cap	C16, T17, T54	HOLD, UNCAP, RECAP
Kit	C11, C6, C7, C8, F28, T22, T35	HOLD, OPEN
Kit Tab	T21	HOLD, OPEN
Canister	C1, C6, C8, T18, T2, T26, T57	HOLD, INSERT, REMOVE
Tube	C2, C6, C7, C8, F17, T17, T23, T24, T26, T27, T28, T29, T30, T4, T70	HOLD, INSERT
Needle	C8, T21, T28, T33, T60	UNCAP, HOLD, PIERCE, UNPIERCE
Needle Cap	C14, T28, T4	UNCAP
Rinse Glass	C12, C6, T18, T2, T34, T35, T38, T39, T51, T58, T69	HOLD
Red Plug	F26, T21	HOLD, INSERT, REMOVE
Glass Vial	T10	HOLD, OPEN
Yellow Plug	T21	HOLD, INSERT
Tube Clamp	C16, T28, T65	HOLD, CLAMP, UNCLAMP
Scissors	C16, C8, T68, T68_	HOLD, CUT

Table 2: TRACEBOT USE-CASE

All perturbations named "HOLD" are in the 6 linear directions, to analyze all the different ways to hold the object, the other perturbations have a specific direction.

Further down you will find how the perturbations were measured, for now lets focus on the variations in the grasps and the directions for α.

3.4.1 Grippers

Table 3 shows the different gripper configurations. Remember, the objective is to give the technical specifications for the manipulator, but for this we have to consider what will be our future manipulator capable to do.

Contacts	Fingers	mu f.coeff	Graph Code	Description	Grasp Name
All	5	0.3	0	Original	0
		0.5	1		
D	4	0.3	2	Gripper 1	1
		0.5	3		
D,P	4	0.3	4	Gripper 2	2
		0.5	5		
D,Px,P	4	0.3	6	Gripper 3	3
		0.5	7		

Table 3: Different Gripper Capabilities

The grasps were originally designed as the TRACEBOT videos presented them (a human performing them) but for our grippers we decided to make some changes, for instance, only use 4 fingers and reduce the amount of possible contact locations in the hand. We also varied the friction coefficient to determine which perturbations can be resisted.

- D: Distal contact (furthest part from the palm)
- Px: Proximal contact (closest part to the palm)
- P: Palm can make contact

The idea behind this division of grippers is to have a low force gripper (the number 1) which can manipulate with its fingers, and have the other manipulator perform *power grasps*. We were unsure about how many possible contacts for the powerful gripper, that's why the data was analysed for gripper 2 and 3 (they are mutually exclusive)

3.4.2 Grasp analysis

With all grasps and grippers defined we can analyse the first output, the grasp classification and grasping properties. The figure 13 shows the first three rows of the table created **(all the information is available on the sheet named *GRASP* on the excel file)**

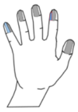
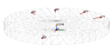
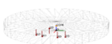
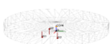
					nc				rank(G)				Indeterminate				Graspable				Friction Force Closure			
Object	name	hand contacts	Reference photo	contact location	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
Petri	C12				5	4	4	4	6	6	6	6	False	False	False	False	True	True	True	True	True	True	True	True
Petri	C6				5	4	4	4	6	6	6	6	False	False	False	False	True	True	True	True	True	True	True	True
Petri	C8				3	3	3	3	6	6	6	6	False	False	False	False	True	True	True	True	True	True	True	True

Figure 13: Grasp Analysis Sample

From the table we can analyse the validity of the grasps, the first column with generated information is the comparison between the amount of contact points, as previously stated the grippers differ from the original grasp conception in the amount of contact location and the amount of fingers used, but for some grasps, the original is equivalent to the robotics. Then we have Indeterminate and Graspable, we are looking for grasps that is both not indeterminate and graspable, to be considered valid. The final column is the quality metric, friction form closure, it means that the object can be held securely with the friction effects, now we pass to alpha to determine how much is the magnitude that can be resisted.

