**ALMA MATER STUDIORUM - UNIVERSITÁ DI BOLOGNA**

Scuola di Ingegneria

Dipartimento di
Ingegneria dell'Energia Elettrica e dell'Informazione
"Guglielmo Marconi"
DEI

**MASTER DEGREE IN AUTOMATION ENGINEERING**

**GRADUATION THESIS**
in
System Theory and Advanced Control

# DESIGN AND CONTROL OF A FAST AND PRECISE AERIAL MANIPULATOR

CANDIDATE:
**Andrea Eusebi**

SUPERVISOR:
**Chiar.mo Prof.
Lorenzo Marconi**

Academic Year 2019/2020
Graduation Session II

*... a Karen e Christian, per avermi offerto quest'incredibile opportunità, per aver creduto in me e per avermi supportato in ogni singolo momento di questi lunghi mesi;*

*... a Shibashis, per la sua inaspettata amicizia e la sua genuina spontaeità;*

*... al Prof. Marconi, per la grande disponibilità dimostrata e per i preziosi consigli che ha saputo darmi;*

*... a mio zio Prof. D. Mucci, per il preziossisimo supporto in delicate questioni matematiche e per la revisione finale di questo elaborato;*

*... ai miei compagni di università: Andrea A., Andrea S., Alessio e Manuel, per avermi accolto fin da subito come un amico e aver condiviso insieme a me questi anni di studio;*

*... ai miei amici di Modena, troppi per essere menzionati tutti, per avermi lasciato, ciascuno a proprio modo, ricordi che rimarrano indelebili nel mio cuore;*

*... ai miei familiari, in particolare ai miei nonni, per l'amore che mi hanno donato, ed anche ai miei genitori, per l'incommensurabile affetto e per i grandi sacrifici che hanno reso possibile tutto questo;*

*... a tutte quelle persone che non ho potuto qui menzionare, per la loro amicizia, rispetto e generosità;*

*... e infine a Clara, per avermi fatto conoscere il significato della parola vita, per rimanere al mio fianco ogni giorno e per essere mia compagna in questo viaggio meraviglioso...*

*... a tutti voi, un immenso GRAZIE.*

*Bologna, 09/10/2020*

# Abstract

This report describes the work involved during a six months project carried on at *Autonomous Systems Lab* of ETH Zurich. The goal of this project is to develop a fast and precise robotic manipulator for aerial interaction. The omnidirectional MAV from ASL researchers is the aerial platform on which the manipulator will have to be mounted. First stage of the project involves an intensive literature research aimed at identifying those ideas and solutions which could fit our project requirements. From this review, the 3 DOF Delta parallel manipulator has been selected: lightness, reduced moving mass, precision and stiffness are the main advantages of this robotic manipulator. Subsequently, an extensive work has been focused on the study of robot kinematics and dynamics. A non linear optimization problem has been formulated to solve the synthesis task of geometric parameters by means of a so called genetic algorithm. Most of Delta mechanical components have been designed using CAD software and machined by 3D printing technology, while just few elements have been obtained by third party suppliers. Finite element analysis has been exploited for mechanical validation. Also system control has been taken into account: an inverse kinematics-base approach has been developed aimed at compensating for aerial platform pose errors and thus maintain end-effector position to millimetric accuracy. Lastly, real field tests have been run to evaluate system performances, data have been collected and presented to show positive and negative aspects of the designed system. Final conclusions about possible future improvements is the last contribution provided by this work.

# Abstract

Questo report descrive il lavoro svolto in un progetto della durata di sei mesi presso il laboratorio *Autonomous Systems Lab* dell'università ETH - Politecnico federale di Zurigo. L'obiettivo di questo progetto è lo sviluppo di un manipulatore robotico per compiti di manipolazione aerea ad elevata velocità e precisione. Il suddetto manipolatore dovrà infatti essere montato su una specifica piattaforma aerea sviluppata dai ricercatori dell'Autonomous Systems lab. Inizialmente, la prima fase del progetto prevede un'estesa ricerca bibliografica finalizzata a prendere conoscenza dello stato dell'arte nella materia e identificare i prototipi e le soluzioni più interessanti e in linea con i nostri obiettivi. Da questa ricerca, il manipolatore robotico Delta a 3 gradi di libertà è stato selezionato in quanto in grado di offrire ottime performance in termini di velocità, peso e precisione. Successivamente, si è passati a studiare nel dettaglio la cinematica e la dinamica del robot Delta, le cui equazioni sono riportate con cura in questo testo. La sintesi dei parametri geometrici del manipolatore Delta è stata formulata come un problema di ottimizzazione non lineare basato su un algoritmo *genetico*. In seguito, la maggior parte dei componenti meccanici del robot sono stati progettati utilizzando appositi software CAD e validati utilizzando l'analisi agli elementi finiti. Una volta approvato il design meccanico, le singole componenti sono state fabbricate utilizzando lo stampaggio 3D e solo alcune parti secondarie sono state ottenute da fornitori terzi. In maniera altrettanto approfondita, un sistema di controllo basato sulla cinematica inversa del robot Delta è stato progettato e sviluppato per offrire le massime performance in termini di precisione e velocità nella compensazione degli errori di posizione generati dalla piattaforma aerea. Infine, diversi test sul campo sono stati effettuati per ottenere una misurazione diretta delle abilità del robot ed evidenziare le possibili lacune di quest'ultimo, i risultati di questi test sono riportati e brevemente discussi nel capitolo finale di questo report. Alcune considerazioni finali riguardanti possibili futuri sviluppi del manipolatore Delta sviluppato nei sei mesi di lavoro rappresentano l'ultimo contributo di questo progetto.

# Contents

# Symbols and Acronyms

## Delta Robot Symbols

| | |
|---|---|
| $R$ | fixed base radius |
| $r$ | movable plate radius |
| $L$ | upper link length |
| $l$ | lower link length |
| $\overline{R}$ | virtual fixed base radius after radial shift |
| $\theta_i$ | *i-th* joint angles |
| $\boldsymbol{P}$ | movable plate center point |
| $\boldsymbol{B}_i$ | begin tip of *i-th* upper link |
| $\boldsymbol{C}_i$ | end point of *i-th* upper link |
| $\boldsymbol{P}_i$ | end point of *i-th* lower link |
| $I$ | set of geometric parameters |
| $\overline{I}$ | set of independent geometric parameters |
| $\boldsymbol{\alpha}$ | angular shift vector |

## Usage of subscripts and superscripts

| | |
|---|---|
| $x_A$ | x coordinate of point $A$ |
| $\boldsymbol{x}^A$ | x axis vector of frame $A$ |
| ${}^A x$ | x coordinate projected in frame $A$ |

## Acronyms and Abbreviations

| | |
|---|---|
| ETH | Eidgenössische Technische Hochschule |
| ASL | Autnomous Systems Lab |
| UAV | Unmanned Aerial Vehicle |

| | |
|---|---|
| MAV | Micro Aerial Vehicle |
| DOF | Degrees of Freedom |
| CFRP | Carbon Fiber Reinforced Polymer |
| CAD | Computer Aided Design |
| FEA | Finite Element Analysis |
| IK | Inverse Kinematics |
| FK | Forward Kinematics |

# Chapter 1

# Introduction

## 1.1 Framework

*Unmanned aerial vehicles* are aircrafts which do not require any human pilot on board to fly and perform missions. Depending on its functionalities and characteristics an UAV can be remotely controlled or fly autonomously and some times it is designed to operate in both conditions.
*Micro aerial vehicles* belong to the class of miniature UAVs and, as the name suggests, they are the small version of standard UAVs. Typical dimensions of MAVs are usually around few decimeters but in the recent years new vehicles have been developed which measure less than 5 centimeters.
While standard UAVs have been studied and developed for several decades, in particular for military purposes, MAVs have been subjects of interested in the last years thanks to technology progress and hardware miniaturization [1].
MAVs have been built for multiple purposes, both for military and civilian sake; lightweight and portability are the main features for which these devices are used and studied. Regarding this, these small aircrafts allow remote observation of hazardous environments which are inaccessible to ground vehicles or too risky for human operations.

As already said, the ability of reaching places which are out of range of humans is one of the key feature of MAVs; however, sometimes it would be also desirable to physically interact with these environments. For this reason, an intensive study of aerial manipulation has been carried on in the last years, leading to several aerial systems designs and prototypes able to interact with the surrounding environment and perform complex tasks like transporting objects from one place to an other or holding sensor equipment for infrastructure inspection.
As can be imagined, the possible applications of aerial manipulation systems are extremely large but, at the same time, these devices suffer from several problems, as depicted also in [2]; the main difficulties which arise in this field are:

- the presence of a manipulator adds weight to the structure, and thus more powerful motors are required to ensure safe flight

- a dynamic coupling is established between the moving arm and the MAV. The manipulator will cause the center of mass of the overall system to move: this will generate strong disturbances to the aerial platform and requires to be properly taken into account in the control phase

- the aircraft will be subject to forces and torques generated in the interactions and this could compromise its stability

- aerodynamic perturbations are generated while flying in proximity of surfaces or walls

- due to the low weight of the system, also wind effects have to be considered and cancel while flying close to objects.

In conclusion, micro aerial manipulation systems development is today a work-in-progress topic, several research groups are working in this field and constantly proposing new designs and prototypes; however, while these solutions have already proved good results, further developments still have to be achieved in order to optimally solve all the aforementioned issues.

## 1.2 Objectives of this Thesis

The main objective of this project is to develop a *fast* and *precise lightweight* aerial manipulator for high precision tasks in aerial interaction.
To ensure highest precision, the overall system (MAV + manipulator) will have to be able to reject any disturbance due to wind and center of mass displacement. In particular, the manipulator must be able to compensate for errors, and maintain the end-effector position to millimetric accuracy.
The developed arm will be mounted on an aerial platform designed by Autonomous System Lab researchers [3], which is shown in figure 1.1.

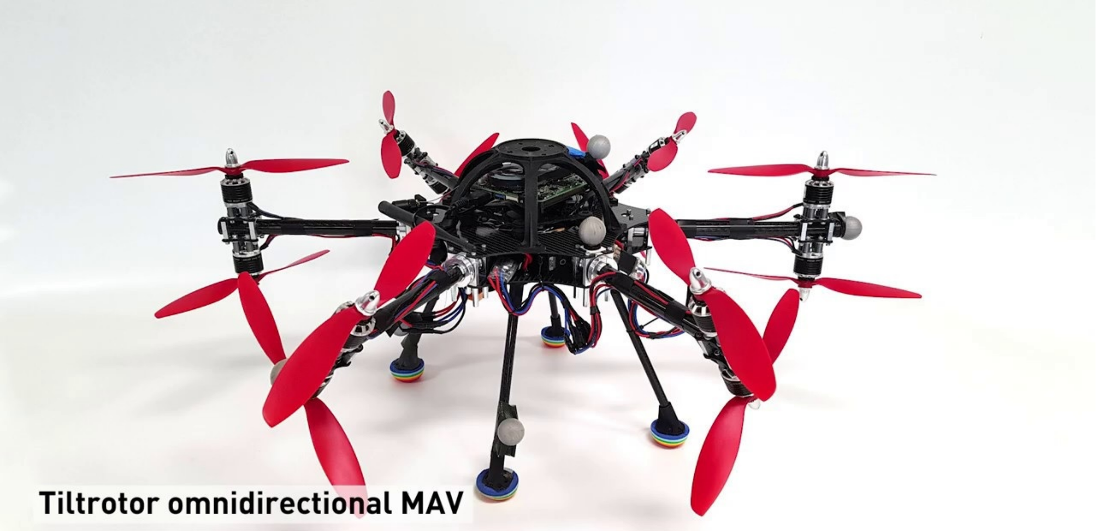


Figure 1.1: ASL's OMAV

This platform is an omnidirectional MAV equipped with six couples of tiltable propellers. The system is able to exert torque and force in any direction and thus can move freely in all six DOF.

The workflow carried on in this thesis can be summarized as follow:

- Literature study
- Mechanical design and prototyping of a fast and light aerial manipulator
- Design and implementation of a control framework for the manipulator and the base
- Design of tests and testing of the complete system
- Detailed evaluation of results and performances.

# Chapter 2

# Literature Study

As mentioned in the introduction, in the first part of this project an intensive literature study has been carried on, several research papers regarding aerial manipulation systems have been subject of study in order to get in touch with the topic and identify interesting features for project.
The present chapter will firstly provide a review of aerial manipulation state of the art, then some prototypes from literature will be shown and briefly analyzed. Subsequently, the differences between parallel and serial devices will be extensively presented since they will provide useful information for the definitive manipulator choice, which will be presented at the end of the chapter.

## 2.1 Aerial Manipulation Systems Review

The review of aerial manipulation systems provided by Ding et al. [4] is the first document taken into consideration. The authors provide a wide review of aerial manipulation state of the art and assert that there are three main categories of manipulating devices in the aerial context:

- Grippers
- Multi DOF rigid body manipulators
- Cable and tether based systems

Before proceeding to analyze pros and cons of each typology, Ding et al. remark that the selection of the mechanical structure is a key step in this context; in fact, and as already underlined, the introduction of a manipulation device creates a dynamic coupling between this and the MAV. The coupling must be properly taken into account and it is clear that a ”superior” mechanical structure can drastically reduce the complexity of the controller and leads also to a system with better performance. The importance of this aspect is highlighted also by Ollero and Siciliano [2] in their book.

## Mechanical Design

As anticipated in the previous section, Ding et al. divided the mechanical design of manipulating devices in three main categories; it is important to underline that the selection of the mechanical structure directly affects also its control, this aspect, however, will be discussed in the next section.
The first typology of mechanical design for aerial manipulation is the *gripper*: this device typology is the simplest and, as stated by Ding et al., the most widely adopted solution until now. It is usually made of two fingers which get closed to catch the object and typically they are characterized by only one DOF. A sample image representing a MAV endowed with a gripper is shown in 2.1.

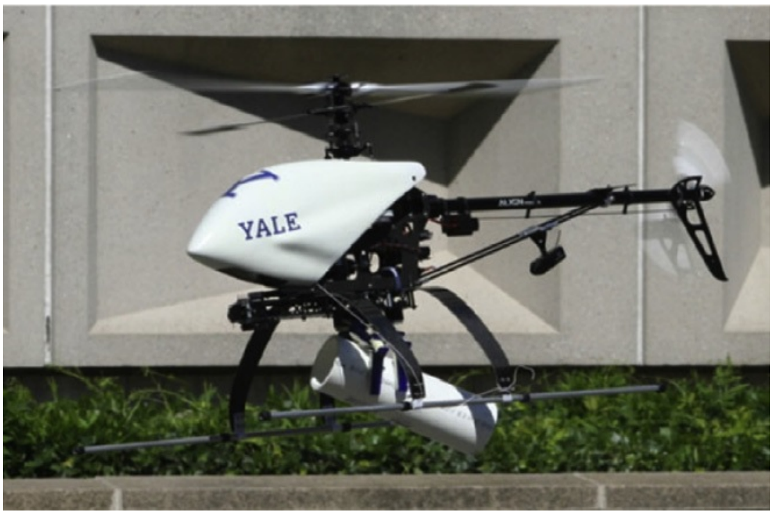


Figure 2.1: MAV with a gripper from Yale University (picture from [4])

The authors state that the main advantages of gripper solution are the following:

- Easy to build
- Conveniently modeled and controlled
- Relatively inexpensive

While the main disadvantages are:

- Limited workspace
- Limited grasping ability of mass and volume

The second typology mentioned by the authors is the *robotic manipulator*, these devices are composed of two parts: one or more multi DOF arms attached to the MAV and a gripper which can be optionally equipped with different kind of sensors.

As can be easily imagined, the presence of the manipulator drastically expands the available workspace and, nevertheless, its redundancy also allows to compensate for MAV position errors and thus ensures higher precision. However, this comes at the cost of strongly increasing the center of mass variation of the system which, as a consequence, requires an increased effort at the control level.
The general structure of the arm consists of a series-connected manipulator, while only a few prototypes of parallel mechanisms have been developed.
Summarizing, the main advantages of the robotic manipulator device are:

- Large workspace
- Ability to compensate for MAV position errors in some direction/orientation (depending on the degree of redundancy)
- High dexterity

While the main drawbacks are:

- More complex mechatronics system
- Heavy weight
- Difficult to control
- Severe coupling interference with the MAV

A sample image representing a MAV endowed with a robotic manipulator is shown in figure 2.2.

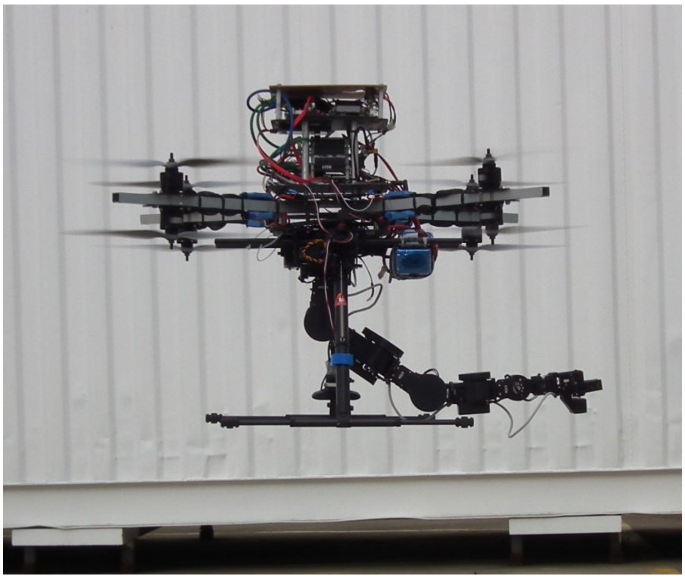


Figure 2.2: Manipulator prototype from Beihang University (picture from [4])

As previously said, the gripper solution is actually the more spread, however, in recent times, more and more researchers have preferred to adopt the manipulator design since undoubtedly can offer better performances in complex tasks; this seems to be the dominant trend for the next future too.
The third and last typology of manipulation device taken into consideration by Ding et al. are those equipped with a *cable* or a *tether*. The authors underline that this typology has not encountered very interest in general and it has been mainly adopted for transporting tasks.
The most critical point of these systems is the unavoidable variation in center of mass due to its intrinsic elastic nature. From a mechanical point of view, these devices are very simple and the effort needs to be focused almost exclusively at the control level to compensate for disturbances. However, these devices do not guarantee to be as precise as manipulator ones.

If we now focus on our task requirements, the selection of the most suitable typology seems to be straightforward: the advantages in terms of precision, dexterity and workspace size provided by the multi-DOF robotic manipulator largely overtake its drawbacks. Moreover, small workspace gripper devices could not be adopted at all since they would need to work too near to surfaces, generating strong aerodynamic interference; at the same time, also cable based systems are not a feasible solution for this project since they do not guarantee to perform precise tasks, which is one of the most important requirement.
In conclusion, we immediately *reject* any design based on grippers or cable systems but directly focus on the more sophisticated multi-DOF solution.

