Laboratoire d'Intégration des Systèmes et des Technologies

DIASI / Département Intelligence Ambiante et Systèmes Interactifs

DIASI/21-099/Rév. 0

21/10/2021

MECHANICAL DESIGN FOR TRACEBOT MANIPULATOR DEVELOPMENT OF A TASK-CENTERED GRASP ANALYSIS

RICO URIBE Ricardo

Commissariat à l'énergie atomique et aux énergies alternatives
Institut List | Département Intelligence Ambiante et Systèmes Interactifs
CEA Saclay - Nano-Innov - Bâtiment 861 - Point Courrier 173
91191 Gif-sur-Yvette Cedex
T. +33 (0)1 69 08 00 99
www-list.cea.fr

Établissement public à caractère industriel et commercial | RCS Paris B 775 685 019

TITRE :	**DEVELOPMENT OF A TASK-CENTERED GRASP ANALYSIS**	DIASI/21-099
AUTEUR :	**RICO URIBE RICARDO**	
UNITE :	**DIASI/LABO**	Page 2
PROJET :	**MECHANICAL DESIGN OF TRACEBOT MANIPULATOR**	

NOMBRE DE PAGES : #

Commissariat à l'énergie atomique et aux énergies alternatives
Institut List | Département Intelligence Ambiante et Systèmes Interactifs
CEA Saclay - Nano-Innov - Bâtiment 861 - Point Courrier 173
91191 Gif-sur-Yvette Cedex
T. +33 (0)1 69 08 00 99
www-list.cea.fr

RESUME :

MOTS CLÉS : Grasping, TRACEBOT, Task-Centered, Analysis, Python

1		*Validation électronique*	*Validation électronique*	*Validation électronique*	*Validation électronique*	
0		RICO URIBE Ricardo	Chef de labo	Chef de projet	Chef de service	
Rév.	Date	Rédacteur	Vérificateur	Emetteur	Approbateur	Pages modifiées

LISTE DE DIFFUSION

DESTINATAIRES	SOCIETE	Adresses Email
Thibault TORALBA	ENSTA Paris	thibault.toralba@ensta-paris.fr

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

Confidential Report
Not for diffusion or distribution

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

Anglais
Key Words:THE DOG
FRANCAIS
Mots Cles:LE CHIEN

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.

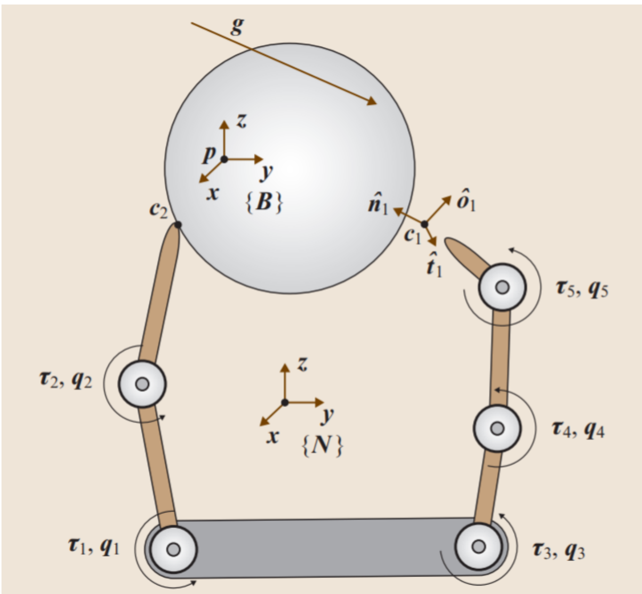

Figure 1: Main Notions for Grasping

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.