3.4.3 Alpha analysis

In the case of α we analysed all the combinations between x, y, z for the linear directions; for the moment directions, we analysed each axis independently and their negative directions. **(Alpha directions: X+Y+Z,X+Y-Z,X+Y,X-Y+Z,X-Y-Z,X-Y,X+Z,X-Z,X,-X+Y+Z,-X+Y-Z,-X+Y,-X-Y+Z,-X-Y-Z,-X-Y,-X+Z,-X-Z,-X,Y+Z,Y-Z,Y,-Y+Z,-Y-Z,-Y,Z,-Z,mX,-mX,mY,-mY,mZ,-mZ)**

With the α information we can create a polygon indicating the direction preference to aid in future path planing, figure 14 shows the polygon for the exercise configuration.

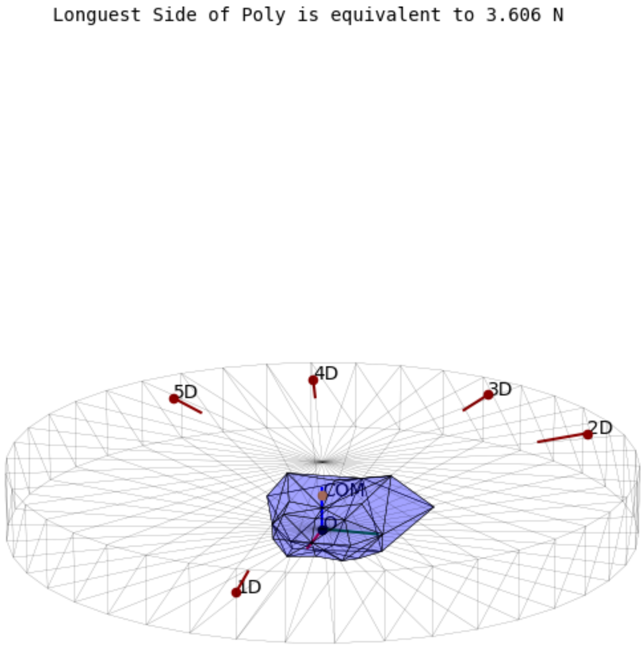


Figure 14: Alpha polygon of the Petri-C12 original configuration

Another way to visualise the data is with table on the excel file (sheet ***DIR RANKING***), figure 15 shows a sample of the table, as the sheet name name implies this table is useful to determine if a direction is better than other (if the α magnitude is greater).

Object	name	contact location	X+Y+Z				X+Y-Z				X+Y				X-Y+Z				X-Y-Z				X-Y				X+Z				X-Z				X			
			0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
Petri	C12		20	21	19	21	24	24	24	24	15	19	15	19	18	20	18	20	23	23	23	23	14	18	14	18	25	25	25	25	26	26	26	26	12	22	12	22
Petri	C6		18	14	22	21	6	8	4	8	11	14	14	13	18	14	17	16	6	8	4	8	11	14	12	12	13	14	14	15	1	3	1	1	2	1	6	2
Petri	C8		17	17	17	17	9	9	9	9	14	14	14	14	17	17	17	17	9	9	9	9	14	14	14	14	8	8	8	8	3	3	3	3	1	1	1	1
Petri	F20		19	21	19	21	24	24	24	24	15	19	15	19	18	20	18	20	23	23	23	23	13	18	13	18	25	25	25	25	26	26	26	26	12	22	12	22
Petri	T10		17	17	2	4	6	6	2	4	12	12	2	4	23	22	2	4	8	8	2	4	14	14	2	4	25	24	2	4	10	10	2	4	19	19	2	4
Petri	T2		12	14	22	21	9	8	4	8	17	16	14	13	13	15	17	16	9	8	4	8	17	16	12	12	6	7	14	15	3	2	1	1	1	1	6	2
Petri	T3		9	14	6	7	6	5	5	6	3	3	1	1	18	18	15	16	15	13	15	15	16	9	15	5	13	19	12	14	11	10	11	13	10	7	2	2
Petri	T4		4	6	4	6	6	19	6	19	6	12	6	12	4	6	4	6	6	19	6	19	6	12	6	12	1	1	1	1	6	18	6	18	6	10	6	10