## Modeling and Control

Concerning the control phase, Ding et al. state that there are three different approaches which have been adopted in the context of aerial manipulation. The first solution is called *simplified* approach and is the one which received more attention at the beginning of the study of aerial manipulation systems. In fact, grippers were the first devices developed for aerial interaction and, as already highlighted, they are very small and lightweight. For this reason, researchers initially ignored the changes of center of mass and inertia during manipulation. In more recent times, with the adoption of more and more multi-DOF robotic arms this assumption became unacceptable and thus researches started developing two new different control solutions.
The first of these new solutions treats the MAV and the manipulator as two independent systems; each device has its own controller and the dynamic coupling between them is regarded as an *interference* problem. This approach is defined by the authors as *independent*.
The second methodology considers both the MAV and the manipulator as a unique system and treats the coupling as an *internal* problem. This solution

will be referred to as *overall* approach and clearly ensures higher accuracy with respect to the independent one: an unique control unit commands the overall system based on the coupled kinematics and dynamics. This, however, has the drawback of drastically increasing the complexity of the controller.
The three discussed solutions are now compared in the following table to underline pros and cons of each one:

| Modeling and Control Method | Difficulty | Accuracy | Feasibility |
|---|---|---|---|
| Simplified Approach | Low | Low | High |
| Independent Approach | Medium | Medium | High |
| Overall Approach | High | High | Medium |

Table 2.1: Comparison of Different Modelling and Control Methods

As shown by the table and as previously mentioned, the overall approach allows for the highest accuracy between the different solutions, at the cost of an increase in complexity. Since one of the key goals of this project is to design a system as much precise as possible, it could seem straightforward to adopt this approach. Anyway, it is difficult and impractical to get a real measure of the benefits of this solution with respect to the two ones, as well as of the difference in complexity.
Taking into account this consideration, it is reasonable to discard at this first stage only the simplified approach since it is clearly not suitable for our goals and to maintain into consideration both the independent and the overall solution. The overall solution will be the preferred choice but the final decision between them will be taken further on in the project and it will be mostly based on considerations about controller complexity, available time and differences in performances.
In other words, the overall approach will be the preferred one but, if enough time will not be available or in case the complexity of the overall solution will demonstrate to be too high, it may be interesting, or necessary, trying to develop the independent approach.

## 2.2 Prototypes and Design from Literature

After this initial review about aerial manipulation state of the art, several design and solutions proposed by researchers have been subject of attention. It has to be said that, in this part of the project, the main focus has been given to those systems which are in line with the decisions adopted in the previous sections, i.e., no gripper or cable solutions have been taken into account since already discarded.
This study has been carried on with the aim of both increasing knowledge in the field as well as identifying those features which could fit our project requirements. The most relevant solutions will be now briefly presented and then some final considerations will be discussed at the end of this section.

The first prototype taken into consideration is the one that Ollero and Siciliano proposed in their book [2]: the authors provide a methodology for designing a very lightweight multi-joint serial manipulator and then show a sample robotic arm. The system is conceptually divided in three parts: end-effector, main body and base, as shown in figure 2.3.

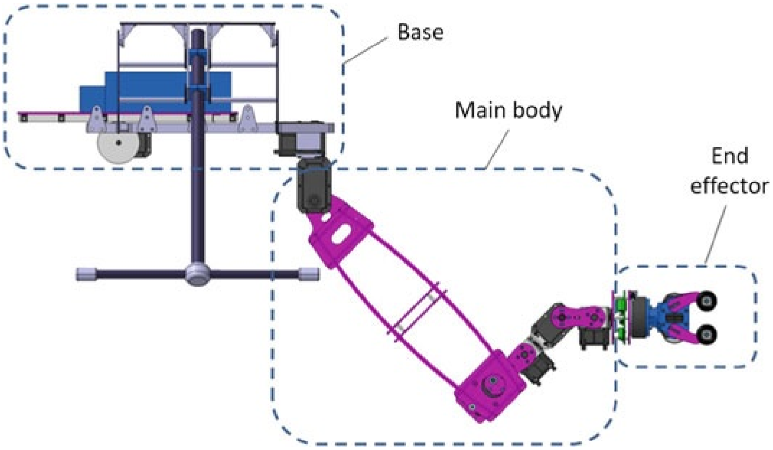


Figure 2.3: Multi DOF serial manipulator scheme(picture from [2])

The end-effector is a gripper made of two claws which are connected to the same motor and get closed to hold objects by mechanical pressure. The main body is composed of six serial actuated rotational joints and most of the links are made of CFRP which ensures high stiffness but very low weight. Moreover, the robotic arm is able to retract within the motor chain and adopt a compact configuration thanks to its particular "U" design which allows to minimize center of mass displacement during normal flight. Lastly, the base, which is the fixed component supporting the mobile part of the arm, is also made of CFRP and it is equipped with a small servo motor which moves the batteries of the system through a linear guide to counterbalance the displacement of center of mass of the aerial

manipulator induced by the movement of the arm.
The second solution studied is the one proposed by Kamel et al. in their paper [5]: the authors developed a large workspace, parallel manipulator which enables physical interaction on the side, as well as below of the aerial robot. The mechanical structure of the manipulator is based on the well-known five bar linkages to which one additional motor is added to perform tasks below the robot. Most interesting aspects of this design concern the workspace, which is large and designed to ensure the robot to avoid any singular configuration while performing tasks; moreover, the geometric parameters of the arm are selected to reach very high dexterity and a global conditioning index is used to measure its performance.
The aerial manipulation platform proposed by Jimenez-Ciano et al. [6] is specifically projected to perform structure inspection and maintenance, especially for bridges, and it consists of a multi-DOF serial manipulator mounted on the top of the multirotor platform. This is a not so common solution which can guarantee very good results in performing measurements from the downside, but it seems not the best option for the tasks involved in this project.

The next three presented solutions are both based on the 3 DOF Delta parallel robot. The first of the mentioned prototypes is the system developed by Keemink et al. [7] which is able to perform ultrasonic non-destructive testing experiments thanks to its compliant end-effector: it is in fact provided with a particular Cardan gimbal that allows the ultrasonic sensor to compliantly interact with the environment. The cited manipulator is shown in fig 2.4.

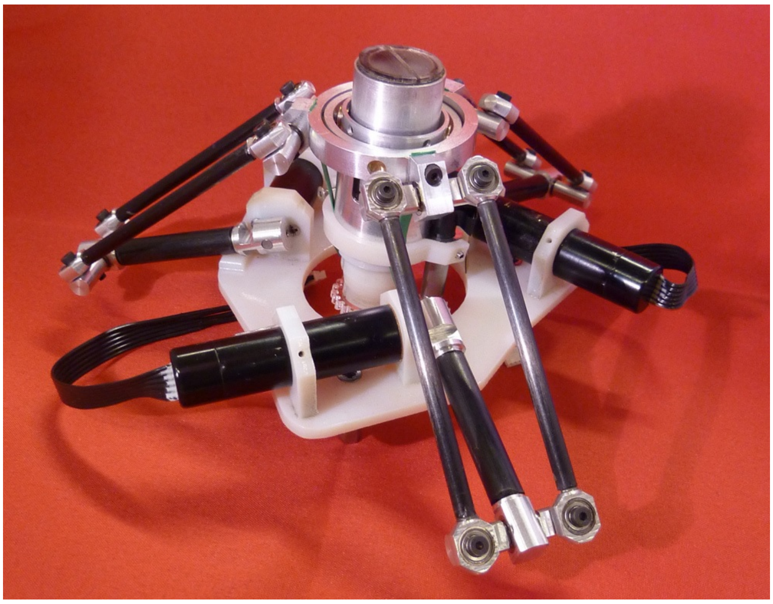


Figure 2.4: Delta robot manipulator with compliant end-effector (from [7])

The system is also endowed with a compliant element in the direction of interaction and two passive rotational DOF. This results in a manipulator which provides seven DOF relatively to the MAV and it passively orients itself perpendicularly to the surface of contact to counteract faulty positioning errors. As a final note, the authors decided to mount the manipulator on the side of the aerial platform to allow for an easier contact with vertical surfaces; however, this solution has the drawback of greatly displacing the center of mass of the overall system.

Chermprayong et al. [8] instead developed a Delta aerial manipulator for precise spot repair; the robot is able to deposit polyurethane foam on surfaces to sealing cracks and holes. The authors use a kinematic-based approach to compensate for fluctuations of the aerial platform due to errors or wind disturbances.

To conclude, the manipulator proposed by Cho and Shim [9] is the last prototype presented here. It is a parallel manipulator based on the Delta robot but with an an additional leg, providing the system with a fourth DOF. The end-effector control is based on an inverse-kinematics of the robot and it uses an attached visual sensor to capture information about surrounding environment. A lidar system is also mounted on the end-effector to measure distance from surfaces and a vacuum gripper is used to perform pick and place tasks.

In conclusion to this literature review, we can identify two main typologies of manipulator adopted by researchers: serial manipulators and parallel ones; in particular, the parallel solution seems to be often based on the Delta robot; this aspect will be treated in more details in the next section. Moreover, researchers have decided to mount the manipulator in different points of the aerial platform, this decision being mainly guided by the specific tasks that the system will have to perform. Considering that our system will have to be able to provide physical interaction but to maintain limited mass and inertia displacement, the best solution seems to be the one having the manipulator underside of the aerial platform; in fact, the design which mounts the manipulator over the MAV is not suitable for most of common tasks and the one using the robot on the side suffers from stability problems due to the strong variation of the position of the center of mass.

The end-effector choice is another key feature which has been differently developed by researchers: as for the base position, also the design of the end-effector is strongly affected by the goals that the system will have to achieve. Standard claws-based and vacuum grippers are often adopted and it is common to see the usage of sensors like cameras or lidars. The idea of implementing a compliant end-effector seems very interesting and could be adopted also in this project since it is a easy but effective way to counteract positioning errors. This idea will be further investigated in the next parts of the project according to time availability.

## 2.3 Parallel vs Serial Manipulators in the Aerial Context

As mentioned in the previous section, there are mainly two typologies of multi-DOF arms which have been used in the aerial context: serial manipulators and parallel manipulators. As the names suggest, serial devices are designed as a series of links connected by motor-actuated joints that extend from a base to an end-effector. Parallel systems are instead composed by close loop kinematic chains: the end-effector is linked to the base of the robot by multiple independent chains.

Both typologies have been extremely used in industry and in several other fields; serial manipulators have been the first ones to be developed and today represent the dominant device in industry while parallel solutions began to be used in more recent years but they started to attract attention of more and more companies thanks to their great performances in terms of speed and stiffness.

Patel and George [10] published an interesting document in which a detailed comparison between serial and parallel devices is carried on. In particular, the authors underline that serial manipulators suffer from bad precision due to their cantilever structure which made them susceptible to bending at high load and vibrating at high speed. On the other side, the main advantage of serial devices is that they ensure a larger workspace with respect to parallel ones: serial devices can in fact reach points which are out of range for parallel solutions.

Looking at parallel manipulators advantages, the authors assert that they can provide higher positioning precision also in high load demanding tasks since total load can be shared by a multiple number of parallel links; moreover, in these structures, errors will just be averaged and not added cumulatively as instead happens in serial devices.

Focusing on the features which more affects the aerial scenario we can already make some considerations: parallel designs have all the actuators attached to the base and centered with it, while serial manipulators have motors which move and are very far from the base. From this consideration we can make three simple but important deductions:

- parallel robots have less dispersion of mass, i.e., most of the mass (which is usually identified by the motors) is focused near the base
- parallel robots generate less variation of center of mass since they have less moving mass with respect to serial solutions
- parallel solutions also generate smaller variation of moments of inertia for the full system.

To complete the comparison between these two device typologies a table, based on the work provided in [10], is reported below highlighting pros and cons of both of them.

| Manipulator Feature | Parallel Manipulators | Serial Manipulators |
|---|---|---|
| Location of actuators | Near the base | On the links |
| Moving mass | Low | High |
| Dynamic interference | Low | High |
| Workspace size | Small | Large |
| Errors propagation | Averaged | Added |
| Preferred property | Stiffness | Dexterity |
| Direct kinematics | Complex | Simple and unique |
| Inverse kinematics | Simple and unique | Complex |
| Preferred application | Precise Positioning | Gross Motion |

Table 2.2: Comparison between Parallel and Serial Manipulators

As depicted by the table, it seems reasonable to assert that parallel solutions offer very nice performances in the aerial context, especially when high precision and stability are important requirements. From these straightforward considerations and also by keeping in mind that in this project we wish to design a very precise and stiff manipulator, it has been chosen to adopt a parallel configuration. In fact, their advantages in terms of small moving mass, reduced dynamic interference, averaged errors and stiffer behavior seem to be much more in line with our project specifications; serial manipulators remain however a good alternative which is still greatly widespread also in the aerial field.

## 2.4 Conclusions and Selected Design

As already said, it has been decided to adopt a parallel manipulator for our project; however, there exist several parallel designs in the literature and the selection of a particular one does not seem to be so easy. Nevertheless, by looking at the design which have already been used in the aerial manipulation it can be noticed that a great number of researchers decided to go for the so-called Delta robot; this trend has been highlighted also in section 2.2 and some interesting designs can be found in [7] and [8].

The Delta robot is a parallel manipulator with 3 kinematic chains, providing it with 3 translational DOF, it has been extremely used in the industry field, in particular in the packaging sector, thanks to its very high speed and precision. A conceptual scheme of the Delta robot is shown in figure 2.5.

However, modified versions of the Delta manipulator have been developed too: Cho and Shim [9] used a 4 legs version of the Delta robot and Danko et al. [11] a 6 legs version; each leg adds and extra DOF and thus provides the manipulator with higher and higher redundancy. The latter cited design, for example, is able

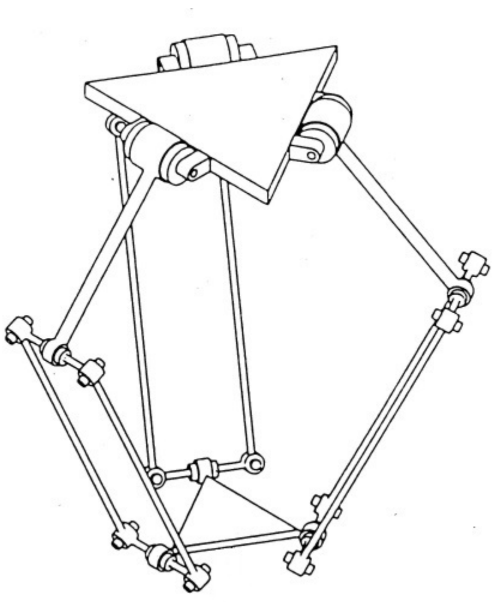


Figure 2.5: Schematic representation of the 3 DOF Delta robot

to move freely in all 6 directions: it has in fact 3 DOF in the translation domain and 3 DOF in the rotational domain.
Other typologies of parallel manipulators have been developed too, as e.g. the ones based on the five bar linkages proposed by Kamel et al. [5], however, these solutions do not seem to be as much effective as the ones based on the Delta robot.

After some further analysis, it has been decided to adopt a classical 3 DOF Delta robot configuration: in fact, the increase in complexity and weight given by higher Delta versions does not appear to be counterbalanced by an improvements in performances. The lack of motion capabilities in the rotational domain of the standard Delta robot is not considered to be too limiting, but instead could be overcome by the adoption of a compliant end-effector allowing for a self-orientation of the system, as previously underlined. Moreover, it must be reminded that the manipulator will be mounted on an omni-directional platform which can guarantee to provide the manipulator with the desired orientation.
In conclusion, standard Delta robots provide interesting advantages with respect to more sophisticated solutions, the main ones are the reduction of total mass and the decrease in complexity, both from a structural point of view and from control; for these reasons it has been decided to go for a standard 3 DOF Delta robot.

Finally, it is however very important to recall that the goal of this project is to design a very lightweight and precise manipulator, more specifically, the first design constraints imposed by the project and by the aerial platform payload capacity could be expressed as following:

- maximum robot weight: 750 grams
- end-effector desired accuracy: lower than 1 cm

Moreover, the system will have to be able to perform actions sufficiently far from surfaces to avoid aerodynamic disturbances, thus implying that the manipulator will have to be able to extend enough from its base. It will also be assumed that the maximum payload weight for our Delta robot will be 200 grams.
The study and development of our Delta manipulator will be presented in detail in the next chapters.

# Chapter 3

# Delta Robot

The current chapter will describe all the work involved during the analysis of the kinematics and dynamics of our Delta robot. This part of the report will firstly introduce the geometry and the parameters of the Delta robot, then it will focus on the kinematics of position, velocity and acceleration; lastly, dynamics equations will be presented. The analysis of both kinematics and dynamics will be shown here in details and all steps to final equations will be precisely explained.

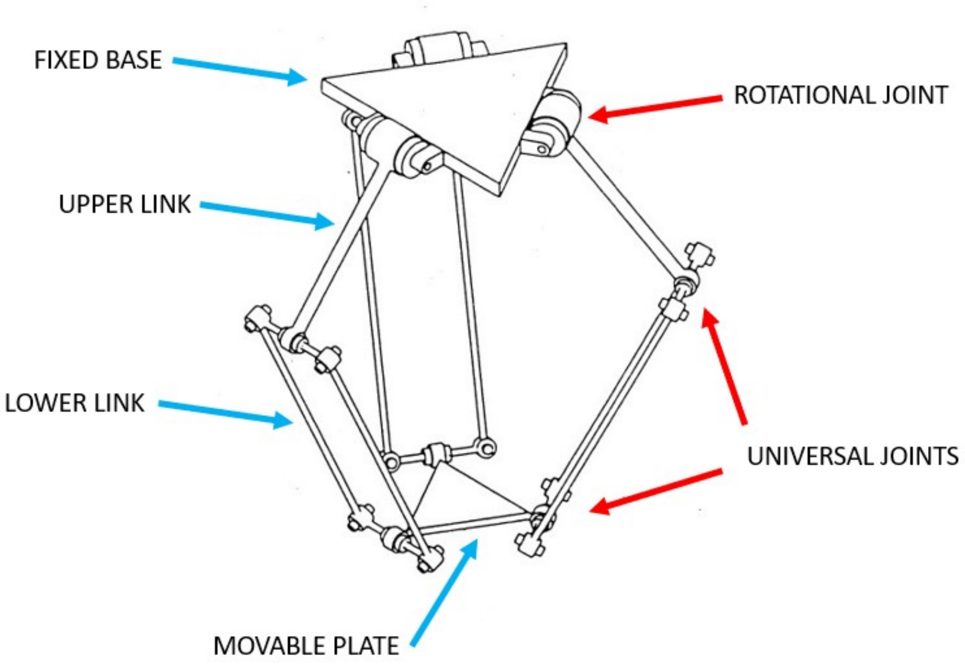


Figure 3.1: Delta robot structure with labelled components and joints

## Introduction to the Delta Robot

As already mentioned, the Delta robot is a 3 DOF parallel manipulator made of 3 independent and identical kinematic chains starting from a base, which henceforth will be referred to as the *fixed base*, to a final component, which will be referred to as the *movable plate*. The angular displacement between each kinematic chain is $360°/3 = 120° = 2\pi/3$. Each one of these chains is composed of 2 rigid links that will be called *upper link* and *lower link*; the joints connecting the upper link to the lower link and the ones connecting the lower link to the movable plate are *universal joints*, which provide rotation 2 or more axes, while the coupling between the fixed base and the upper link is managed by simple 1 DOF *rotational joints*. A schematic representation with labeled components is shown in figure 3.1 in the previous page, links and components names are indicated by light blue arrows while joints names by red ones.
It must be immediately remarked that the lower link is in turn made of 2 separate thin rods that form, together with the coupling elements to the upper link and the movable plate, a *parallelogram*; this is one of the key features of the Delta robot, it is in fact thanks to this particular structure that the movable plate maintains its orientation constant and parallel to the fixed base.

The motion of the robot is governed by the actuated rotational joints that connect the fixed base to the upper links; the angle assumed by each one of these joints will be indicated by $\theta_i$, where $i = 1, 2, 3$ refers to the *i-th leg* or kinematic chain of the robot.
From a geometric point of view, the Delta manipulator is described by 4 parameters, which are:

- $L$ - the length of each upper link
- $l$ - the length of each lower link
- $R$ - the radius of the fixed base
- $r$ - the radius of the movable plate.