(inputenc) Package inputenc Error: Invalid UTF-8 byte "94See the inputenc package documentation for explanation.The document does not appear to be in UTF-8 encoding.Try adding as the first line of the fileor specify an encoding such as [latin1]inputencin the document preamble.Alternatively, save the file in UTF-8 using your editor or another toolc(inputenc) Package inputenc Error: Invalid UTF-8 byte "82See the inputenc package documentation for explanation.The document does not appear to be in UTF-8 encoding.Try adding as the first line of the fileor specify an encoding such as [latin1]inputencin the document preamble.Alternatively, save the file in UTF-8 using your editor or another toolc(inputenc) Package inputenc Error: Invalid UTF-8 byte "94See the inputenc package documentation for explanation.The document does not appear to be in UTF-8 encoding.Try adding as the first line of the fileor specify an encoding such as [latin1]inputencin the document preamble.Alternatively, save the file in UTF-8 using your editor or another toolc(inputenc) Package inputenc Error: Invalid UTF-8 byte "82See the inputenc package documentation for explanation.The document does not appear to be in UTF-8 encoding.Try adding as the first line of the fileor specify an encoding such as [latin1]inputencin the document preamble.Alternatively, save the file in UTF-8 using your editor or another toolc(inputenc) Package inputenc Error: Invalid UTF-8 byte "94See the inputenc package documentation for explanation.The document does not appear to be in UTF-8 encoding.Try adding as the first line of the fileor specify an encoding such as [latin1]inputencin the document preamble.Alternatively, save the file in UTF-8 using your editor or another toolc(inputenc) Package inputenc Error: Invalid UTF-8 byte "82See the inputenc package documentation for explanation.The document does not appear to be in UTF-8 encoding.Try adding as the first line of the fileor specify an encoding such as [latin1]inputencin the document preamble.Alternatively, save the file in UTF-8 using your editor or another toolc(inputenc) Package inputenc Error: Invalid UTF-8 byte "94See the inputenc package documentation for explanation.The document does not appear to be in UTF-8 encoding.Try adding as the first line of the fileor specify an encoding such as [latin1]inputencin the document preamble.Alternatively, save the file in UTF-8 using your editor or another toolc(inputenc) Package inputenc Error: Invalid UTF-8 byte "82See the inputenc package documentation for explanation.The document does not appear to be in UTF-8 encoding.Try adding as the first line of the fileor specify an encoding such as [latin1]inputencin the document preamble.Alternatively, save the file in UTF-8 using your editor or another toolc(inputenc) Package inputenc Error: Invalid UTF-8 byte "94See the inputenc package documentation for explanation.The document does not appear to be in UTF-8 encoding.Try adding as the first line of the fileor specify an encoding such as [latin1]inputencin the document preamble.Alternatively, save the file in UTF-8 using your editor or another toolc(inputenc) Package inputenc Error: Invalid UTF-8 byte "82See the inputenc package documentation for explanation.The document does not appear to be in UTF-8 encoding.Try adding as the first line of the fileor specify an encoding such as [latin1]inputencin the document preamble.Alternatively, save the file in UTF-8 using your editor or another toolc

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.

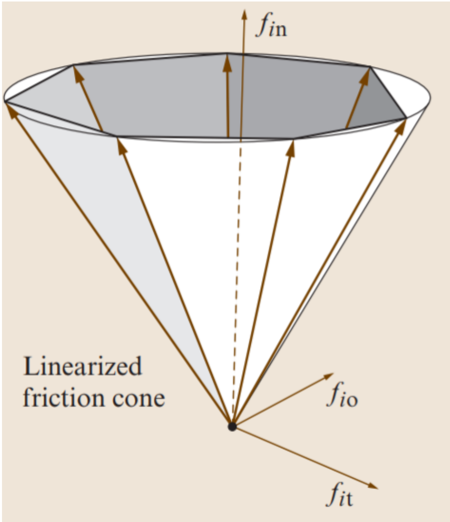

Figure 2: Faceted Friction Cone

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

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 alpha 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:

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

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

3.1 Structure

```
TASK ORIENTED GRASP ANALYSIS
├ code
├ excel
├ report
├ slides
└ stl
```

3.2 Development Order

3.3 Code in-depth

3.4 Work Flow for the data generation

3.5 Results and Data analysis

As starters I had to learn all theory pertaining the grasping concept, after this was done I created a code base in python (but using Object Oriented Logic) to calculate all the necessary matrices and operate with them. It was created as classes to be able to call more than one of the same element and change one attribute (for instance, to have the same object but two different grasps). The proposed work flow is to analysed thousands of different grasps and manipulator configurations to determine the best hand for each task and object. If this workflow is maintained and accomplished the objective of giving the technical specifications of the final manipulator will be achieved.

3.6 Current Work

In this moment I'm working on the force analysis for different grasp configurations using the task oriented metric to create a table were we can extract the necessary force for the contact normals to maintain the object in the air with out slipping, a work flow of the current work can be found in fig(3), for the project in general the HF type was chosen for its simplicity but the amount of information it saves.

3.7 Future Work

For the moment the next steps aren't clear, We will be analysing each grasp taxonomy to determine the required force for the actuators, this information will be pertinent if we stay with a human hand architecture for the manipulator, if not, the next steps would be the automatic generation of different manipulators architectures to find the most appropriate ones.