Object	name	contact location	-X+Y+Z				-X+Y-Z				-X+Y				-X-Y+Z				-X-Y-Z				-X-Y				-X+Z				-X-Z				-X			
			0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
Petri	C12		5	4	4	3	8	7	6	5	2	2	2	2	7	6	7	9	10	10	11	11	3	3	5	8	9	8	8	6	11	9	10	7	1	1	1	1
Petri	C6		22	14	25	25	8	4	11	3	15	14	21	17	22	14	20	22	8	4	7	3	15	14	13	14	14	14	18	20	3	6	2	6	5	2	10	5
Petri	C8		21	21	21	21	6	6	6	6	19	19	19	19	21	21	21	21	6	6	6	6	19	19	19	19	13	13	13	13	4	4	4	4	2	2	2	2
Petri	F20		5	3	5	3	9	8	9	8	2	2	2	2	6	6	6	6	10	10	10	10	3	4	3	4	8	5	8	5	11	9	11	9	1	1	1	1
Petri	T10		17	17	2	4	6	6	2	4	12	12	2	4	23	22	2	4	8	8	2	4	14	14	2	4	25	24	2	4	10	10	2	4	19	19	2	4
Petri	T2		16	18	25	25	7	3	11	3	14	12	21	17	23	19	20	22	7	3	7	3	15	24	13	14	23	13	18	20	4	5	2	6	1	24	10	5
Petri	T3		7	8	8	10	4	4	10	11	2	2	4	4	25	26	15	25	22	17	15	24	26	24	15	26	21	21	15	18	14	11	15	17	20	15	15	12
Petri	T4		6	2	6	2	6	23	6	23	6	14	6	14	6	2	6	2	6	23	6	23	6	14	6	14	2	4	2	4	6	21	6	21	6	11	6	11

Object	name	contact location	Y+Z				Y-Z				Y				-Y+Z				-Y-Z				-Y				Z				-Z			
			0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
Petri	C12		13	12	13	10	16	13	16	12	4	5	3	4	17	14	17	14	19	16	20	17	6	11	9	13	22	17	22	16	21	15	21	15
Petri	C6		24	14	26	26	10	10	8	10	20	12	24	19	24	14	23	23	26	10	8	10	20	12	16	18	17	14	19	24	4	7	3	7
Petri	C8		25	23	25	23	11	11	11	11	23	25	23	25	25	23	25	23	11	11	11	11	23	25	23	25	16	16	16	16	5	5	5	5
Petri	F20		14	12	14	12	16	13	16	13	4	7	4	7	17	14	17	14	20	16	20	16	7	11	7	11	22	17	22	17	21	15	21	15
Petri	T10		16	16	2	4	2	1	2	1	4	4	2	4	21	21	2	4	1	2	2	2	5	5	2	4	22	26	2	4	3	3	1	3
Petri	T2		20	21	26	26	23	10	8	10	21	22	24	19	19	24	23	23	11	10	8	10	21	23	16	18	23	20	19	24	5	6	3	7
Petri	T3		8	12	7	8	5	6	9	9	1	1	3	2	23	23	15	22	19	20	15	21	24	22	15	23	17	25	14	20	12	16	13	19
Petri	T4		6	8	6	8	6	25	6	25	6	16	6	16	6	8	6	8	6	25	6	25	6	16	6	16	3	5	3	5	6	22	6	22

Figure 15: Direction and Alpha Analysis Sample

This table is a ranking between all the linear directions and the four grippers for each grasp, so is not recommended to compare the grasps between them because the value of the best direction could be in a different order of magnitude. We can also compare the grippers between them because it allows you to visually know if a gripper configuration changed the original direction preference.