All these geometric parameters will be henceforth collected in the *set of geometric parameters* $I = [L, l, R, r]$ which uniquely identifies a Delta robot.
A graphical representation of the these parameters is displayed in figure 3.2.

## Ratio of Links Lenght

Before starting to develop the kinematics equations of the Delta robot, a few more considerations are due to be done.
Firstly, the ratio between the upper link length, $L$, and the lower link, $l$, is a very

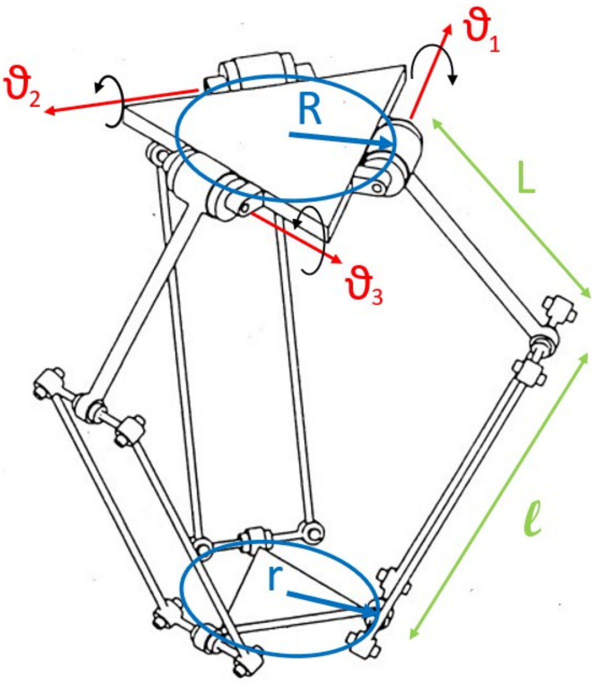


Figure 3.2: Delta robot with highlighted geometric parameters

important value which directly affects system performances, in particular, having $l > L$ implies that:

- the robot is endowed with a larger range of motion
- it can reach lower speed
- it can reach higher torque

while having $L > l$ implies:

- smaller range of motion
- higher speed
- lower torque.

Considering that both speed and torque performances can be compensated for by the choice of the motor, it is evident that the key point in this selection is the range of motion: from these initial considerations we can immediately assert that we want a Delta robot with:

$$l > L \tag{3.1}$$

this is in fact the usual design choice adopted in almost the totality of Delta robots.

## 3.1 Kinematics

### Nomenclature and Reference Frames

It is now necessary to define a couple of coordinates frames to describe position and orientation of geometric points in space.
There are two main Cartesian reference frames that will be vastly used in the next sections. The first one is a reference frame rigidly attached to the fixed base and having the following properties:

- center $O_B$ of the frame coincident with the center of the fixed base
- $x^B$-$y^B$ plane coincident with the plane where the vectors $\theta_i$ lay
- $x^B$ axis pointing toward the first leg of the robot.

This Cartesian reference frame will be henceforth referred to as $B$ and, in order to indicate points expressed in that frame it will be used the following notation $\{\}_B$. Moreover, since this frame will be the one with respect to which most of the geometric equations will be expressed, the subscript B could be sometimes omitted too, so as:

$$\begin{aligned} \{\}_B \equiv& \{\} \\ O_B \equiv& O \quad . \end{aligned}$$

The second important reference frame is the one attached to the movable plate and we will refer to this frame as $P$ and indicate points expressed in it by $\{\}_P$. This frame has the following properties:

- center $O_P$ of the frame coincident with the center of the movable plate
- $x^P$-$y^P$ plane coincident with the plane belonging to the movable plate
- $x^P$ axis pointing toward the first leg of the robot.

From these assumptions and thanks to the Delta robot property implying that the movable plate can not produce any angular movement, it can be immediately stated that:

- the three axes of frame $B$ and the three axes of frame $P$ are parallel and maintain always the same direction, respectively
- the position of the center of the movable plate, that is coincident with origin of $P$, can be expressed in $B$ as:

$$\{\mathbf{O}_P\}_B = \{\mathbf{P}\}_B = \{\mathbf{P}\} = \begin{bmatrix} x_P \\ y_P \\ z_P \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad . \tag{3.2}$$

A quick remark, the usage of superscripts in this text indicates that we are referring to the coordinate axes of a specific reference frame, while subscripts are used to indicate the coordinates of a specific point.
Both these two reference frames are now shown in the figure 3.3 alongside with the relevant geometric points of the Delta robot, which will be immediately presented.

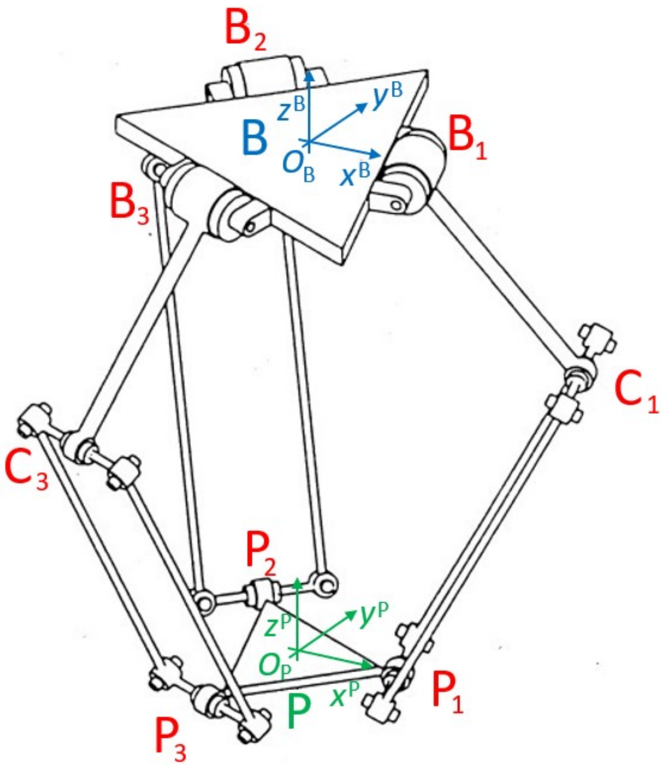


Figure 3.3: Delta robot Cartesian frames and and geometric points

Three points, $\mathbf{B}_i$, $\mathbf{C}_i$ and $\mathbf{P}_i$ are specified for each Delta robot leg $i$, where $i = 1, 2, 3$. The point $\mathbf{B}_i$ can be seen as the *hip* of the robot and it identifies the joint between the fixed base and the upper link, the *knee* $\mathbf{C}_i$ instead identifies the connection point between each upper link and corresponding lower link (modeled as a single straight rod). Lastly, the *ankle* point $\mathbf{P}_i$ represents the joint between each lower link and the movable plate.

It is now necessary to introduce the last items that will be used in the study of the robot. First of all, we wish to express the angular displacement between the different legs of the Delta robot using a vector, to which we will refer as the *angular displacement vector*, denoted by $\boldsymbol{\alpha}$. Therefore, we set this $\boldsymbol{\alpha}$ as:

$$\boldsymbol{\alpha} = \begin{bmatrix}\alpha_1\\ \alpha_2\\ \alpha_3\end{bmatrix} = \begin{bmatrix}0\\ 2\pi/3\\ 4\pi/3\end{bmatrix} = \begin{bmatrix}0, 2\pi/3, 4\pi/3\end{bmatrix}^\mathsf{T} \quad . \tag{3.3}$$

Having said this, we will now define three new reference frames with origin coincident with $O_B$ and laying on the same $x^B$-$y^B$ plane of $B$ and three frames with origin coincident with $O_P$ and laying on the same $x^P$-$y^P$ plane of $P$. These new frames are obtained from $B$ and $P$ by simple rotations with respect to $z^B$ and $z^P$, respectively. The angles identifying the rotation of each frame are exactly the angular displacements among the legs of the Delta robots and thus the are expressed by the vector $\boldsymbol{\alpha}$.
These new frames will be referred to as *local* and named, respectively, $L_1, L_2, L_3$ and $M_1, M_2, M_3$. They are called local since each $L_i$ frame will have the $x$ coordinate pointing toward the corresponding $\mathbf{B}_i$ point, and each $M_i$ frame will have the $x$ coordinate pointing the corresponding $\mathbf{P}_i$ point; for this reason they will be very suitable to describe points belonging to the leg they are pointing towards.
From these considerations, we can write the rotation matrices transforming points expressed in the local frames $L_i$, with $i = 1, 2, 3$ into the basic frame $B$ as:

$$\mathbf{R}_B^i = \begin{bmatrix}\cos(\alpha_i) & -\sin(\alpha_i) & 0\\ \sin(\alpha_i) & \cos(\alpha_i) & 0\\ 0 & 0 & 1\end{bmatrix}, \quad \textit{where} \;\; i = \{1, 2, 3\} \tag{3.4}$$

in a similar way, we can write the rotation matrices transforming points expressed in $M_i$, with $i = 1, 2, 3$ into the basic frame $P$ as:

$$\mathbf{R}_P^i = \begin{bmatrix}\cos(\alpha_i) & -\sin(\alpha_i) & 0\\ \sin(\alpha_i) & \cos(\alpha_i) & 0\\ 0 & 0 & 1\end{bmatrix}, \quad \textit{where} \;\; i = \{1, 2, 3\} \, . \tag{3.5}$$

It should be noticed that both $L_i$ and $M_i$ have been omitted and written simply as $i$, this notation will be maintained through the whole text because no confusion can be generated between them, since $L_i$ frames will be used always alongside $B$ and frames $M_i$ with only $P$.
We can also easily derive the rotation matrices transforming points expressed in $B$ (or $P$) into local frames $L_i$ (or $M_i$) by simply transposing (3.4) (or (3.5)):

$$\mathbf{R}_i^B = \mathbf{R}_i^P = \begin{bmatrix}\cos(\alpha_i) & \sin(\alpha_i) & 0\\ -\sin(\alpha_i) & \cos(\alpha_i) & 0\\ 0 & 0 & 1\end{bmatrix}, \quad \textit{where} \;\; i = \{1, 2, 3\} \, . \tag{3.6}$$

As a final remark, it should be noticed that $L_1$ and $B$ frames are exactly coincident as well as $M_1$ and $P$; in fact, all (3.4), (3.5) and (3.6)agree with the identity matrix $\mathbf{I}_{3\times 3}$ when $i = 1$.

## Kinematics Analysis

Using the reference frames defined in the previous section we now compute the coordinates of those points which remain fixed with respect to one or more of these frames.
We can start by noting that each point $\mathbf{B}_i$ of the fixed base can be represented in its local frame $L_i$ as:

$$\{\mathbf{B}_i\}_{L_i} = \{\mathbf{B}_i\}_i = \begin{bmatrix} R \\ 0 \\ 0 \end{bmatrix} , \quad where \;\; i = \{1, 2, 3\} \; . \tag{3.7}$$

In fact, it suffices to look at figure 3.4 and recall that each local frame $L_i$ is rotated with respect to the $z^B$ axis by a rotation of 0, $2\pi/3$ and $4\pi/3$, respectively.

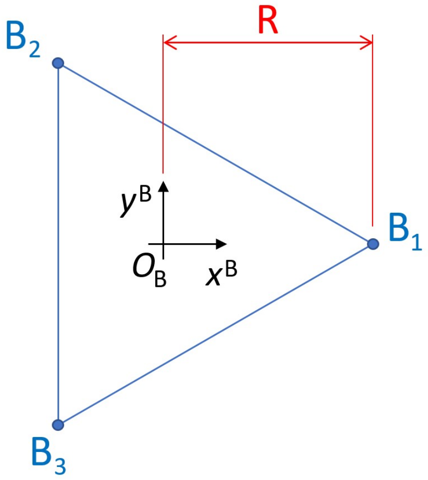


Figure 3.4: Schematic top view of the fixed base

Similarly, keeping as reference figure 3.5 we can derive the coordinates of each point $\mathbf{P}_i$ belonging to the movable plate with respect to each frame $M_i$ as:

$$\{\mathbf{P}_i\}_{M_i} = \begin{bmatrix} r \\ 0 \\ 0 \end{bmatrix} , \quad where \;\; i = \{1, 2, 3\} \; . \tag{3.8}$$

We can now derive the coordinates of all $\mathbf{B}_i$ points in the basic frame $B$ by simply multiplying each point $\mathbf{B}_i$ expressed in its local frame by the rotation matrix $\mathbf{R}_B^i$ presented in (3.4), so that in general:

$$\{\mathbf{B}_i\}_B = \{\mathbf{B}_i\} = \mathbf{R}_B^i \, \{\mathbf{B}_i\}_i = \begin{bmatrix} R\cos(\alpha_i) \\ R\sin(\alpha_i) \\ 0 \end{bmatrix} , \quad where \;\; i = \{1, 2, 3\} \; . \tag{3.9}$$

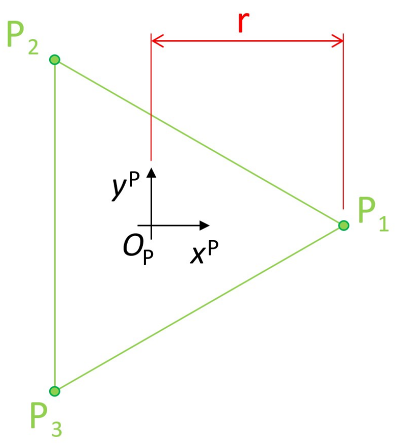


Figure 3.5: Schematic top view of the movable plate

By computing these matrix multiplications we find the coordinates for $\mathbf{B}_1$, $\mathbf{B}_2$ and $\mathbf{B}_3$ in $B$ frame:

$$\{\mathbf{B}_1\} = \mathbf{R}_B^1 \{\mathbf{B}_1\}_1 = \begin{bmatrix} \cos(0) & -\sin(0) & 0 \\ \sin(0) & \cos(0) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} R \\ 0 \\ 0 \end{bmatrix} \tag{3.10}$$

$$\{\mathbf{B}_2\} = \mathbf{R}_B^2 \{\mathbf{B}_2\}_2 = \begin{bmatrix} \cos(2\pi/3) & -\sin(2\pi/3) & 0 \\ \sin(2\pi/3) & \cos(2\pi/3) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}R \\ \frac{\sqrt{3}}{2}R \\ 0 \end{bmatrix} \tag{3.11}$$

$$\{\mathbf{B}_3\} = \mathbf{R}_B^3 \{\mathbf{B}_3\}_3 = \begin{bmatrix} \cos(4\pi/3) & -\sin(4\pi/3) & 0 \\ \sin(4\pi/3) & \cos(4\pi/3) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}R \\ -\frac{\sqrt{3}}{2}R \\ 0 \end{bmatrix}. \tag{3.12}$$

Notice that $\mathbf{B}_1$ does not change coordinates in the transformation between the two frames; this is correct since $B$ and $B_1$ are coincident, i.e., the corresponding rotation between them is the identity matrix.
We can now repeat the same procedure to find the coordinates of points $\mathbf{P}_i$ in the basic frame $P$. The *i-th* point $\{\mathbf{P}_i\}_P$ expressed in $P$ frame can be obtained by multiplying rotation matrix presented in (3.5) by $\{\mathbf{P}_i\}_i$:

$$\{\mathbf{P}_i\}_P = \mathbf{R}_P^i \{\mathbf{P}_i\}_{M_i} = \begin{bmatrix} r\cos(\alpha_i) \\ r\sin(\alpha_i) \\ 0 \end{bmatrix}, \quad where \;\; i = \{1, 2, 3\} \tag{3.13}$$

we then find the coordinates values by the already seen matrices multiplications:

$$\{\mathbf{P}_1\}_P = \mathbf{R}_P^1 \, \{\mathbf{P}_1\}_{M_1} = \begin{bmatrix} r \\ 0 \\ 0 \end{bmatrix} \tag{3.14}$$

$$\{\mathbf{P}_2\}_P = \mathbf{R}_P^2 \, \{\mathbf{P}_2\}_{M_2} = \begin{bmatrix} -\frac{1}{2}r \\ \frac{\sqrt{3}}{2}r \\ 0 \end{bmatrix} \tag{3.15}$$

$$\{\mathbf{P}_3\}_P = \mathbf{R}_P^3 \, \{\mathbf{P}_3\}_{M_3} = \begin{bmatrix} -\frac{1}{2}r \\ -\frac{\sqrt{3}}{2}r \\ 0 \end{bmatrix} . \tag{3.16}$$

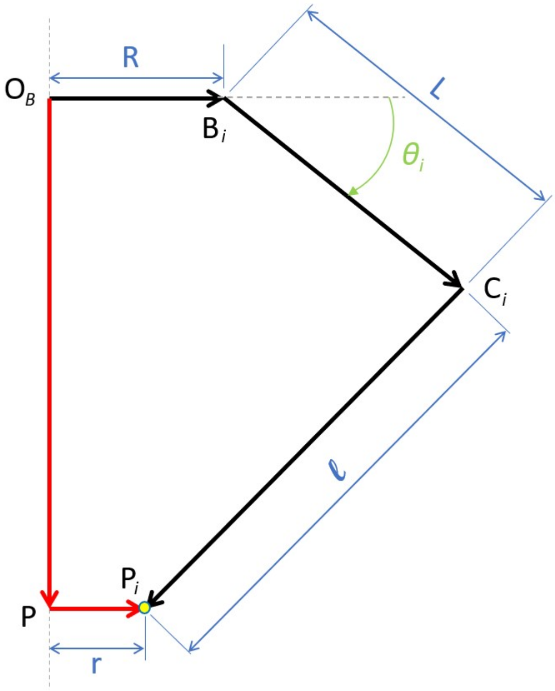


Figure 3.6: Vector-loop closure for the generic *i-th* Delta's leg

It is now the time to introduce an important concept which will be the base for the study of the kinematics of the Delta robot: let's thus define the three *vector-loop closure equations*, these geometric equations being used to represent the three kinematic closed chains which compose the robot; the geometric meaning of the vector loop closure for the *i-th* Delta's leg is shown in figure 3.6 and the corresponding geometric equations are presented in (3.17):

$$\begin{aligned}\{\mathbf{B}_i\}_B + \{\mathbf{C}_i - \mathbf{B}_i\}_B + \{\mathbf{P}_i - \mathbf{C}_i\}_B &= \{\mathbf{P}\}_B + \{\mathbf{P}_i - \mathbf{P}\}_B \\ &= \{\mathbf{P}\}_B + \mathbf{R}_B^P\{\mathbf{P}_i - \mathbf{P}\}_P \\ &= \{\mathbf{P}\}_B + \{\mathbf{P}_i\}_P \quad .\end{aligned} \tag{3.17}$$

It should be noticed that $\mathbf{R}_B^P = \mathbf{R}_P^B = \mathbf{I}_{3\times 3}$ since the two basic frames $B$ and $P$ have always the same orientation while they differ only for a translation offset. Moreover, let's remark that $\{\mathbf{P}_i - \mathbf{P}\}_P = \{\mathbf{P}_i\}_P$ since $\{\mathbf{P}\}_P$ is the origin of $P$ frame and thus could be omitted (the same is done for $\{\mathbf{B}_i\}_B$ where $\{\mathbf{O}_B\}_B$ is omitted).