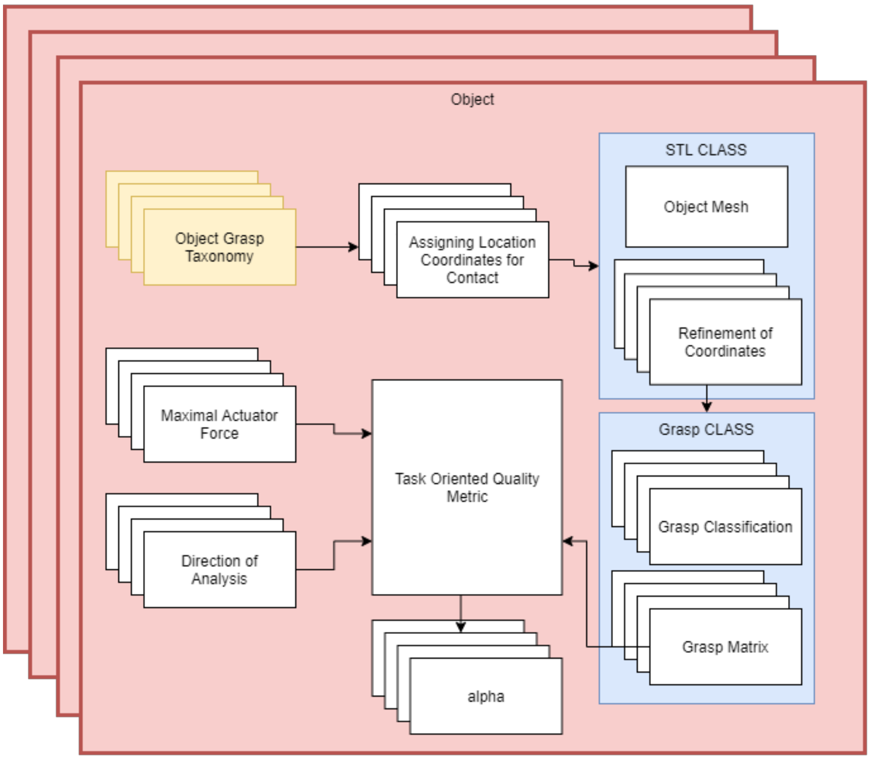

Figure 3: Flow of information to fill a table with Alphas

4 Code

The code base to solve this project is being written in Python, all mathematical calculations are based in the equations presented in the *Grasping* section.

For ease of use and understanding the code is divided in different categories. Classes are their own file, mathematical tools used by more than one class have their own file, the quality metrics have their own file, and at last the data structures created are joined together.

To test the code I solved some of the examples proposed in the book[4].

4.1 Architecture

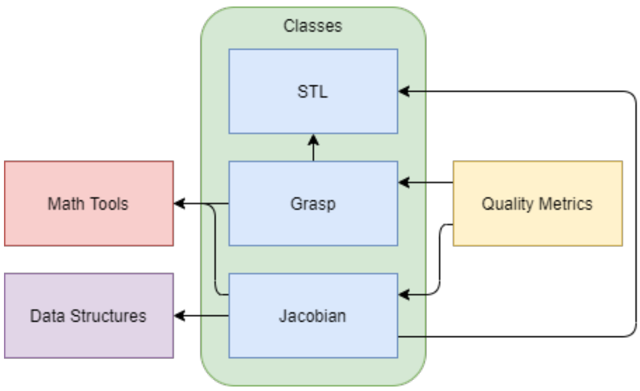

Figure 4: Code Architecture (the arrow points to the used file)

Conclusion

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

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, 10

H Component Selection Matrix. 8

N Newton. 11, 12

N Nullspace of a Matrix. 10

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, 17

g Sum of External Forces at p expressed in $\{N\}$. 7, 11

p Origin of Object Frame. 7, 17

CEA Atomic Energy Commissary. 6

DoF Degrees of Freedom. 10

FFC Frictional Form Closure. 10, 11

HF Hard Finger. 8, 9

LP Linear Programming. 10–12

nc Number of Contact Points. 7–9, 11

ng Number of faces. 11

nq Number of joints. 12

PwoF Point with-out Friction. 8–10

SF Soft Finger. 8, 9

Glossary

Trivial A trivial system is one that has the vector zero as an answer. 10

List of Figures

List of Tables

Appendix

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.