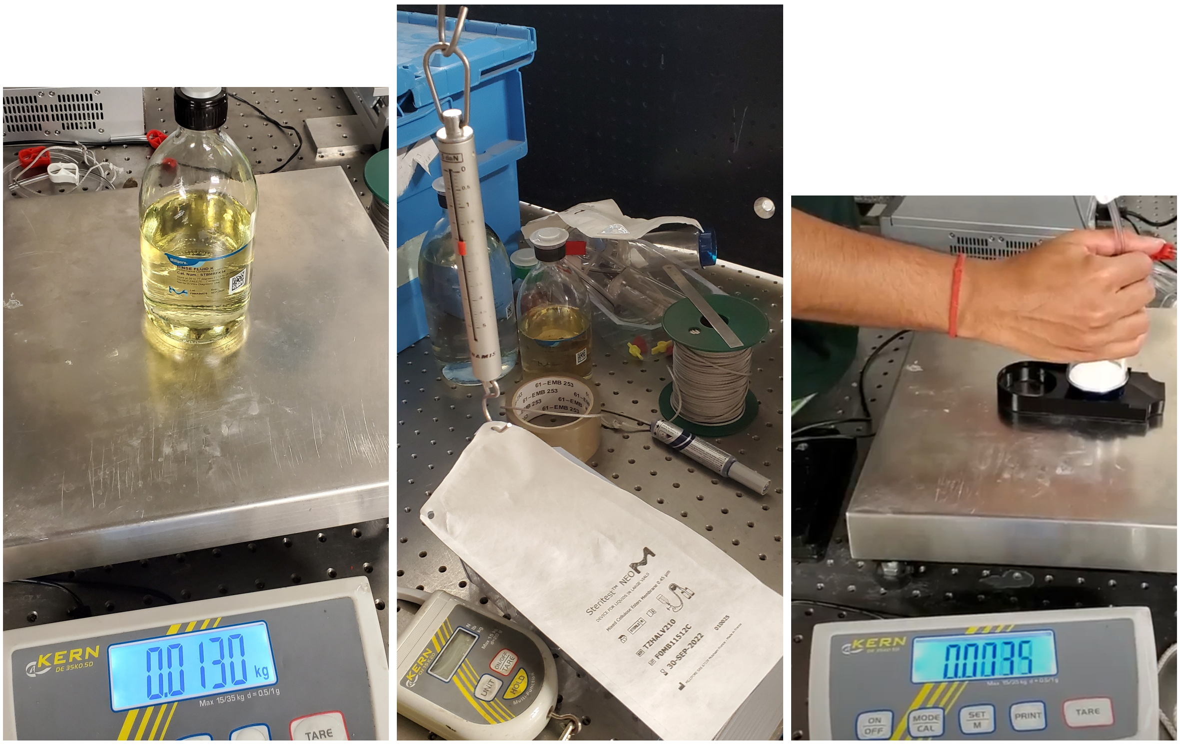
Now that we have a table of all the values of α for all the directions indicated and for each gripper configuration and each grasp, we can determine the force required for each perturbation.

3.4.4 Known Perturbations

Because the project didn't provided us with the force information, we measured ourselves the forces required to perform each identified tasks (perturbations on the objects).

We measured experimentally the force of each task, and categorized them in three groups: weight, pulling and pushing. Weight was measured with a scale (fig16 (a)), pulling was measured with the peak force in a dynamometer (force measuring tool) (fig16 (b)) and finally pushing that was the most tricky to measure. Pushing was measured on the scale by performing the task and taking the force of the maximum weight registered (fig16 (c)).

The rest of the images and data taken from the experiments are available in the repository (report folder) and on the excel file respectively.



(a) (b) (c)

Figure 16: Measuring (a) Weight (b) Pulling (c) Pushing

3.4.5 Results

With the grasps defined and valid, with the alpha tables and the known perturbations, several force data was collected, each one comparing a force vs a different sub group of the project, On the excel file we have the sheets shown on figure 17 that group the data onto their corresponding name.

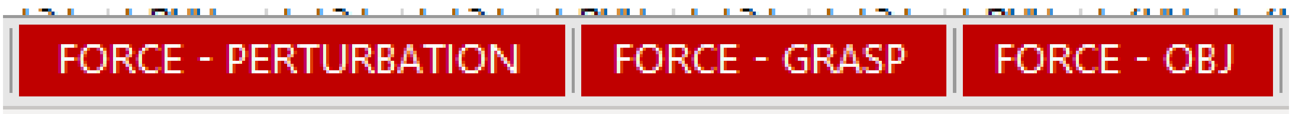


Figure 17: Excel Sheets that group the Force information

We will focus on the perturbation grouping. This sheet analyses each perturbation and provides 2 sets of information, the minimum required force to perform the task (resist the perturbation) (this will be the object's grasp with the minimum requirement) and the force where all the grasps are possible. (fig 18)