The only vectors whose coordinates have not already been found are $\{\mathbf{C}_i - \mathbf{B}_i\}_B$ and $\{\mathbf{P}_i - \mathbf{C}_i\}_B$; it is however quite easy to find the values of the former by simple geometric evaluations, in fact we can immediately say that the vector representing the *i-th* upper link is:

$$\{\mathbf{C}_i - \mathbf{B}_i\}_B = \begin{bmatrix} L\cos(\theta_i) \\ 0 \\ -L\sin(\theta_i)\end{bmatrix} \quad . \tag{3.18}$$

Concerning the coordinates of the vector representing the lower link it is not possible to say a priori which are their values, but they can be obtained by simple manipulations of (3.17), thus:

$$\{\mathbf{P}_i - \mathbf{C}_i\}_B = \{\mathbf{P}\}_B + \{\mathbf{P}_i\}_P - \{\mathbf{B}_i\}_B - \{\mathbf{C}_i - \mathbf{B}_i\}_B \tag{3.19}$$

and with some computations we get:

$$\begin{aligned}\{\mathbf{P}_i - \mathbf{C}_i\}_B &= \begin{bmatrix} x \\ y \\ z\end{bmatrix} + \begin{bmatrix} r\cos(\alpha_i) \\ r\sin(\alpha_i) \\ 0\end{bmatrix} - \begin{bmatrix} R\cos(\alpha_i) \\ R\sin(\alpha_i) \\ 0\end{bmatrix} - \begin{bmatrix} L\cos(\theta_i) \\ 0 \\ -L\sin(\theta_i)\end{bmatrix} \\ &= \begin{bmatrix} x \\ y \\ z\end{bmatrix} - \overline{R}\begin{bmatrix} \cos(\alpha_i) \\ \sin(\alpha_i) \\ 0\end{bmatrix} - \begin{bmatrix} L\cos(\theta_i) \\ 0 \\ -L\sin(\theta_i)\end{bmatrix} \\ & \textit{where} \quad \overline{R} = R - r \quad .\end{aligned} \tag{3.20}$$

Taking into account the just introduced term $\overline{R}$ we can now make an important observation: from the last equation (3.20) it is evident that both geometric parameters $R$ and $r$ are *not really independent* since each pair of $\mathbf{B}_i$ and $\mathbf{P}_i$ points have the same orientation with respect to the two basic frames $B$ and $P$. In fact, the real independent parameter for a Delta robot is the difference of the upper and lower radius, i.e., $\overline{R}$. We can conclude that the set of geometric parameters $I$ is actually made of three independent values, and we can introduce the set of *independent* parameters as $\overline{I} = [\,L, l, \overline{R}\,]$.

Making this subtraction means translating radially each leg of the Delta robot by a quantity equal to $r$ (the term we are in fact subtracting); this idea will greatly simplify the geometric description of the robot since it will cause all $\mathbf{P}_i$ points to coincide with the center of the movable plate, i.e., $\mathbf{P}$, which can be seen as the final point of the robot. This way, we are guaranteed that the last point of each Delta's leg $\mathbf{P}_i$ will coincide with the final point of the robot and there will be no more the previous position offset due to the radius $r$ of the lower link.

The effects of this operation are now shown in figure 3.7 for all three legs of the Delta robot and, in more detail, for one single leg in figure 3.8 of the next section.

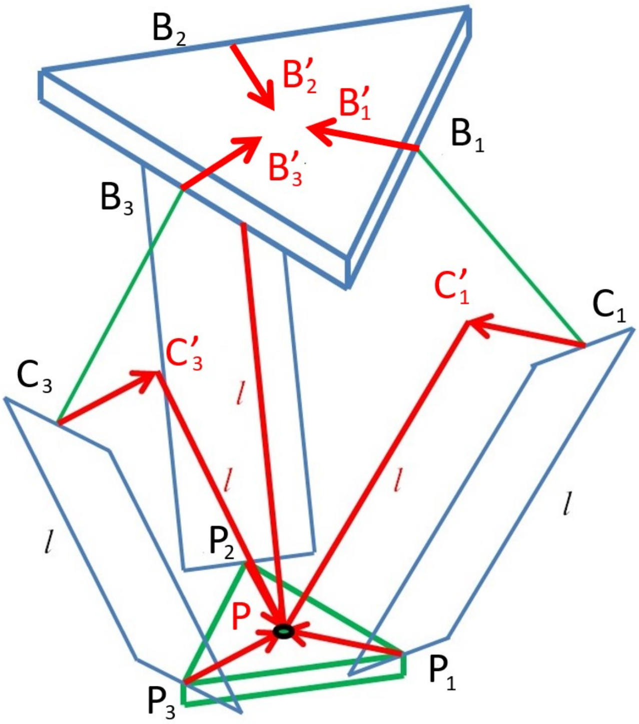


Figure 3.7: Radial shift equal to $r$ for each leg of the Delta robot

As already said, with this solution we greatly simplify the kinematic description of the robot, however, we have now to adjust the coordinates of $\mathbf{B}_i$ points accordingly. By denoting with $\overline{R} = R - r$ the new imaginary radius and by $\overline{\mathbf{B}}_i$ the resulting hip points, we get:

$$\{\overline{\mathbf{B}}_1\} = \begin{bmatrix} \overline{R} \\ 0 \\ 0 \end{bmatrix} \ , \quad \{\overline{\mathbf{B}}_2\} = \begin{bmatrix} -\dfrac{1}{2}\overline{R} \\ \dfrac{\sqrt{3}}{2}\overline{R} \\ 0 \end{bmatrix} \ , \quad \{\overline{\mathbf{B}}_3\} = \begin{bmatrix} -\dfrac{1}{2}\overline{R} \\ -\dfrac{\sqrt{3}}{2}\overline{R} \\ 0 \end{bmatrix} \tag{3.21}$$

and, as obvious, all $\mathbf{P}_i$ points being now coincident with the center of the movable plate, i.e. the final point of the robot $\mathbf{P}$. By using the already seen notation we can underline that the $\overline{\mathbf{P}}_i$ points, with respect to the $B$ frame, are coincident with $\mathbf{P}$:

$$\{\overline{\mathbf{P}}_1\} = \{\overline{\mathbf{P}}_2\} = \{\overline{\mathbf{P}}_3\} = \{\mathbf{P}\} \quad . \tag{3.22}$$

Also, it should be remarked that, with respect to the frame $P$, each $\overline{\mathbf{P}}_i$ will have null coordinates since they are coincident with the final point of the robot (which is the center of the movable plate):

$$\{\overline{\mathbf{P}}_1\}_P = \{\overline{\mathbf{P}}_2\}_P = \{\overline{\mathbf{P}}_3\}_P = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \ . \tag{3.23}$$

To conclude, we could now rewrite the vector closure loop equations provided in (3.19) with these latter points as:

$$\begin{aligned} \{\overline{\mathbf{P}}_i - \mathbf{C}_i\}_B &= \{\mathbf{P}\}_B + \{\overline{\mathbf{P}}_i\}_P - \{\overline{\mathbf{B}}_i\}_B - \{\mathbf{C}_i - \overline{\mathbf{B}}_i\}_B \\ &= \{\mathbf{P}\}_B - \{\overline{\mathbf{B}}_i\}_B - \{\mathbf{C}_i - \overline{\mathbf{B}}_i\}_B \quad . \end{aligned} \tag{3.24}$$

### 3.1.1 Forward Kinematics

Most of the equations developed in this and next section, about inverse kinematics, are based on [12] and [13].
The forward kinematics problem can be stated this way: given the *vector of actuated joints* $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3]^\intercal$, calculate the resulting Cartesian position of the movable plate, i.e., the end-point of the robot which has been defined in (3.2) as $\{\mathbf{P}\}_B = [x, y, z]^\intercal$.
Notice that the solution of the forward kinematics for parallel robots is usually very complex since it requires to solve a system of multiple coupled nonlinear algebraic equations; moreover, in general this kinematic problem fails to have uniqueness of solutions.

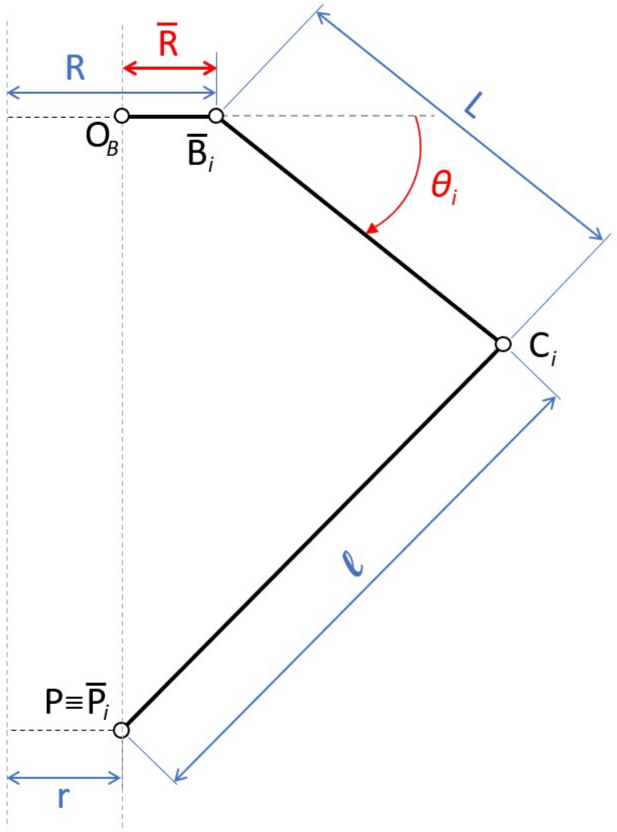


Figure 3.8: $x$-$z$ plane view of *i-th* Delta's leg after radial shift

Figure 3.8 represents a schematic $x$-$z$ plane view for the generic *i-th* Delta's leg after the radial $r$ shift, note that now the radius of the fixed base is $\overline{R}$ and that $\overline{\mathbf{P}}_i$ is coincident with $\mathbf{P}$. However, two remarks should be done about this image: firstly, the center of the fixed base $\mathbf{O}_B$ has been taken vertically aligned with $\mathbf{P}$ but all the results will obviously hold for any position of the latter point; secondly, in general the vector representing the lower link $\{\overline{\mathbf{P}}_i - \mathbf{C}_i\}$ of each Delta's leg does not belong to this $x$-$z$ plane since, as expected, the movable plate can translate in all three directions and thus this vector could have a component in the $y$ direction too.

Keeping in mind these two remarks and that in the forward kinematics problem the given input is the vector $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3]^\intercal$ we can write the geometric equation providing $\mathbf{C}_i$ with respect to the corresponding *i-th* local frame as:

$$\{\mathbf{C}_i\}_i = \{\overline{\mathbf{B}}_i\}_i + \{\mathbf{C}_i - \overline{\mathbf{B}}_i\}_i \quad . \tag{3.25}$$

The coordinates of the upper link vector has already been derived before we adopted the radial shift in (3.18), however its coordinates remain the same since all its points are equally translated, thus:

$$\{\mathbf{C}_i - \mathbf{B}_i\}_i = \{\mathbf{C}_i - \overline{\mathbf{B}}_i\}_i = \begin{bmatrix} L\cos(\theta_i) \\ 0 \\ -L\sin(\theta_i) \end{bmatrix} . \tag{3.26}$$

Finally, by considering (3.21), (3.25) and (3.26) we can now find the coordinates of $\{\mathbf{C}_i\}_i$ in the local frame $i$ as:

$$\{\mathbf{C}_i\}_i = \begin{bmatrix} \overline{R} + L\cos(\theta_i) \\ 0 \\ -L\sin(\theta_i) \end{bmatrix} . \tag{3.27}$$

As already seen, it is immediate to express the coordinates of the generic $\mathbf{C}_i$ point in the base frame $B$ by means of a rotation (3.4):

$$\begin{aligned} \{\mathbf{C}_i\} = \mathbf{R}_B^i \{\mathbf{C}_i\}_i &= \begin{bmatrix} \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \overline{R} + L\cos(\theta_i) \\ 0 \\ -L\sin(\theta_i) \end{bmatrix} \\ &= \begin{bmatrix} \cos(\alpha_i)(\overline{R} + L\cos(\theta_i)) \\ \sin(\alpha_i)(\overline{R} + L\cos(\theta_i)) \\ -L\sin(\theta_i) \end{bmatrix} \end{aligned} \tag{3.28}$$

and therefore the three knee points $\{\mathbf{C}_i\}$ result to be, respectively:

$$\{\mathbf{C}_1\} = \begin{bmatrix} \cos(0) \cdot (\overline{R} + L\cos(\theta_1)) \\ \sin(0) \cdot (\overline{R} + L\cos(\theta_1)) \\ -L\sin(\theta_1) \end{bmatrix} = \begin{bmatrix} \overline{R} + L\cos(\theta_1) \\ 0 \\ -L\sin(\theta_1) \end{bmatrix} \tag{3.29}$$

$$\{\mathbf{C}_2\} = \begin{bmatrix} \cos(2\pi/3) \cdot (\overline{R} + L\cos(\theta_2)) \\ \sin(2\pi/3) \cdot (\overline{R} + L\cos(\theta_2)) \\ -L\sin(\theta_2) \end{bmatrix} = \begin{bmatrix} -\dfrac{1}{2} \cdot (\overline{R} + L\cos(\theta_2)) \\ \dfrac{\sqrt{3}}{2} \cdot (\overline{R} + L\cos(\theta_2)) \\ -L\sin(\theta_2) \end{bmatrix} \tag{3.30}$$

$$\{\mathbf{C}_3\} = \begin{bmatrix} \cos(4\pi/3) \cdot (\overline{R} + L\cos(\theta_3)) \\ \sin(4\pi/3) \cdot (\overline{R} + L\cos(\theta_3)) \\ -L\sin(\theta_3) \end{bmatrix} = \begin{bmatrix} -\dfrac{1}{2} \cdot (\overline{R} + L\cos(\theta_3)) \\ -\dfrac{\sqrt{3}}{2} \cdot (\overline{R} + L\cos(\theta_3)) \\ -L\sin(\theta_3) \end{bmatrix} . \tag{3.31}$$

We are now ready to introduce the so called *three virtual spheres* problem: since we have already found the position of the three knee points $\mathbf{C}_i$, we can imagine to have three spheres with radius $l$ each one centered in one $\mathbf{C}_i$. The forward kinematics problem consists then in looking for intersection points of these three spheres, which are in fact the possible solutions of the forward kinematics.
This concept is valid also thanks to the adopted radial shift of robot legs, it would have been in fact necessary to take into account the position offset of each single $\mathbf{P}_i$ point if we had not made such an assumption.
Translating this statement in mathematical expressions, it ends up being a system of three non linear equations, each one corresponding to a sphere in 3D space with center $\mathbf{C}_i$ and radius $l$. Projecting all the coordinates into the base frame $B$ we get:

$$\begin{cases} [x - (\cos\alpha_1\,(\overline{R} + L\cos\theta_1))]^2 + [y - (\sin\alpha_1\,(\overline{R} + L\cos\theta_1))]^2 + [z + L\sin\theta_1]^2 = l^2 \\ [x - (\cos\alpha_2\,(\overline{R} + L\cos\theta_2))]^2 + [y - (\sin\alpha_2\,(\overline{R} + L\cos\theta_2))]^2 + [z + L\sin\theta_2]^2 = l^2 \\ [x - (\cos\alpha_3\,(\overline{R} + L\cos\theta_3))]^2 + [y - (\sin\alpha_3\,(\overline{R} + L\cos\theta_3))]^2 + [z + L\sin\theta_3]^2 = l^2 \end{cases} \tag{3.32}$$

As already said, it is a system of three non linear coupled equations in the three unknowns $(x, y, z)$. We refer to [12] for a detailed algorithm aimed at solving the system 3.32. However, the solution of this problem can be obtained by exploiting a mathematical software tool; in our case, we will use MATLAB to find the coordinates of point $\mathbf{P}$ and will not put any further effort in the study of this algebraic problem, since it is not the aim of this project.
Anyway, it is important to underline that the intersection of three spheres in space leads, in general, to multiple solutions. In particular, there are three different scenarios which may take place: firstly, it could happen that we obtain two different real solutions and, in this case, it is necessary to select as feasible solution only the one which ensures that the end point $\mathbf{P}$ of the robot is *below* the fixed base (since we want to work in this area only); secondly, it could happen that the problem admits only one real solution, which means that the three spheres meet tangentially; lastly, if the problem returns no real solutions it means that the distance among the centers $\mathbf{C}_i$ of the spheres is too high and thus the specific $\boldsymbol{\theta}$ configuration is not feasible.

### 3.1.2 Inverse Kinematics

The inverse kinematics problem consists in finding the values of the vector of actuated joint variables $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3]^\intercal$ as function of the position of the end point of the robot $\mathbf{P} = [x, y, z]^\intercal$. From this definition, it is clear that IK provides exactly the opposite results with respect to the FK problem; moreover, the IK is probably the most interesting one since it allows to compute the angles that the actuated joints have to reach in order to move the end-effector of the robot to the desired point.

As for the FK, also the inverse problem generally leads to multiple solutions which, differently from the forward one, could be all feasible and acceptable, for this reason it will be necessary to provide a rule to decide which solution is the *optimal* one, to be adopted in the sequel of the work. Anyway, we will focus on this topic further on, for now we start by finding the equations of the inverse kinematics for the 3 DOF Delta robot.

The basic idea behind the solution of the IK problem consists in looking for the intersection between a given *circle* of radius $L$ and centered in $\overline{\mathbf{B}}_i$ and a given *sphere* of radius $l$ and centered in $\mathbf{P}$, this procedure must be repeated for each Delta leg and each iteration in general provides two different possible solutions, i.e., two possible joint angles $\theta_i$ which are feasible solutions for that specific leg $i$. Considering all three legs, we will end up with 6 joint angles which will generate $2^3 = 8$ different combinations, i.e., 8 different possible solutions to choose among.

Let's thus rewrite the equation for the generic *i-th* leg of the three vector-loop closure in its final shape (3.24):

$$\{\overline{\mathbf{P}}_i - \mathbf{C}_i\}_B = \{\mathbf{P}\}_B - \{\overline{\mathbf{B}}_i\}_B - \{\mathbf{C}_i - \overline{\mathbf{B}}_i\}_B \quad .$$

We have now to impose that the Euclidean norm of the lower link vector is exactly equal to $l$, so:

$$\|\{\overline{\mathbf{P}}_i - \mathbf{C}_i\}_B\| = \|\{\mathbf{P}\}_B - \{\overline{\mathbf{B}}_i\}_B - \{\mathbf{C}_i - \overline{\mathbf{B}}_i\}_B\| = l \quad . \tag{3.33}$$

Now, by recalling the definition of $\{\mathbf{C}_i\}_i$ in (3.25) we can rewrite it into $B$ frame and then plug it into (3.33), thus:

$$\{\mathbf{C}_i\}_B = \mathbf{R}_B^i\{\mathbf{C}_i\}_i = \mathbf{R}_B^i(\{\overline{\mathbf{B}}_i\}_i + \{\mathbf{C}_i - \overline{\mathbf{B}}_i\}_i) = \{\overline{\mathbf{B}}_i\}_B + \{\mathbf{C}_i - \overline{\mathbf{B}}_i\}_B \tag{3.34}$$

$$\|\{\overline{\mathbf{P}}_i - \mathbf{C}_i\}_B\| = \|\{\mathbf{P}\}_B - \{\mathbf{C}_i\}_B\| = \|\{\mathbf{P} - \mathbf{O}^B\}_B - \{\mathbf{C}_i - \mathbf{O}^B\}_B\| = l \tag{3.35}$$

where in the last equation it has been remarked that $\{\mathbf{O}^B\}_B$ is usually omitted for $\{\mathbf{P}\}_B$ and $\{\mathbf{C}_i\}_B$.

Squaring both sides of (3.35) we get, for the generic *i-th* leg:

$$l^2 = \|\{\overline{\mathbf{P}}_i - \mathbf{C}_i\}_B\|^2 = \|\{\mathbf{P}\}_B - \{\mathbf{C}_i\}_B\|^2 = l_{i_x}^2 + l_{i_y}^2 + l_{i_z}^2 \tag{3.36}$$

and then, by recalling the definition of $\{\mathbf{P}\}_B$ and $\{\mathbf{C}_i\}_B$ from (3.2) and (3.28) we can write:

$$\{\mathbf{P}\}_B - \{\mathbf{C}_i\}_B = \begin{bmatrix} x - \cos(\alpha_i)(\overline{R} + L\cos(\theta_i)) \\ y - \sin(\alpha_i)(\overline{R} + L\cos(\theta_i)) \\ z + L\sin(\theta_i) \end{bmatrix} . \tag{3.37}$$

Finally, by combining (3.36) and (3.37) we obtain these three scalar equations:

$$l^2 = [x - \cos(\alpha_i)(\overline{R} + L\cos(\theta_i))]^2 + [y - \sin(\alpha_i)(\overline{R} + L\cos(\theta_i))]^2 + [z + L\sin(\theta_i)]^2 \tag{3.38}$$

$$where \;\; i = \{1, 2, 3\} .$$

Recapping, we have obtained three independent non linear scalar equations where the unknown is $\theta_i$ for each *i-th* equation.
Since the development of these equations is quite long, we decided not to report it here.
Finally, after some algebraic manipulations, we can write the resulting equations as:

$$2L(\overline{R} - x\cos(\alpha_i) - y\sin(\alpha_i))\cos(\theta_i) + 2zL\sin(\theta_i) + x^2 + y^2 + z^2 + \overline{R}^2 + L^2 - \\ - 2\overline{R}(x\cos(\alpha_i) + y\sin(\alpha_i)) - l^2 = 0 \tag{3.39}$$

$$where \;\; i = \{1, 2, 3\}$$

and it should be remarked another time that the unknowns are the three $\theta_i$ where $i = \{1, 2, 3\}$.
Following the procedure adopted also in [12], we can say that these equations can be represented in the following compact form as:

$$E_i\cos(\theta_i) + F_i\sin(\theta_i) + G_i = 0 \tag{3.40}$$

$$for \;\; i = \{1, 2, 3\} .$$

Where $E_i$, $F_i$ and $G_i$ are, respectively:

$$E_i = 2L(\overline{R} - x\cos(\alpha_i) - y\sin(\alpha_i)) \tag{3.41}$$

$$F_i = 2zL \tag{3.42}$$

$$G_i = x^2 + y^2 + z^2 + \overline{R}^2 + L^2 - 2\overline{R}(x\cos(\alpha_i) + y\sin(\alpha_i)) - l^2 \quad . \tag{3.43}$$

Equation (3.40) frequently appears in robotics and mechanism kinematics and it is readily solved using the so called *Tangent Half-Angle Substitution.*
So, if we now define:

$$t_i = \tan\left(\frac{\theta_i}{2}\right) \quad for \;\; i = \{1, 2, 3\} \tag{3.44}$$

then, thanks to trigonometric properties we can write:

$$\cos(\theta_i) = \left(\frac{1-t_i^2}{1+t_i^2}\right) \quad and \quad \sin(\theta_i) = \left(\frac{2t_i}{1+t_i^2}\right) \ . \tag{3.45}$$

We can now plug-in both terms presented in (3.45) into (3.40) and obtain:

$$\begin{aligned} &E_i\left(\frac{1-t_i^2}{1+t_i^2}\right) + F_i\left(\frac{2t_i}{1+t_i^2}\right) + G_i = 0 \\ &E_i(1-t_i^2) + F_i(2t_i) + G_i(1+t_i^2) = 0 \\ &(G_i - E_i)t_i^2 + 2F_it_i + G_i + E_i = 0 \\ &\qquad\qquad where \ \ i = \{1,2,3\} \end{aligned} \tag{3.46}$$

which are three common quadratic equations solvable with the usual methods, so:

$$t_{i_{1,2}} = \frac{-F_i \pm \sqrt{E_i^2 + F_i^2 - G_i^2}}{G_i - E_i} \tag{3.47}$$

$$where \ \ i = \{1,2,3\} \quad .$$

Then, for each *i-th* pair of solutions $t_{i_{1,2}}$ we get a pair of corresponding $\theta_{i_{1,2}}$ angle solutions for the *i-th* actuated joint:

$$\theta_{i_1} = 2\ \arctan(t_{i_1}) \quad and \quad \theta_{i_2} = 2\ \arctan(t_{i_2}) \tag{3.48}$$

$$where \ \ i = \{1,2,3\} \quad .$$

We have thus obtained three pairs of joint angles solutions for a total of 8 different combinations.

Regarding the choice among the several returned solutions, it is reasonable to say that, in a real world scenario, the *optimal* solution vector $\overline{\boldsymbol{\theta}} = [\overline{\theta}_1, \overline{\theta}_2, \overline{\theta}_3]$ is the one which *minimizes the sum of the differences* between the current angle $\theta_i(t^*)$ of the *i-th* joint and the adopted solution $\theta_{i_{1/2}}$ for that joint; in this way we ask the actuated joints to produce the smallest angular travel among all possible solutions. The mathematical formula expressing this concept is now presented:

$$\overline{\boldsymbol{\theta}} = \underset{\theta_{1_i},\theta_{2_j},\theta_{3_k}}{\arg\min} \big((\theta_1(t^*) - \theta_{1_i}) + (\theta_2(t^*) - \theta_{2_j}) + (\theta_3(t^*) - \theta_{3_k})\big) \tag{3.49}$$

$$where \ \ i,j,k = \{1,2\} \quad and \ \ t^* \ \ indicates \ \ the \ \ actual \ \ time \ .$$

Obviously, there could be also other principles to establish which is the optimal solution, these other methods could be even more refined than the one presented here; however, at least for now, we will maintain this as optimality principle and try to validate it both in simulation and on the real hardware.

As already mentioned, we used MATLAB for developing a script which provides the IK routine for a Delta robot based on the presented equations; this script returns also a graphical representation of the computed solutions like the one displayed in the figure 3.9, which shows the first solution returned by the inverse kinematics solver.

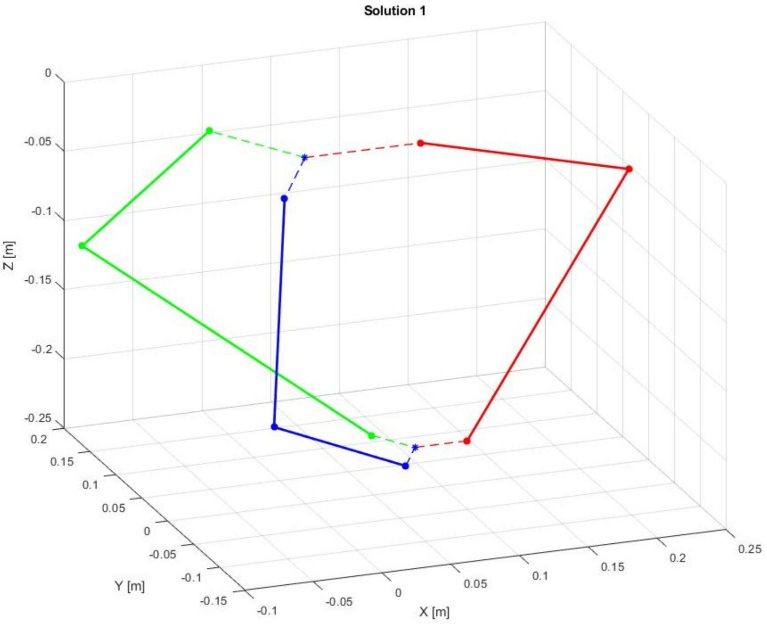


Figure 3.9: Graphical representation of a IK solution provided by MATLAB script

Up to now, it has not been taken into consideration the possibility that one or more of the IK solutions could be not feasible. Moreover, it could happen that the square root term in (3.47) is negative, in these cases we would end up with a complex term meaning that we have found a non feasible solution.
In particular, the inverse kinematics routine will provide a pair of *feasible* solutions for the *i-th* leg if and only if:

$$\frac{|G_i|}{\sqrt{E_i^2 + F_i^2}} \leq 1 \quad . \tag{3.50}$$

If this condition is met, then the Euclidean distance between $\overline{\mathbf{B}}_i$, which is the center of the circle with radius $L$, and $\mathbf{P}$, which is the center of the sphere with radius $l$, is less than or equal to $L + l$, and thus there can be a pair of solutions (possibly coincident).

### 3.1.3 Velocity Kinematics

The current section, together with the next one regarding the kinematics of acceleration, will be both based on the work done in the previous sections about position kinematics, and thus on [12] and [13]; however, it will be taken as reference also the work proposed by Guglielmetti and Longchamp in [14] and [15], that actually appears to be more elegant and clear for what concern the study of velocity and acceleration for the Delta robot.
We can start by recalling the vector-loop closure equations which allow to write the coordinates of the final point of the robot $\mathbf{P}$ in base frame $B$, thus, from (3.24) and (3.34) we can write, for the generic *i-th* leg:

$$\{\mathbf{P}\}_B = \{\mathbf{C}_i\}_B + \{\mathbf{P} - \mathbf{C}_i\}_B \quad . \tag{3.51}$$

It is straightforward to rewrite these equations into local coordinates $L_i$ by simply multiplication with the rotation matrix $\mathbf{R}_i^B$ defined in (3.6):

$$\{\mathbf{P}\}_i = \mathbf{R}_i^B\{\mathbf{P}\}_B = \mathbf{R}_i^B\left(\{\mathbf{C}_i\}_B + \{\mathbf{P} - \mathbf{C}_i\}_B\right) = \{\mathbf{C}_i\}_i + \{\mathbf{P} - \mathbf{C}_i\}_i \quad . \tag{3.52}$$

Then, by recalling (3.27) we can write $\mathbf{P}$ in local coordinates $L_i$ using the notation proposed by Guglielmetti and Longchamp:

$$\{\mathbf{P}\}_i = \{\mathbf{C}_i\}_i + \{\mathbf{P} - \mathbf{C}_i\}_i = \begin{bmatrix} \overline{R} \\ 0 \\ 0 \end{bmatrix} + L\,\boldsymbol{\gamma}_i + l\,\boldsymbol{\beta}_i \tag{3.53}$$

where:

$$\boldsymbol{\gamma}_i = \begin{bmatrix} \cos(\theta_i) \\ 0 \\ -\sin(\theta_i) \end{bmatrix} \quad \textit{and} \quad \boldsymbol{\beta}_i = \frac{1}{l}\left(\{\mathbf{P} - \mathbf{C}_i\}_i\right) \quad \textit{where} \;\; i = \{1,2,3\}\ . \tag{3.54}$$

It can be immediately seen that the *unitary* vectors just introduced, $\boldsymbol{\gamma}_i$ and $\boldsymbol{\beta}_i$, are, respectively, the unit vector with direction of the *i-th* upper link $\{\mathbf{C}_i - \overline{\mathbf{B}}_i\}_i$ and the unit vector with direction of the *i-th* lower link $\{\overline{\mathbf{P}}_i - \mathbf{C}_i\}_i$ both expressed in local frame $L_i$.
As expressed at the beginning of the chapter, the final point of the robot in base frame $B$ is $\mathbf{P} = [x, y, z]^\intercal$. However, after the radial shift proposed in the Kinematic Analysis section it is true that $\mathbf{P}$ is coincident with all $\overline{\mathbf{P}}_i$; moreover, we can now express the coordinates of all these coincident point in the *i-th* local frame $L_i$ as already seen:

$$\{\mathbf{P}\}_i = \mathbf{R}_i^B\{\mathbf{P}\}_B = \begin{bmatrix} {}^i x \\ {}^i y \\ {}^i z \end{bmatrix} \quad . \tag{3.55}$$

It should also be remarked that the $z$-coordinate of $\{\mathbf{P}\}$, expressed either in $B$ or in one local frame $L_i$, is always *negative* since we want the robot to work below the fixed base.

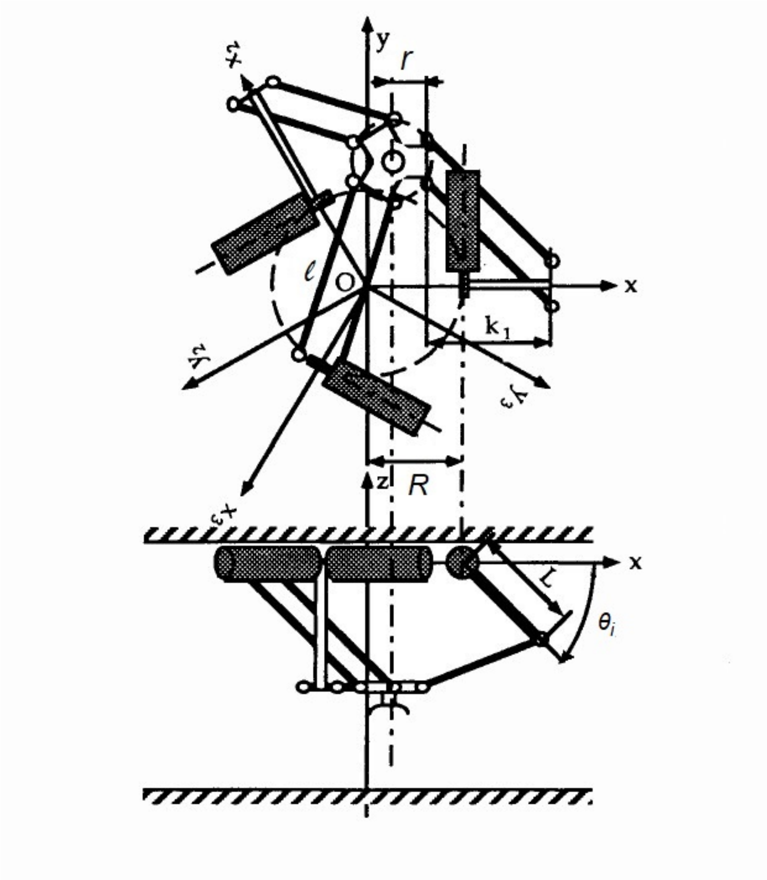


Figure 3.10: Lower and side views of a Delta robot (image from [14])

Keeping as reference the image displayed in figure 3.10, taken by [14] and adapted to our needs, we can denote by $k_i$ the $x$-coordinate of the *i-th* lower link vector $\{\overline{\mathbf{P}}_i - \mathbf{C}_i\}$ and we can now express this vector as:

$$\{\mathbf{P} - \mathbf{C}_i\}_i = \begin{bmatrix} l_x \\ l_y \\ l_z \end{bmatrix} = \begin{bmatrix} -k_i \\ {}^iy \\ {}^iz + L\sin(\theta_i) \end{bmatrix} \tag{3.56}$$

where $k_i$ is a positive term and ${}^iz + L\sin(\theta_i)$ is *always* negative since $|{}^iz| > |L\sin(\theta_i)|$ due to construction constraints.
Exploiting the norm of $\{\mathbf{P} - \mathbf{C}_i\}$ as done also in relation (3.33) we can find $k_i$ as:

$$k_i = \sqrt{l^2 - {}^iy^2 - ({}^iz + L\sin(\theta_i))^2} \quad . \tag{3.57}$$

Finally, we can write the $x$-coordinate of $\mathbf{P}$ in local frame $L_i$ in this way:

$$\begin{aligned} {}^{i}x &= R + L\cos(\theta_i) - k_i - r \\ &= \overline{R} + L\cos(\theta_i) - k_i \ . \end{aligned} \tag{3.58}$$

At this point we are ready to find the kinematics equations of velocity for the Delta robot: assuming that we have already computed both the joint angles $\boldsymbol{\theta}$ and the end point position $\mathbf{P}$ from position kinematics we can compute the time derive of (3.58), plugging (3.57) into it and then obtain the *i-th* line of the velocity kinematics which results to be:

$${}^{i}\dot{x} = \frac{1}{k_i}\left[L\left((L\sin(\theta_i) + {}^{i}z)\cos(\theta_i) - k_i\sin(\theta_i)\right)\dot{\theta}_i + {}^{i}y\,{}^{i}\dot{y} + (L\sin(\theta_i) + {}^{i}z)\,{}^{i}\dot{z}\right] \ . \tag{3.59}$$

Recalling that these three equations are written in local coordinates $L_i$ with $i = \{1, 2, 3\}$ and remembering that we can transform them into base frame $B$ by the already seen rotations matrices defined in (3.4), we can write, after some computations not shown here, the three kinematics equations of velocity for the Delta robot:

$$\begin{cases} -k_1\dot{x} + y\dot{y} + (L\sin\theta_1 + z)\dot{z} = L\left[k_1\sin\theta_1 - (L\sin\theta_1 + z)\cos\theta_1\right]\dot{\theta}_1 \\ \dfrac{2k_2 + 3x + \sqrt{3}y}{4}\dot{x} + \dfrac{\sqrt{3}x + y - 2k_2\sqrt{3}}{4}\dot{y} + (L\sin\theta_2 + z)\dot{z} = \\ \qquad\qquad L\left[k_2\sin\theta_2 - (L\sin\theta_2 + z)\cos\theta_2\right]\dot{\theta}_2 \\ \dfrac{2k_3 + 3x - \sqrt{3}y}{4}\dot{x} + \dfrac{2k_3\sqrt{3} - \sqrt{3}x + y}{4}\dot{y} + (L\sin\theta_3 + z)\dot{z} = \\ \qquad\qquad L\left[k_3\sin\theta_3 - (L\sin\theta_3 + z)\cos\theta_3\right]\dot{\theta}_3 \end{cases} \tag{3.60}$$

which can be written in matrix form as:

$$\mathbf{M}(\mathbf{P}, \boldsymbol{\theta})\dot{\mathbf{P}} = \mathbf{V}(z, \boldsymbol{\theta})\dot{\boldsymbol{\theta}} \quad . \tag{3.61}$$

Where $\dot{\mathbf{P}}$ and $\dot{\boldsymbol{\theta}}$ are:

$$\dot{\mathbf{P}} = \{\dot{\mathbf{P}}\}_B = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \quad , \quad \dot{\boldsymbol{\theta}} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \tag{3.62}$$

and the two matrices just introduced are defined as follows:

$$\mathbf{M}(\mathbf{P}, \boldsymbol{\theta}) = \begin{bmatrix} -k_1 & y & L\sin(\theta_1) + z \\ \dfrac{2k_2 + 3x + \sqrt{3}y}{4} & \dfrac{\sqrt{3}x + y - 2k_2\sqrt{3}}{4} & L\sin(\theta_2) + z \\ \dfrac{2k_3 + 3x - \sqrt{3}y}{4} & \dfrac{2k_3\sqrt{3} - \sqrt{3}x + y}{4} & L\sin(\theta_3) + z \end{bmatrix} \tag{3.63}$$

$$\mathbf{V}(z,\boldsymbol{\theta}) = diag\big(L\,[k_i \sin(\theta_i) - (L\sin(\theta_i)+z)\cos(\theta_i)]\,\big),\;\; i=\{1,2,3\}\;. \qquad (3.64)$$