FORCE	IMAGE	DESCRIPTION	VECTOR	MINIMUM REQUIRED								ALL GRASPS POSSIBLE								
				0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	LIMIT 20 N
Marker_Cap-hold_Y		[[0.022, 'W', 'com']]	[0.0, 0.022, 0.0, 0.0, 0.0, 0.0]	0.022	0.016	0.022	0.018	0.022	0.016	0.022	0.016	0.061	0.037	0.061	0.037	0.061	0.037	0.061	0.037	20
Marker_Cap-hold_X		[[0.022, 'W', 'com']]	[0.022, 0.0, 0.0, 0.0, 0.0, 0.0]	0.023	0.023	0.061	0.03	0.023	0.023	0.023	0.023	0.061	0.037	0.061	0.037	0.061	0.037	0.061	0.037	20
Red_Plug-hold_X		[[-0.007, 'W', 'com']]	[0.007, 0.0, 0.0, 0.0, 0.0, 0.0]	0.023	0.018	0.023	0.018	0.023	0.018	0.023	0.018	0.029	0.019	0.029	0.019	0.029	0.019	0.029	0.019	20
Marker_Cap-hold_-Y		[[0.022, 'W', 'com']]	[0.0, -0.022, 0.0, 0.0, 0.0, 0.0]	0.024	0.018	0.061	0.037	0.024	0.018	0.024	0.018	0.061	0.037	0.061	0.077	0.061	0.037	0.061	0.037	20
Marker-hold_-X		[[0.098, 'W', 'com']]	[-0.098, 0.0, 0.0, 0.0, 0.0, 0.0]	0.028	0.025	0.068	0.065	0.062	0.061	0.068	0.065	0.312	3.065	0.312	0.19	0.312	3.065	0.312	0.19	20
Red_Plug-hold_-X		[[-0.007, 'W', 'com']]	[-0.007, 0.0, 0.0, 0.0, 0.0, 0.0]	0.029	0.013	0.029	0.013	0.029	0.013	0.029	0.013	0.175	0.033	0.175	0.033	0.175	0.033	0.175	0.033	20

Figure 18: Force - Perturbation table sample

With this table's information we can create the graph on fig 19, the graph compares serves to compare all grippers with a chosen limit of 20 N, (this limit was given by the engineer working on the construction of the manipulator), with this limit we can divide the tasks onto 2 grippers, the fine and low force one, and the *power grasp* one.

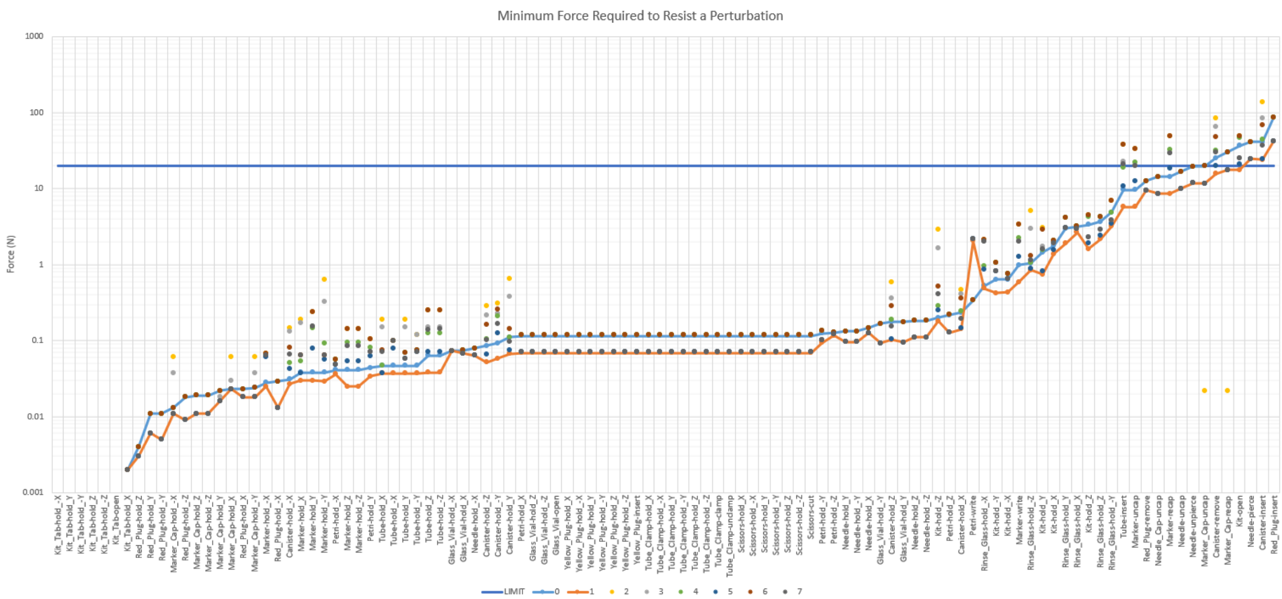


Figure 19: Force - Perturbation Graph

There is a lot of information to work with, so lets focus on the tasks that are close to the limit, and the best original gripper, the worst low force gripper and the best power grasp gripper. (best and worst are qualitative but they refer to the coefficient of friction, 0.3 is worse compared to 0.5)