Moreover, by looking at $\mathbf{M}(\mathbf{P},\boldsymbol{\theta})$ and $\mathbf{V}(z,\boldsymbol{\theta})$ it is useful to introduce two new matrices $\overline{\mathbf{M}}(\mathbf{P},\boldsymbol{\theta})$ and $\overline{\mathbf{V}}(z,\boldsymbol{\theta})$ which will be used in future computations. Concerning this, the following relations hold:

$$\mathbf{M}(\mathbf{P},\boldsymbol{\theta}) = l \begin{bmatrix} \boldsymbol{\beta}_1^\intercal \mathbf{R}_1^B \\ \boldsymbol{\beta}_2^\intercal \mathbf{R}_2^B \\ \boldsymbol{\beta}_3^\intercal \mathbf{R}_3^B \end{bmatrix} = l \begin{bmatrix} \mathbf{R}_B^1 \boldsymbol{\beta}_1 & \mathbf{R}_B^2 \boldsymbol{\beta}_2 & \mathbf{R}_B^3 \boldsymbol{\beta}_3 \end{bmatrix}^\intercal =: l\,\overline{\mathbf{M}}(\mathbf{P},\boldsymbol{\theta}) \qquad (3.65)$$

$$\mathbf{V}(z,\boldsymbol{\theta}) = lL \begin{bmatrix} \boldsymbol{\beta}_1^\intercal \overline{\boldsymbol{\gamma}}_1 & 0 & 0 \\ 0 & \boldsymbol{\beta}_2^\intercal \overline{\boldsymbol{\gamma}}_2 & 0 \\ 0 & 0 & \boldsymbol{\beta}_3^\intercal \overline{\boldsymbol{\gamma}}_3 \end{bmatrix} =: lL\,\overline{\mathbf{V}}(z,\boldsymbol{\theta}) \qquad (3.66)$$

where $\overline{\boldsymbol{\gamma}}_i$ is defined as the partial derivative of $\boldsymbol{\gamma}_i$ with respect to $\theta_i$, so, from (3.54) we can write:

$$\overline{\boldsymbol{\gamma}}_i = \frac{\partial \boldsymbol{\gamma}_i}{\partial \theta_i} = \begin{bmatrix} -\sin(\theta_i) \\ 0 \\ -\cos(\theta_i) \end{bmatrix}\;. \qquad (3.67)$$

The two introduced relations (3.65), (3.66) can be verified by looking at (3.54) and (3.56) and by recalling the rotation matrices defined in (3.6) and (3.4); their proof is not reported here since not strictly necessary for the aims of this project. As last important remark, it must be outlined that the so-called *Jacobian* of the Delta robot is:

$$\mathbf{J}(\mathbf{P},\boldsymbol{\theta}) = \mathbf{M}(\mathbf{P},\boldsymbol{\theta})^{-1}\,\mathbf{V}(z,\boldsymbol{\theta}) \qquad (3.68)$$

and its inverse:

$$\mathbf{J}(\mathbf{P},\boldsymbol{\theta})^{-1} = \mathbf{V}(z,\boldsymbol{\theta})^{-1}\,\mathbf{M}(\mathbf{P},\boldsymbol{\theta}) \qquad (3.69)$$

The Jacobian is in fact the matrix representing the *linear* mapping between end-effector velocities (in Cartesian space) and joint velocities (in joint space) and viceversa. It provides a lot of information about velocities, forces and singularities too; for these and other reasons it is greatly exploited in robotics field and it will be used to write the equation of the inverse velocity kinematics of the Delta robot, so, from (3.61):

$$\dot{\boldsymbol{\theta}} = \mathbf{V}(z,\boldsymbol{\theta})^{-1}\,\mathbf{M}(\mathbf{P},\boldsymbol{\theta})\,\dot{\mathbf{P}} = \mathbf{J}(\mathbf{P},\boldsymbol{\theta})^{-1}\,\dot{\mathbf{P}} \qquad (3.70)$$

that expresses joint velocities as linear combinations of end-point velocities.

### 3.1.4 Acceleration Kinematics

Following the reasoning and computations developed in the previous section, we can now derive the relationships between work space and joint space accelerations by the time derivative of (3.61), which is:

$$\dot{\mathbf{M}}\,\dot{\mathbf{P}} + \mathbf{M}\,\ddot{\mathbf{P}} = \dot{\mathbf{V}}\,\dot{\boldsymbol{\theta}} + \mathbf{V}\,\ddot{\boldsymbol{\theta}} \tag{3.71}$$

then, it is necessary to find the components of the first unknown vector, $\dot{\mathbf{M}}\,\dot{\mathbf{P}}$; we can thus start by writing, from (3.65):

$$\dot{\mathbf{M}} = l \begin{bmatrix} \dot{\boldsymbol{\beta}}_1^{\mathsf{T}}\mathbf{R}_1^B \\ \dot{\boldsymbol{\beta}}_2^{\mathsf{T}}\mathbf{R}_2^B \\ \dot{\boldsymbol{\beta}}_3^{\mathsf{T}}\mathbf{R}_3^B \end{bmatrix} \ . \tag{3.72}$$

It is also possible to define $\{\mathbf{C}_i\}_i$ by looking at (3.53) as:

$$\{\mathbf{C}_i\}_i = \begin{bmatrix} R \\ 0 \\ 0 \end{bmatrix} + L\boldsymbol{\gamma}_i \tag{3.73}$$

and thus, by referring also to the definition of $\boldsymbol{\beta}_i$ reported in (3.54), the time derivative of the generic $\dot{\boldsymbol{\beta}}_i$ becomes:

$$\dot{\boldsymbol{\beta}}_i = \frac{1}{l}\left(\{\dot{\mathbf{P}} - \dot{\mathbf{C}}_i\}_i\right) = \frac{1}{l}\left(\{\dot{\mathbf{P}}\}_i - L\overline{\boldsymbol{\gamma}}_i\dot{\theta}_i\right) \ . \tag{3.74}$$

We are now ready to write the *i-th* component of the vector $\dot{\mathbf{M}}\,\dot{\mathbf{P}}$ as:

$$\left(\dot{\mathbf{M}}\,\dot{\mathbf{P}}\right)_i = l\dot{\boldsymbol{\beta}}_i^{\mathsf{T}}\mathbf{R}_i^B\dot{\mathbf{P}} = \left(\{\dot{\mathbf{P}}\}_i - L\dot{\theta}_i\overline{\boldsymbol{\gamma}}_i\right)^{\mathsf{T}}\mathbf{R}_i^B\dot{\mathbf{P}} \ . \tag{3.75}$$

But it has already been underlined that all $\overline{\mathbf{P}}_i$ are coincident with $\mathbf{P}$ in base frame $B$ and $\{\mathbf{P}\} = \{\overline{\mathbf{P}}_i\} = \mathbf{R}_B^i\{\overline{\mathbf{P}}_i\}_i$, so it is true that, for any choice of $i$:

$$\{\mathbf{P}\} = \mathbf{R}_B^i\{\overline{\mathbf{P}}_i\}_i \Rightarrow \{\dot{\mathbf{P}}\} = \dot{\mathbf{P}} = \dot{\mathbf{R}}_B^i\,\{\mathbf{P}\}_i + \mathbf{R}_B^i\{\dot{\mathbf{P}}\}_i = \mathbf{R}_B^i\{\dot{\mathbf{P}}\}_i \tag{3.76}$$

and by transposing both sides:

$$\dot{\mathbf{P}}^{\mathsf{T}} = \left(\mathbf{R}_B^i\{\dot{\mathbf{P}}\}_i\right)^{\mathsf{T}} \Rightarrow \dot{\mathbf{P}}^{\mathsf{T}} = \left(\{\dot{\mathbf{P}}\}_i\right)^{\mathsf{T}}\left(\mathbf{R}_B^i\right)^{\mathsf{T}} \Rightarrow \dot{\mathbf{P}}^{\mathsf{T}} = \left(\{\dot{\mathbf{P}}\}_i\right)^{\mathsf{T}}\mathbf{R}_i^B \ . \tag{3.77}$$

Rewriting (3.75):

$$\begin{aligned} \left(\dot{\mathbf{M}}\,\dot{\mathbf{P}}\right)_i &= \left(\{\dot{\mathbf{P}}\}_i^{\mathsf{T}}\mathbf{R}_i^B - L\dot{\theta}_i\overline{\boldsymbol{\gamma}}_i^{\mathsf{T}}\mathbf{R}_i^B\right)\dot{\mathbf{P}} \\ &= \dot{\mathbf{P}}^{\mathsf{T}}\dot{\mathbf{P}} - L\dot{\theta}_i\overline{\boldsymbol{\gamma}}_i^{\mathsf{T}}\mathbf{R}_i^B\dot{\mathbf{P}} \ . \end{aligned} \tag{3.78}$$

It is now useful to introduce the following diagonal matrix:

$$\dot{\tilde{\boldsymbol{\theta}}} = diag(\dot{\theta}_i)_{3x3} = \begin{bmatrix} \dot{\theta}_1 & 0 & 0 \\ 0 & \dot{\theta}_2 & 0 \\ 0 & 0 & \dot{\theta}_3 \end{bmatrix} \tag{3.79}$$

and it is finally possible to write the overall $\dot{\mathbf{M}}\dot{\mathbf{P}}$ vector, so that it is:

$$\begin{aligned} \dot{\mathbf{M}}\dot{\mathbf{P}} &= \begin{bmatrix} (\dot{\mathbf{M}}\dot{\mathbf{P}})_1 \\ (\dot{\mathbf{M}}\dot{\mathbf{P}})_2 \\ (\dot{\mathbf{M}}\dot{\mathbf{P}})_3 \end{bmatrix} \\ &= \dot{\mathbf{P}}^{\mathsf{T}}\dot{\mathbf{P}} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - L\dot{\tilde{\boldsymbol{\theta}}} \begin{bmatrix} \overline{\boldsymbol{\gamma}}_1^{\mathsf{T}}\mathbf{R}_1^B \\ \overline{\boldsymbol{\gamma}}_2^{\mathsf{T}}\mathbf{R}_2^B \\ \overline{\boldsymbol{\gamma}}_3^{\mathsf{T}}\mathbf{R}_3^B \end{bmatrix} \dot{\mathbf{P}} \\ &= \dot{\mathbf{P}}^{\mathsf{T}}\dot{\mathbf{P}} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - L\dot{\tilde{\boldsymbol{\theta}}}\overline{\mathbf{A}}\dot{\mathbf{P}} \end{aligned} \tag{3.80}$$

where $\overline{\mathbf{A}}$ is:

$$\overline{\mathbf{A}} = \begin{bmatrix} \overline{\boldsymbol{\gamma}}_1^{\mathsf{T}}\mathbf{R}_1^B \\ \overline{\boldsymbol{\gamma}}_2^{\mathsf{T}}\mathbf{R}_2^B \\ \overline{\boldsymbol{\gamma}}_3^{\mathsf{T}}\mathbf{R}_3^B \end{bmatrix} . \tag{3.81}$$

It is now the moment to analyze the other unknown vector, $\dot{\mathbf{V}}\,\dot{\boldsymbol{\theta}}$; we can start by deriving the value of the *i-th* element of $\dot{\mathbf{V}}\,\dot{\boldsymbol{\theta}}$, that is:

$$\left(\dot{\mathbf{V}}\,\dot{\boldsymbol{\theta}}\right)_i = lL\,\frac{d(\boldsymbol{\beta}_i^{\mathsf{T}}\overline{\boldsymbol{\gamma}}_i)}{dt}\,\dot{\boldsymbol{\theta}} \tag{3.82}$$

where the term $\frac{d(\boldsymbol{\beta}_i^{\mathsf{T}}\overline{\boldsymbol{\gamma}}_i)}{dt}$ can be rewritten as:

$$\frac{d(\boldsymbol{\beta}_i^{\mathsf{T}}\overline{\boldsymbol{\gamma}}_i)}{dt} = \dot{\boldsymbol{\beta}}_i^{\mathsf{T}}\overline{\boldsymbol{\gamma}}_i + \boldsymbol{\beta}_i^{\mathsf{T}}\dot{\overline{\boldsymbol{\gamma}}}_i . \tag{3.83}$$

But looking in more details at $\dot{\overline{\boldsymbol{\gamma}}}_i$ and recalling (3.54) and (3.67), we can notice that:

$$\dot{\overline{\boldsymbol{\gamma}}}_i = \frac{\partial\overline{\boldsymbol{\gamma}}_i}{\partial\theta_i}\frac{\partial\theta_i}{\partial t} = \begin{bmatrix} -\cos(\theta_i) \\ 0 \\ \sin(\theta_i) \end{bmatrix} \dot{\theta}_i = -\dot{\theta}_i\boldsymbol{\gamma}_i . \tag{3.84}$$

Rewriting consequently (3.82):

$$\left(\dot{\mathbf{V}}\,\dot{\boldsymbol{\theta}}\right)_i = lL\left(\dot{\boldsymbol{\beta}}_i^{\mathsf{T}}\overline{\boldsymbol{\gamma}}_i - \dot{\theta}_i\boldsymbol{\gamma}_i^{\mathsf{T}}\boldsymbol{\beta}_i\right)\dot{\boldsymbol{\theta}} \tag{3.85}$$

and we can finally express the entire vector $\dot{\mathbf{V}}\,\dot{\boldsymbol{\theta}}$ as:

$$
\begin{aligned}
\dot{\mathbf{V}}\,\dot{\boldsymbol{\theta}} &= lL\left(\begin{bmatrix}\dot{\boldsymbol{\beta}}_1^\mathsf{T}\overline{\boldsymbol{\gamma}}_1 & 0 & 0\\ 0 & \dot{\boldsymbol{\beta}}_2^\mathsf{T}\overline{\boldsymbol{\gamma}}_2 & 0\\ 0 & 0 & \dot{\boldsymbol{\beta}}_3^\mathsf{T}\overline{\boldsymbol{\gamma}}_3\end{bmatrix} - \dot{\tilde{\boldsymbol{\theta}}}\begin{bmatrix}\boldsymbol{\beta}_1^\mathsf{T}\boldsymbol{\gamma}_1 & 0 & 0\\ 0 & \boldsymbol{\beta}_2^\mathsf{T}\boldsymbol{\gamma}_2 & 0\\ 0 & 0 & \boldsymbol{\beta}_3^\mathsf{T}\boldsymbol{\gamma}_3\end{bmatrix}\right)\dot{\boldsymbol{\theta}}\\
&= lL\left(\begin{bmatrix}\dot{\boldsymbol{\beta}}_1^\mathsf{T}\overline{\boldsymbol{\gamma}}_1 & 0 & 0\\ 0 & \dot{\boldsymbol{\beta}}_2^\mathsf{T}\overline{\boldsymbol{\gamma}}_2 & 0\\ 0 & 0 & \dot{\boldsymbol{\beta}}_3^\mathsf{T}\overline{\boldsymbol{\gamma}}_3\end{bmatrix}\dot{\boldsymbol{\theta}} - \begin{bmatrix}\boldsymbol{\beta}_1^\mathsf{T}\boldsymbol{\gamma}_1 & 0 & 0\\ 0 & \boldsymbol{\beta}_2^\mathsf{T}\boldsymbol{\gamma}_2 & 0\\ 0 & 0 & \boldsymbol{\beta}_3^\mathsf{T}\boldsymbol{\gamma}_3\end{bmatrix}\dot{\boldsymbol{\theta}}^2\right) \qquad (3.86)\\
&= lL\left(\overline{\mathbf{B}}_1\dot{\boldsymbol{\theta}} - \overline{\mathbf{B}}_2\dot{\boldsymbol{\theta}}^2\right)
\end{aligned}
$$

where:

$$
\dot{\boldsymbol{\theta}}^2 = \begin{bmatrix}\theta_1^2\\ \theta_2^2\\ \theta_3^2\end{bmatrix}\,. \qquad (3.87)
$$

We are now ready to write the inverse kinematics equations for the acceleration of the Delta robot, which are the relationships providing joint angles accelerations as function of: end-point accelerations, joint angles and end-point velocities, joint and end-point positions, i.e., $\ddot{\boldsymbol{\theta}} = f(\ddot{\mathbf{P}}, \dot{\boldsymbol{\theta}}, \dot{\mathbf{P}}, \boldsymbol{\theta}, \mathbf{P})$.

By recalling the matrix equations of the acceleration kinematics introduced in (3.71) and also relations (3.65), (3.66), (3.80) and (3.86), the IK equations for acceleration in matrix form become:

$$
\begin{aligned}
\ddot{\boldsymbol{\theta}} &= \mathbf{V}^{-1}\left[\mathbf{M}\ddot{\mathbf{P}} + \dot{\mathbf{M}}\dot{\mathbf{P}} - \dot{\mathbf{V}}\dot{\boldsymbol{\theta}}\right]\\
&= \frac{1}{lL}\overline{\mathbf{V}}^{-1}\left[l\overline{\mathbf{M}}\ddot{\mathbf{P}} + \dot{\mathbf{P}}^\mathsf{T}\dot{\mathbf{P}}\begin{bmatrix}1\\1\\1\end{bmatrix} - L\dot{\tilde{\boldsymbol{\theta}}}\overline{\mathbf{A}}\dot{\mathbf{P}} - lL\overline{\mathbf{B}}_1\dot{\boldsymbol{\theta}} + lL\overline{\mathbf{B}}_2\dot{\boldsymbol{\theta}}^2\right] \qquad (3.88)\\
&= \overline{\mathbf{V}}^{-1}\left[\frac{1}{L}\overline{\mathbf{M}}\ddot{\mathbf{P}} + \frac{1}{lL}\dot{\mathbf{P}}^\mathsf{T}\dot{\mathbf{P}}\begin{bmatrix}1\\1\\1\end{bmatrix} - \frac{1}{l}\dot{\tilde{\boldsymbol{\theta}}}\overline{\mathbf{A}}\dot{\mathbf{P}} - \overline{\mathbf{B}}_1\dot{\boldsymbol{\theta}} + \overline{\mathbf{B}}_2\dot{\boldsymbol{\theta}}^2\right]
\end{aligned}
$$

## 3.2 Dynamics

The study of system dynamics is a crucial point in the study of a robot, the results obtained here will in fact be used in the design phase of the robot, during the motor sizing and also in the control development. It is thus immediately clear that obtaining a truthful and descriptive dynamic model is a key step for every project that aims at achieving good results.
The work carried on in this chapter is mostly based on the paper written by Brinker et al. [16], it provides a large overview of the different dynamics modelling approaches adopted in the Delta robot study.
The authors distinguish between *complete* and *simplified* dynamics models; the latter are obviously simpler since some dynamics effects are not taken into account; the main approximation of these model consists in neglecting the lower link's rotational inertia. However, the authors state that these approximate models suffer for errors in torque up to 10% of the real value while they have similar processing time with respect to complete models (these latter are only 1.6 time more computational consuming). From these considerations it has been reasonably decided to go for a complete dynamic modelling.
As stated in their work, the authors claim that complete dynamic models are almost always based on one of the following three approaches:

- Principle of Virtual Work
- Newton-Euler Formulation
- Lagrangian Formulation.

In the cited paper all of these three solutions have been developed and their time performance compared: results stated that Lagrangian approach is by far the fastest method. Taking as unitary time the one provided by Newton-Euler, the computing timings provided by the authors are reported in table 3.1.

| Model | Time (second) |
|---|---|
| Principle of Virtual Work | 0.73 |
| Newton-Euler Formulation | 1.00 |
| Lagrangian Formulation | 0.37 |

Table 3.1: Time comparison between different dynamic approaches

Taking into account that all these different approaches provide the same result, *Lagrangian Formulation* is the best choice and it will be now presented.
Theory behind Lagrangian mechanics and Lagrangian multipliers is not presented here and the interested reader is remanded to standard mechanics books.

Schematically, we will proceed as follows: firstly, kinetic and potential energies of each body will be derived in order to obtain Lagrangian equations, then, the constraint equations will be obtained by imposing that the distance between each joint connecting upper and lower links and the ones connecting lower links and the movable plate is equal to the length of each lower link, i.e., $l$. Finally, the partial derivatives are derived to finally calculate the required actuation torques.

## Lagrangian Equations for the Delta Robot

The Lagrange equations are derived by considering the total kinetic energy $K$ and the total potential energy $U$ of the Delta robot. In particular, the Lagrangian function is defined by the well know relation:

$$\mathcal{L} = K - U \; . \tag{3.89}$$

The total kinetic energy $K$ for the generic Delta robot can be written as follows:

$$K = K_P + \sum_{i=1}^{3} \Big( K_{U_i} + K_{L_i} \Big) \tag{3.90}$$

where $K_P$ is the kinetic energy owned by the movable plate, $K_{U_i}$ and $K_{L_i}$ are, respectively, the kinetic energy of the *i-th* upper and lower link.
The total potential energy $U$ can be instead written as:

$$U = U_P + \sum_{i=1}^{3} \Big( U_{U_i} + U_{L_i} \Big) \tag{3.91}$$

where the notation has the same meaning as before.
We can now start by deriving the single components of total kinetic energy as done in [16], thus:

$$K_P = \frac{1}{2} m_P \left( \dot{x}^2 + \dot{y}^2 + \dot{z}^2 \right) \tag{3.92}$$

$$K_{U_i} = \frac{1}{2} I_{upper_i} \dot{\theta}_i^2 = \frac{1}{2} \Big( I_{motor_i} + I_{U_i} \Big) \dot{\theta}_i^2 \tag{3.93}$$

$$K_{L_i} = \frac{1}{2} m_{L_i} \left[ \frac{1}{3} \left\| \left( {}^i\dot{\mathbf{p}} - {}^i\mathbf{v}_{U_i} \right) \right\|^2 \; {}^i\dot{\mathbf{p}} \cdot {}^i\mathbf{v}_{U_i} \right] \tag{3.94}$$

where $m_P$ is the mass of the movable plate, $m_{L_i}$ the mass of the *i-th* lower link, $I_{U_i}$ and $I_{motor_i}$ are, respectively, the moments of inertia of the *i-th* upper link and of the *i-th* motor, both with respect to their axis of rotation. Finally, ${}^i\mathbf{v}_{U_i}$ and ${}^i\dot{\mathbf{p}}$ are, respectively, the linear velocity vector of the upper link in its end-point $\{\mathbf{C}_i\}_i$ and the one of the movable plate in $\{\mathbf{P}_i\}_i$ where the left superscript indicates that are both projected in the local reference frame $L_i$.

By defining as $\boldsymbol{\omega}_{U_i}$ the vector representing the angular velocity of the *i-th* upper link, it can be expressed in local coordinates as:

$$
{}^{i}\boldsymbol{\omega}_{U_i} = \dot{\boldsymbol{\theta}}_i = \theta_i \cdot \mathbf{y}^i \tag{3.95}
$$

and the velocity vector of point $\{\mathbf{C}_i\}_i$ in local frame can be rewritten as:

$$
\begin{aligned}
{}^{i}\mathbf{v}_{U_i} &= {}^{i}\boldsymbol{\omega}_{U_i} \times {}^{i}\mathbf{L}_{U_i} \\
&= \dot{\theta}_i \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \times \begin{bmatrix} L\cos(\theta_i) \\ 0 \\ -L\sin(\theta_i) \end{bmatrix} = \dot{\theta}_i \begin{bmatrix} -L\sin(\theta_i) \\ 0 \\ -L\cos(\theta_i) \end{bmatrix}
\end{aligned} \tag{3.96}
$$

while the vector of the velocity of $\{\mathbf{P}_i\}_i$ in local coordinates can be written as:

$$
{}^{i}\dot{\mathbf{p}} = \begin{bmatrix} {}^{i}\dot{x} \\ {}^{i}\dot{y} \\ {}^{i}\dot{z} \end{bmatrix} \;. \tag{3.97}
$$

Now, equation (3.94) about $K_{L_i}$ can be developed as follows:

$$
\begin{aligned}
K_{L_i} &= \frac{1}{2} m_{L_i} \left[ \frac{1}{3} \left\| \left( \begin{bmatrix} {}^{i}\dot{x} \\ {}^{i}\dot{y} \\ {}^{i}\dot{z} \end{bmatrix} - \dot{\theta}_i \begin{bmatrix} -L\sin(\theta_i) \\ 0 \\ -L\cos(\theta_i) \end{bmatrix} \right) \right\|^2 + \begin{bmatrix} {}^{i}\dot{x} \\ {}^{i}\dot{y} \\ {}^{i}\dot{z} \end{bmatrix}^{\mathsf{T}} \cdot \begin{bmatrix} -L\sin(\theta_i) \\ 0 \\ -L\cos(\theta_i) \end{bmatrix} \right] \\
&= \frac{1}{2} m_{L_i} \left[ \frac{1}{3} \left( \left( {}^{i}\dot{x} + \dot{\theta}_i L \sin(\theta_i) \right)^2 + {}^{i}\dot{y}^2 + \left( {}^{i}\dot{z} + \dot{\theta}_i L \cos(\theta_i) \right)^2 \right) - \right. \\
&\qquad \left. - \left( {}^{i}\dot{x}\dot{\theta}_i L \sin(\theta_i) + {}^{i}\dot{z}\dot{\theta}_i L \cos(\theta_i) \right) \right]
\end{aligned} \tag{3.98}
$$

and we can rewrite $K_{L_i}$ in base coordinates by remembering that each local frame is simply rotated by an angle $\alpha_i$, thus:

$$
\begin{aligned}
K_{L_i} = \frac{1}{6} & \left[ \dot{x}^2 + \dot{y}^2 + \dot{z}^2 + L^2\dot{\theta}_i^2 - \dot{x}\dot{\theta}_i L \cos(\alpha_i) \sin(\theta_i) - \right. \\
& \left. - \dot{y}\dot{\theta}_i L \sin(\alpha_i) \sin(\theta_i) - \dot{z}\dot{\theta}_i L \cos(\theta_i) \right] \;.
\end{aligned} \tag{3.99}
$$

It is now possible to write the terms of potential energy presented in (3.91) in the following way:

$$
U_P = m_P \cdot g \cdot z \tag{3.100}
$$

$$
U_{U_i} = -\frac{1}{2} m_{U_i} \cdot g \cdot L \cdot \sin(\theta_i) \tag{3.101}
$$

$$
U_{L_i} = \frac{1}{2} m_{L_i} \cdot g \cdot (z - L\sin(\theta_i)) \;. \tag{3.102}
$$

## Constraint Equations

To find the constraint equations we can start by writing the coordinates in base frame $B$ of the vector representing the lower link $\{\mathbf{P}_i - \mathbf{B}_i\}$ (equal to of $\{\overline{\mathbf{P}}_i - \overline{\mathbf{B}}_i\}$ since the same linear translation is applied to both points):

$$\{\mathbf{P}_i - \mathbf{B}_i\} = \begin{bmatrix} x - (R - r + L\cos(\theta_i))\cos(\alpha_i) \\ y - (R - r + L\cos(\theta_i))\sin(\alpha_i) \\ z + L\sin(\theta_i) \end{bmatrix} \tag{3.103}$$

then, by imposing that the magnitude (i.e., the length of the lower link) of this vector is equal to $l$ we can find the 3 constraint equations we are looking for:

$$\mathit{\Gamma}_i = ||\{\mathbf{P}_i - \mathbf{B}_i\}||^2 - l^2 = 0 \qquad \mathit{where} \quad i = \{1, 2, 3\} \ . \tag{3.104}$$

Which, after some computations, can be rewritten as:

$$\begin{split}\mathit{\Gamma}_i = x^2 + y^2 + z^2 + R^2 + r^2 + L^2 - l^2 - 2Rr\,(1 - L\cos(\theta_i)) - 2rL\cos(\theta_i) - \\ - 2x\,(R - r + L\cos(\theta_i))\cos(\alpha_i) - 2y\,(R - r + L\cos(\theta_i))\sin(\alpha_i) + 2zL\sin(\theta_i) \ .\end{split} \tag{3.105}$$

## Partial Derivatives

As also confirmed by the previously written Lagrange equations, it can be stated that the system has 6 *generalized coordinates*, which are:

$$\mathbf{Q} = \begin{bmatrix} x & y & z & \theta_1 & \theta_2 & \theta_3 \end{bmatrix} \ . \tag{3.106}$$

Moreover, it should be noticed that only 3 of them are really independent while the other 3 are just redundant generalized coordinates. The occurrence of only three DOF can be noticed by the fact that we have found exactly three constraint equations that relate $x, y, z$ to $\theta_1, \theta_2, \theta_3$.

Partial derivatives of the kinetic and potential energy terms presented in the previous section are now shown, but computations are not reported for reasons of brevity:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{x}}\right) = m_P\ddot{x} + \sum_{i=1}^{3}\left(\frac{1}{3}m_{L_i}\ddot{x} - \frac{1}{6}m_{L_i}L\cos(\alpha_i)\left(\ddot{\theta}_i\sin(\theta_i) + \dot{\theta}_i^2\cos(\theta_i)\right)\right) \tag{3.107}$$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{y}}\right) = m_P\ddot{y} + \sum_{i=1}^{3}\left(\frac{1}{3}m_{L_i}\ddot{y} - \frac{1}{6}m_{L_i}L\sin(\alpha_i)\left(\ddot{\theta}_i\sin(\theta_i) + \dot{\theta}_i^2\cos(\theta_i)\right)\right) \tag{3.108}$$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{z}}\right) = m_P \ddot{z} + \sum_{i=1}^{3}\left(\frac{1}{3}m_{L_i}\ddot{z} + \frac{1}{6}m_{L_i}L\left(\dot{\theta}_i^2\sin(\theta_i) - \ddot{\theta}_i\cos(\theta_i)\right)\right) \tag{3.109}$$

$$\frac{\partial \mathcal{L}}{\partial x} = 0 \tag{3.110}$$

$$\frac{\partial \mathcal{L}}{\partial y} = 0 \tag{3.111}$$

$$\frac{\partial \mathcal{L}}{\partial z} = -\,g\cdot\left(m_P + \frac{1}{2}\left(m_{L_1} + m_{L_2} + m_{L_3}\right)\right) \tag{3.112}$$

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}_i}\right) = I_{U_i}\ddot{\theta}_i - \frac{1}{6}m_{L_i}L\Big(&\cos(\alpha_i)\sin(\theta_i)\ddot{x} + \cos(\alpha_i)\cos(\theta_i)\dot{x}\dot{\theta}_i + \\ &+ \sin(\alpha_i)\sin(\theta_i)\ddot{y} + \sin(\alpha_i)\cos(\theta_i)\dot{y}\dot{\theta}_i + \cos(\theta_i)\ddot{z} - \sin(\theta_i)\dot{\theta}_i\dot{z} - 2L\ddot{\theta}_i\Big)\end{aligned} \tag{3.113}$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \theta_i} = &\frac{1}{2}gL\cos(\theta_i)\left(m_{U_i} + m_{L_i}\right) - \\ &- \frac{1}{6}m_{L_i}L\Big(\cos(\alpha_i)\cos(\theta_i)\dot{x}\dot{\theta}_i + \sin(\alpha_i)\cos(\theta_i)\dot{y}\dot{\theta}_i - \sin(\theta_i)\dot{z}\dot{\theta}_i\Big)\end{aligned} \tag{3.114}$$

where $i = \{1, 2, 3\}$ for the two last equations.

It is also necessary to compute the partial derivatives of the constraint equations with respect to the six generalized coordinates, which are:

$$\frac{\partial \mathit{\Gamma}_i}{\partial x} = 2x - 2\left(R - r + L\cos(\theta_i)\right)\cos(\alpha_i) \tag{3.115}$$

$$\frac{\partial \mathit{\Gamma}_i}{\partial y} = 2y - 2\left(R - r + L\cos(\theta_i)\right)\sin(\alpha_i) \tag{3.116}$$

$$\frac{\partial \mathit{\Gamma}_i}{\partial z} = 2z + 2L\sin(\theta_i) \tag{3.117}$$

$$\frac{\partial \mathit{\Gamma}_1}{\partial \theta_1} = 2L\left[\left(-R + r + x\cos(\alpha_1) + y\sin(\alpha_1)\right)\sin(\theta_1) + z\cos(\theta_1)\right] \tag{3.118}$$

$$\frac{\partial \mathit{\Gamma}_2}{\partial \theta_2} = 2L\left[\left(-R + r + x\cos(\alpha_2) + y\sin(\alpha_2)\right)\sin(\theta_1) + z\cos(\theta_2)\right] \tag{3.119}$$

$$\frac{\partial \mathit{\Gamma}_3}{\partial \theta_3} = 2L\left[\left(-R + r + x\cos(\alpha_3) + y\sin(\alpha_3)\right)\sin(\theta_3) + z\cos(\theta_3)\right] \tag{3.120}$$

$$\frac{\partial \mathit{\Gamma}_1}{\partial \theta_2} = \frac{\partial \mathit{\Gamma}_1}{\partial \theta_3} = \frac{\partial \mathit{\Gamma}_2}{\partial \theta_1} = \frac{\partial \mathit{\Gamma}_2}{\partial \theta_3} = \frac{\partial \mathit{\Gamma}_3}{\partial \theta_1} = \frac{\partial \mathit{\Gamma}_3}{\partial \theta_2} = 0\;. \tag{3.121}$$

## Actuation Torques

Finally, the Lagrange multipliers $\lambda_i$ can be obtained for a given state of motion by solving the following system of equations:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} = \sum_{i=1}^{3} \lambda_i \frac{\partial \mathit{\Gamma}_i}{\partial x} \tag{3.122}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{y}}\right) - \frac{\partial L}{\partial y} = \sum_{i=1}^{3} \lambda_i \frac{\partial \mathit{\Gamma}_i}{\partial y} \tag{3.123}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{z}}\right) - \frac{\partial L}{\partial z} = \sum_{i=1}^{3} \lambda_i \frac{\partial \mathit{\Gamma}_i}{\partial z} \tag{3.124}$$

and, after very long computations not shown here, we are now ready to compute the desired actuation torques $\tau_i$ with $i = \{1, 2, 3\}$ by the following equations:

$$\tau_i = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_i}\right) - \frac{\partial L}{\partial \theta_i} - \lambda_i \frac{\partial \mathit{\Gamma}_i}{\partial \theta_i} \quad \mathit{where} \;\; i = \{1, 2, 3\}\ . \tag{3.125}$$

# Chapter 4

# Robot Design

## Design Goals

After this extensive kinematics and dynamics analysis of the Delta robot it is now the time to focus on the design of such system. To successfully design a complex machine like a robotic platform it is fundamental that the goals of the project are very clear and are the ones with respect to which system parameters are optimized.
For these reasons, it appears to be important to proceed with a quick recap about the framework in which we are working and the consequent system requirements.
Firstly, it should be recalled that we wish our Delta robot to be mounted on an aerial platform whose payload capacity is very limited, thus it will be vital to reduce the amount of mass as much as possible; it will also be very important to reduce mass dispersion from the center of the aerial platform, this comes from the necessity of both trying to minimize center of mass offset and to reduce moments of inertia variations for the overall system.
Secondly, it should be reminded that unpredictable aerodynamic perturbations take place when MAV flies close to obstacles and surfaces and it will be then required to develop a manipulator able to sufficiently extend from the base.
It will be also necessary to design a Delta robot with a workspace large enough to allow to compensate for most of MAV's errors.
Last but not least, take-off and landing phases must be properly taken into account, this means that the designed system will have to be able to assume a compact configuration without any collision with MAV's components.

By considering also the other conditions presented in previous chapters (like the one about links ratio in 3.1), we can summarize goals *design* and *constraints* as follows:

- $l > L$
- desired end-effector accuracy $< 1$ cm

- max system weight: 0.750 kg
- mass dispersion reduced as much as possible
- max payload at end-effector: 0.2 kg
- desired workspace dimension: 20 × 20 × 20 cm (sufficiently distant from MAV)
- ability to assume compact configuration.

## 4.1 Geometric Parameters Synthesis

As mentioned in the previous chapter, a Delta robot is fully identified by four geometric parameters which can be grouped together in the so-called *set of geometric parameters* $I$, i.e.:

$$I = [\ L, l, R, r\ ]\ .$$

In a few words, sizing a Delta robot consists in finding the values for all the parameters of $I$.
In particular, we are interested in finding those parameters for which the resulting Delta robot satisfies the constraints just introduced. It should be noticed that the presented constraints affect different aspects of the robot design and thus only few of them can be directly taken into account in this design phase, for example the cap on weight can not be used right now but will be considered during material selection and sizing of specific components.

However, by a mathematical point of view, the number of *independent* parameters for a Delta robot is only 3 due to the fact that $R$ and $r$ are coupled together. In fact, if we look at the equations presented in previous chapter it can be noticed that the two terms do not appear alone but always grouped in the term $\overline{R}$, which was defined as:

$$\overline{R} = R - r\ .$$

We had in fact already introduced the set of *independent* parameters of the generic Delta robot as:

$$\overline{I} = \left[\ L, l, \overline{R}\ \right]\ .$$

## Workspace Analysis

The procedure to obtain the values of the geometric parameters for our Delta robot will be mostly based on the work proposed by Laribi et al. in [17]. The authors have presented a method for finding the *optimal* geometric parameters of a Delta robot for a given desired workspace. They formulate the synthesis task as an *optimization problem* whose solution is based on a so-called *genetic algorithm*. A genetic algorithm is an algorithm which tries to reflect the natural selection process where the fittest individuals are selected for reproduction in order to obtain offspring of the next generation. A schematic representation of genetic algorithm is presented in figure 4.1.

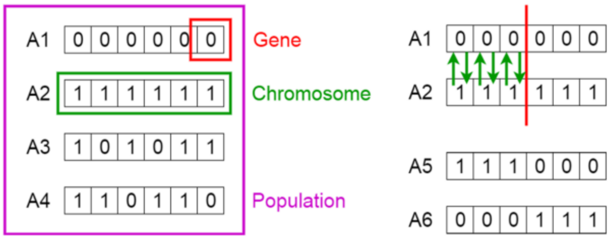


Figure 4.1: Genetic algorithm schematic representation (image from [18])

The algorithm starts with a predefined number of possible solutions (called *population*), each specific solution is referred to as a *chromosome* and it is a set of possible values of the optimization variables, a single variable value is called *gene*. Subsequently, a *fitness* value is assigned to each possible solution based on a objective function. The fittest individuals are then selected for reproduction and they generate offspring by means of exchanging some of their genes with the ones from another chromosome. Lastly, it could happen that in certain new offspring one or more of their genes can be randomly subjected to a mutation based on a predefined probability, this final step is adopted in order to prevent the algorithm from premature convergence into local minima.

As previously mentioned, Laribi et al. propose a method for the synthesis of a Delta robot for a desired input workspace; in particular, they show a mathematical approach to find the robot with the smallest workspace containing the desired volume.