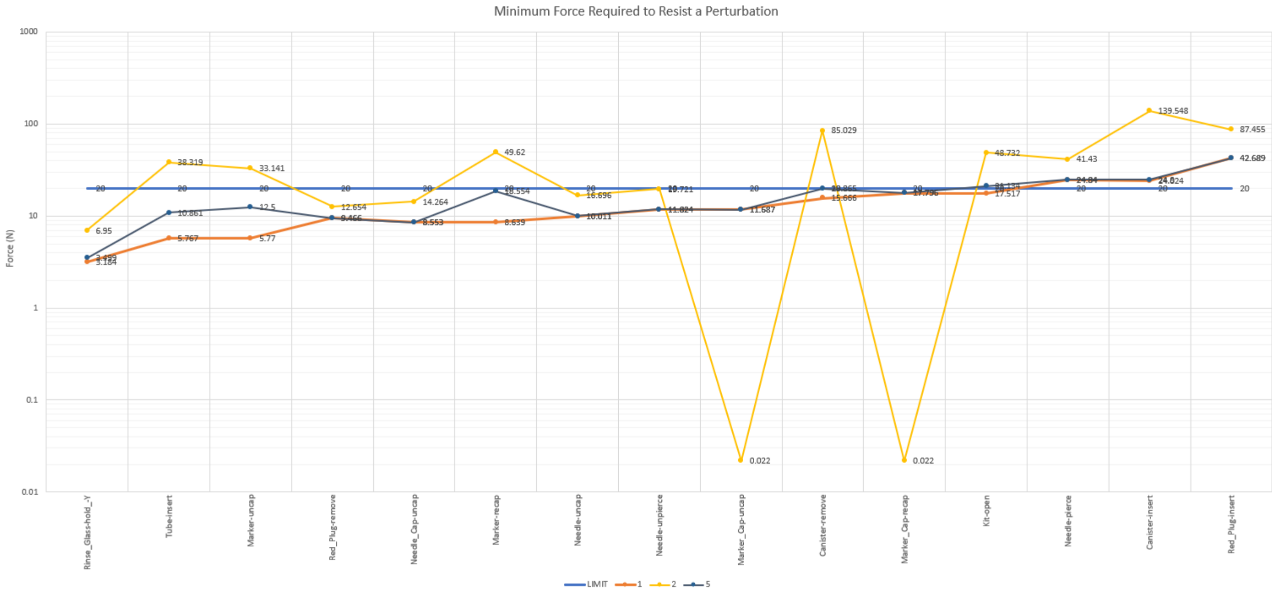


Figure 20: Force - Perturbation Graph *Focused*

From the figure 20 we can immediately extract some conclusions.

- There are 3 task we can't accomplish if both of our grippers have a limit of 20 N, we should consider that the power grasp gripper have a 45 N capability to perform all tasks with limited grasps or just modify how the task is performed.
- The low force gripper can perform almost all tasks, and there is even some tasks it performs better than the theoretical allowed, this is because, after the modifications the grasps performing this tasks became specialized, so they outperform the original 5 finger grasp.
- The power grasp gripper will perform the tasks of: Tube Insert, Marker uncap, Marker recap, Canister remove, Kit open, Needle pierce, Canister insert and Red plug insert.

Something to consider is against what we are performing the task, and in this case we are lucky, because only one of the tasks that the power grasp manipulator will perform is two handed, this one being the Kit Open task (as its name suggest is the action of opening the kit, but what is holding the kit down while we pull from its tab), in this case we can consider a modification to the table where the kit can be locked mechanically and the power grasp hand pulls the tab with a hook. (I mention a hook because the graph 18 shows that no grasp is able to resist the perturbation on the kit tab itself)

The other tasks are self explanatory, but I will gloss over them.

- Tube Insert is the action of inserting the tube into the canister at the end of the process.
- Marker cap and uncap is the action of putting on and off the cap (can be removed if the marker is button actioned)
- Canister remove and insert is to put and remove the canister from its holder (which is bolted to the table)
- Needle pierce is the action of inserting the needle onto the rinse glass (the rinse glass is on the table)
- Red plug insert is the action of plugging the top of the canister.

My recommendations for the technical requirements can be found on the next section.

Recommendation

3.5 Conclusion

As a reader you may have different recommendations or different conclusions, and that is great, I'm far from being an expert in the field but after working 6 months in the project I wouldn't take my conclusions with a grain of salt.

- There should be 2 grippers, one capable of low force manipulation and another with high force capabilities.
- The material of the grippers should be a silicon based plastic, or something with a high coefficient of friction.
- **The Low force gripper should have at least 15 N of possible contact normal force, and the power grasp gripper 25 N.**

There is the possibility to reduce the amount of tasks with clever modifications to the table of by changing the grasp or the force applying component, for instance, for the insertion of the red plug, we can position the plug with the low force gripper and push it down with the palm of the power grasp gripper (without gripping it *per se*)

Keep in mind that the results were obtained with a Hard Finger, when in real life the robot and the human are soft finger, this means that we are analysing a worse possible scenario. Also, all our fingers are capable of applying the same force, which is better compared to the human hand and the results obtained.

3.6 Improvements

With a subject so large, 6 months weren't enough to do everything perfectly, code and mathematically wise everything the implementation is very good, but the data used could be better, for instance, the grasps could have been designed with a robotic manipulator in mind, and not a human hand. The perturbation experimentation could be better, with high refreshing scales and higher resolution tools.

I tried to improve the results by changing the model from Hard Finger to Soft Finger, but the mathematical methods were not available to code the Friction Cone for the Soft Finger model.

There is no appendix because the appendix is the whole repository, in there you can find all the images showing the objects, grasps, forces and polygons. You also have the stl files, the excel with all the code generated data and graphs presentation, and last but not least, the code which is reusable and was made with the best coding practices in mind.

References

[1] CORDIS. *Traceable Robotic Handling of Sterile Medical Products*. Apr. 2021. URL: https://cordis.europa.eu/project/id/101017089.

[2] Caroline Pascal. *Development of an object-centered grasp analysis and synthesis framework*. Nov. 2020.

[3] Máximo A. Roa and Raúl Suárez. "Grasp quality measures: review and performance". In: *Autonomous Robots* 38.1 (July 2014), pp. 65–88. DOI: 10.1007/s10514-014-9402-3.

[4] Bruno Siciliano and Oussama Khatib. *Springer Handbook of Robotics*. 8th ed. Springer Science & Business Media, May 2008. ISBN: 9783540239574.

[5] Ugo Vollhardt et al. "Energy-based stability analysis for grasp selection with compliant multi-fingered hands". In: *2019 18th European Control Conference (ECC)*. IEEE, June 2019. URL: http://dx.doi.org/10.23919/ecc.2019.8795792.

Acronyms & Notation

C_i Contact Point i. 7, 8

G Grasp Matrix. 7, 9

H Component Selection Matrix. 8

N Newton. 10, 11

N Nullspace of a Matrix. 9

R_i Rotation Matrix of Contact Point i. 8

γ Torsional friction. 8

μ Tangential friction. 8

$\widetilde{G}$ *Complete* Grasp Matrix. 8

$\{B\}$ Object Frame. 7

$\{N\}$ World Frame. 7, 8, 26

g Sum of External Forces at p expressed in $\{N\}$. 7, 10

p Origin of Object Frame. 7, 26

CEA Atomic Energy Commissary. 6

DoF Degrees of Freedom. 9

FFC Frictional Form Closure. 9, 10

HF Hard Finger. 8

LP Linear Programming. 9–11

nc Number of Contact Points. 7–10

ng Number of faces. 10

nq Number of joints. 11

PwoF Point with-out Friction. 8, 9

SF Soft Finger. 8

Glossary

Trivial A trivial system is one that has the vector zero as an answer. 9

List of Figures

List of Tables