If we now look back at the final shape of the three vector closure loop equations presented in (3.40) and to its associated solution, both reported here for clarity, and by considering that the workspace of a robot is a region of the three-dimensional Cartesian space made of those points which are attainable by the end-effector, it could be stated that the workspace of the $i-th$ Delta's leg can be obtained by imposing a non-negative sign condition on the square root term of the associated solution. This statement comes from the fact that a negative argument for the square root would mean that the specific point does not belong to the workspace.

$$E_i \cos(\theta_i) + F_i \sin(\theta_i) + G_i = 0$$
$$\Rightarrow t_{i_{1,2}} = \frac{-F_i \pm \sqrt{E_i^2 + F_i^2 - G_i^2}}{G_i - E_i} \; .$$

Imposing the square root term to be non negative, and using the same notation of Laribit et al. we get:

$$\varDelta_i = E_i^2 + F_i^2 - G_i^2 \geq 0 \qquad where \;\; i = \{1, 2, 3\} \tag{4.1}$$

$$G_i^2 - E_i^2 + F_i^2 = h_i(x, y, z) \leq 0 \; . \tag{4.2}$$

Inequality (4.2) represents a volume in space which is exactly the workspace of the *i-th* Delta's leg. The boundary of this workspace is instead identified by the corresponding equation, i.e.:

$$G_i^2 - E_i^2 + F_i^2 = h_i(x, y, z) = 0 \; . \tag{4.3}$$

Moreover, it should be noticed that this volume is expressed in base frame $B$ but it could be immediately rotated into the *i-th* local frame by the already seen rotations around the $\mathbf{z}^B$ axis. By recalling the expressions defining $E_i$, $F_i$ and $G_i$ in (3.43) and by considering the generic *i-th* rotation we can obtain the equations of this volume in its local coordinates as:

$$\overline{h}_i({}^i x, {}^i y, {}^i z) = \left[ \left( {}^i x - \overline{R} \right)^2 + {}^i y^2 + {}^i z^2 + L^2 - l^2 \right]^2 - 4L^2 \left[ \left( {}^i x - \overline{R} \right)^2 + {}^i z^2 \right]^2 \leq 0 \tag{4.4}$$

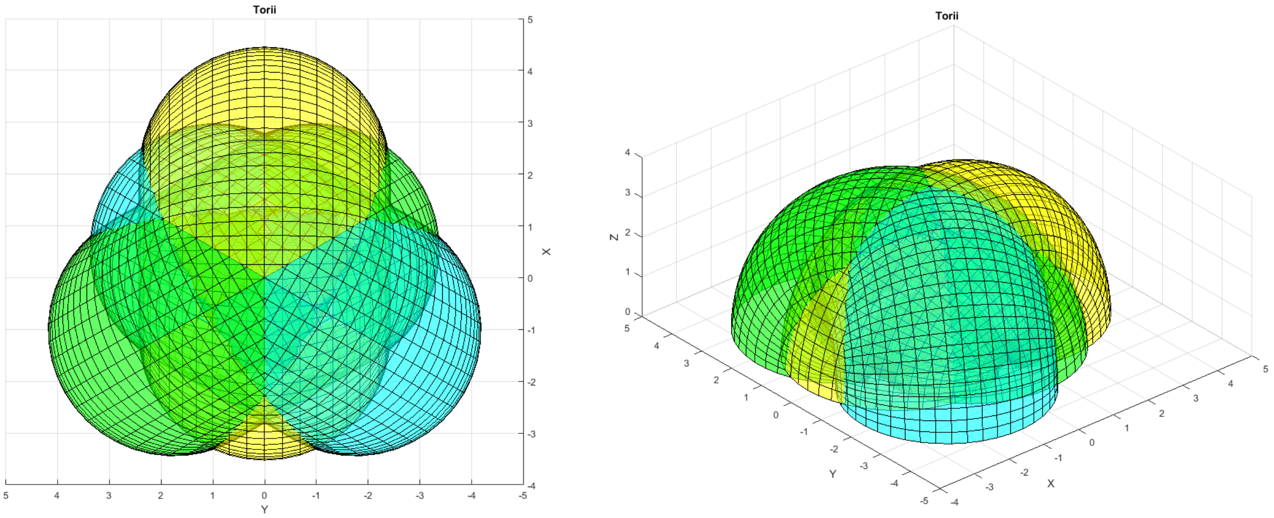
This is the equation of a *torus* having the characteristics presented in table 4.1.

| Characteristic | Local Frame | Base Frame |
|---|---|---|
| Center | $(\overline{R}, 0, 0)$ | $(\overline{R}\cos(\alpha_i), \overline{R}\sin(\alpha_i), 0)$ |
| Axis of revolution | $\mathbf{y}^i$ | $-\sin(\theta_i)\mathbf{x}^B + \cos(\theta_i)\mathbf{y}^B$ |
| External radius | $L$ | $L$ |
| Internal radius | $l$ | $l$ |

Table 4.1: Geometric characteristics of the *i-th* torus volume

Where the left superscripts indicate the frame with respect to which we are expressing point coordinates and right superscript the coordinate axes of a specific reference frame.

The workspace of a Delta robot is thus identified by the volume contained in the *intersection* of these three tori, which are all identical but just shifted by an angle



(a) Upper view

(b) Side view

Figure 4.2: 3 tori representing the workspace of the 3 Delta legs

of 120° with respect to the others. These toric surfaces can be seen in figure 4.2, only their upper halves are shown for the sake of clarity.

It is important to notice that, since we imposed from the beginning of our analysis $L < l$, the just presented tori belong to the class of the so-called *spindle* tori, i.e. the external radius is smaller than the internal one, this can be clearly seen in figure 4.3 which shows a $X - Y$ section of a generic spindle torus together with a 3D view of its lower half.

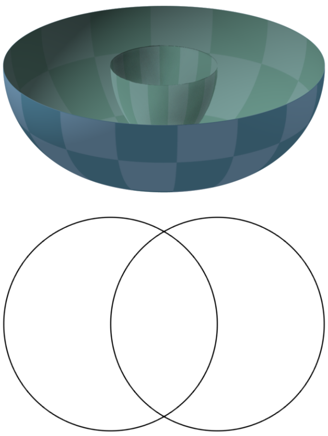


Figure 4.3: $X - Y$ section of a generic spindle torus and corresponding 3D view (image from Wikipedia)

In general, we will not take into consideration the upper half of these three tori since the movable plate can not move over the fixed base due to physical collisions between components; thus, for now on we will be interested in only lower halves of our tori.

As a last but very important consideration, it should be noticed that the sign assumed by $h_i(x_P, y_P, z_P)$ depends on the position of the considered point $P = (x_P, y_P, z_P)$ with respect to the *i-th* torus internal volume; by recalling that the workspace of Delta robot is the intersection of these three toric volumes, three different scenarios are possible:

- $P$ is outside the workspace: $h_i(x_P, y_P, z_P) > 0$ for some $i$ in $\{1, 2, 3\}$
- $P$ is on the boundary of the workspace: $h_i(x_P, y_P, z_P) \leq 0$ for all $i$ in $\{1, 2, 3\}$ and $h_i(x_P, y_P, z_P) = 0$ for some $i$ in $\{1, 2, 3\}$
- $P$ is inside the workspace: $h_i(x_P, y_P, z_P) < 0$ for all $i$ in $\{1, 2, 3\}$

## Problem Formulation

As presented in previous section, the goal of the synthesis phase is to obtain the values of the vector $I$, i.e., the geometric parameters of the Delta robot.

Following the approach proposed by Laribi et al., we can formulate this task as a non linear optimization problem whose aim is to find the values of $I$ for which the resulting workspace is the smallest one containing a given volume $W$.

The optimal dimensional synthesis problem can be thus formulated as follows:

Given: a desired volume $W$

Find: the geometric parameters of a Delta robot $I$ having the smallest workspace containing $W$.

It is thus clear that the first step toward the solution of this problem consists in defining this desired volume $W$.

By recalling design goals and constraints presented at the beginning of this chapter, the desired workspace volume can be chosen as a *cube* with dimension $20 \times 20 \times 20$ cm. Moreover, since we wish to impose some kind of conditions guaranteeing the ability of our Delta robot to assume a sufficiently compact configuration, it has been decided to add to the desired workspace an *extra point* with coordinate $(0.0, 0.0, -0.085)$ (in meters). This last condition guarantees that this point is inside the workspace, and thus that the robot can assume a configuration where its vertical displacement is only 8.5 centimeters. The selection of such a particular value comes from geometric considerations about the available space at the lower side of the MAV and ensures enough space for safe landings and take-offs.

Since our desired volume can be described by a set of *vertices* (the ones of the parallalepiped plus the extra point) which we will refer to as $P_k$, with coordinates $P_k = (x_{P_k}, y_{P_k}, z_{P_k})$, and by considering that the volume identified by these vertices is *convex*, it can be stated that $W$ is inside the workspace of the Delta robot if and only if:

$$h_i(I_H, P_k) \leq 0 \quad with \;\; i = \{1, 2, 3\} \;\; and \;\; k = \{1, ..., N_{pt}\} \tag{4.5}$$

where $I_H = [\, L, l, \overline{R}, H \,]$ is the set of three independent parameters plus a fourth term, $H$, which represents the distance along $z$-axis between Delta's fixed base and the upper face of the presented cube. This is in fact the last independent variable with respect to which we are going to solve the optimization problem and it is used to place the cube at a certain distance from the base in order to minimize as much as possible the aerodynamic perturbations that take place when an aerial platform flies close to objects.

A graphical representation of the desired workspace $W$, identified by its vertices, together with the torus volume of the *i-th* Delta's leg can be seen in figure 4.4.

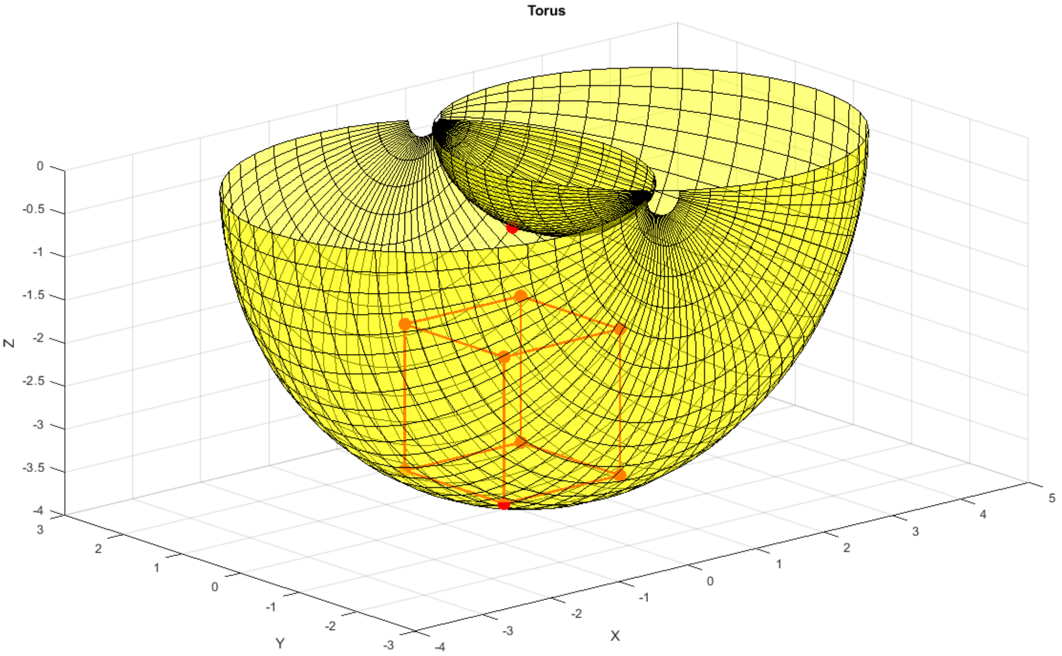


Figure 4.4: Delta's leg workspace together with verteces of $W$ (in red)

It can be noticed from this last image that the lower cube vertices, as well as the upper extra point, are positioned almost at the boundary of the workspace: this means that for that particular end-effector positions the system is in a singular, or almost singular, configuration, which is something that in general we want to avoid. For this reason, it has been decided to increase the size of the cube by 1 cm in each direction; the same procedure has been applied to the $z$-coordinate of the extra point, which has been placed 1 cm nearer to the fixed base. By adopting this *safety margins*, we are guaranteed that the resulting Delta robot will always work sufficiently far from any singular configuration.

Having precisely defined the desired workspace $W$, it is now the time to formulate the optimization problem. If we now recall the brief analysis on the sign of $h_i(I_H, P_k)$ made at the end of previous section, it could be noticed that $h_i(I_H, P_k)$ when applied to a point $P_k$ represents some kind of *distance* of that point to the *i-th* surface, in other words, the more $P_k$ is near to the boundary of the volume, the lower is its magnitude $|h_i(I_H, P_k)|$. The minimum value, which is 0, holds in fact for all points $P_k$ laying on the boundary of the surface. Therefore, minimizing $|h_i(I_H, P_k)|$ with respect to $I_H$ is equivalent to finding the closest surface to $P_k$. But since Delta's workspace is defined as the intersection of three volumes, what we want to minimize is instead:

$$P.o.P._{P_k} = |h_1(I_H, P_k)| + |h_2(I_H, P_k)| + |h_3(I_H, P_k)| \, . \tag{4.6}$$

This last term represents the *power of point* $P_k$ with respect to the three surfaces. The power of point is defined as the relative distance of a given point from a given circle and it is thus a measure of the distance of a point from a surface or planar figure.

By exploiting the concept of power of points and by taking as reference the work developed by Laribi et al., suitably adjusted to our needs, we can formulate the synthesis problem as follows:

$$\begin{array}{lll} \text{min} & F(I_H) & \\ \text{subject to} & h_i(I_H, P_k) \leq 0 & \text{for all the vertices } P_k \text{ of } W \end{array}$$

where $F(I_H)$ is the objective (or *fitness*) function and it is defined as the sum of two terms: $F = F_1 + F_2$, where $F_1$ is a *penalty* term which becomes very large in case at least one $P_k$ is outside the robot workspace and $F_2$ is exactly the power of point just introduce in (4.6), thus:

$$F(I_H, P_k) = F_1(I_H, P_k) + F_2(I_H, P_k) = F_1 + F_2 \tag{4.7}$$

where $F_1$ is a penalty term and it is defined as follows:

$$F_1(I_H, P_k) = \sum_{k=1}^{N_{pt}} \sum_{i=1}^{3} \zeta_i(I_H, P_k) \tag{4.8}$$

and:

$$\zeta_i(I_H, P_k) = \begin{cases} 0 & if \;\; h_i(I_H, P_k) \leq 0 \\ C & if \;\; h_i(I_H, P_k) > 0 \end{cases} \tag{4.9}$$

where $C$ is a very large constant aimed at strongly penalized all solutions which are not feasible, i.e., when at least one vertex is not included into Delta's workspace. This term is very useful to immediately discard all the unfeasible solutions.
The second term of the objective function is, as already said, exactly the power of points presented in (4.6):

$$F_2(I_H, P_k) = |h_1(I_H, P_k)| + |h_2(I_H, P_k)| + |h_3(I_H, P_k)| \, . \qquad (4.10)$$

In figure 4.5 it can be seen a level-set plot of the power of points for a generic Delta robot workspace at a given $X - Y$ section, where it is possible to notice the intersection of the three tori. In this image it can be seen the variation of the values of power of points with respect to $(x, y)$ points, differently from what we do in the optimization problem where we try to minimize power of points with respect to the set of parameters $I_H$. It should be noticed that minima reside at the intersections between two different tori, i.e., the intersection between the workspaces of two different Delta's legs.

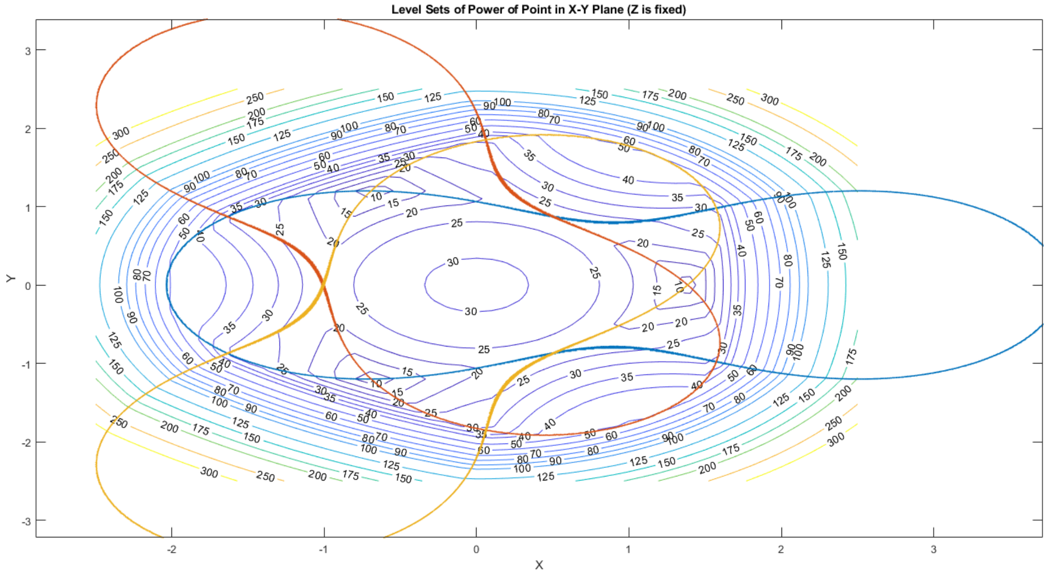


Figure 4.5: Power of points with respect to $(x, y)$ for a given Delta workspace

The previous image validates what we already said about power of points and its meaning about distance from a surface.

## Results

The results presented here comes from a MATLAB script based on the *Optimization Toolbox*, which is a MATLAB library for optimization problems providing a ready to use genetic algorithm function. The mentioned script requires as input the number of variables and the corresponding intervals, as well as an optimization function to compute fitness of generated solutions. This function computes the fitness value for a specific solution $I$ which is then managed by the genetic algorithm script which is responsible for the crossover between the best momentary solutions and the successive random mutation.
The independent variables intervals can be found in the next table:

| | $L$ | $l$ | $\overline{R}$ | $H$ |
|---|---|---|---|---|
| Lower bound | 0.05 | 0.05 | 0.03 | 0.19 |
| Upper bound | 0.2 | 0.3 | 0.05 | 0.29 |

Table 4.2: Optimization variables bounds (in meters)

While the algorithm parameters are the following:

- Population size: 70
- Maximum generations number: 150
- Number of variables: 4
- $L < l$ .

The optimal solution $I^*$ returned by the optimization problem is:

| | $L$ | $l$ | $\overline{R}$ | $H$ |
|---|---|---|---|---|
| Solution | 0.1558 | 0.2430 | 0.4637 | 0.1903 |

Table 4.3: Optimization problem solution (in meters)

Where it should be recalled that $\overline{R} = R - r$ and thus we have the freedom to fix one of the two radii as we prefer,so that the other one is determined consequently. After some geometric evaluation about MAV's available space for Delta mounting, it has been decided to fix $R = 0.0844$ m, while the corresponding movable plate radius being $r = 0.038$ m.
We have finally found the optimal values of the set of geometric parameters $I^*$ for our task; the desired Delta robot is now uniquely identified as:

$$I^* = [\,L^*, l^*, R^*, r^*\,] = [\,0.1558,\ 0.2430,\ 0.0844,\ 0.038\,] \tag{4.11}$$

## 4.2 Components Design

Having obtained the values of the geometric parameters for our Delta robot we are now ready to start designing single components. In doing this, it will be important to satisfy the constraints about weight and mass dispersion presented at the beginning of the chapter.
As a remind, a Delta robot is composed by four main parts, which are:

- 1 mounting (*fixed*) base
- 3 upper links (one for each leg)
- 3 lower links (one for each leg)
- 1 movable plate.

In addition to these, other minor components are always present in a Delta robot; we are referring to joints and other connecting elements. Moreover, motors are important components for a robot too: their selection and sizing will be treated in details in next sections, it is in fact necessary to have a precise measure of system weight before looking for an adequate actuation system.
Ball joints will be the first presented components since the structure of both upper and lower links depend on joint design; after that, the four main Delta components will be shown in details. For most of them, the mechanical structure has been validated by Finite Element Analysis tests which show deformation results under applied loads. All components have been designed on Siemens NX® CAD software.

### Ball Joints

As previously mentioned, a Delta robot is endowed with ball or universal joints for the coupling between upper and lower links as well as between these latters and the movable plate; these particular rotational joints allow for a relative motion in all three axes and are fundametal for the correct behaviour of a Delta robot.
After some joints-suppliers research, it has been decided to adopt IGUS® plastic ball joints. The main advantages of these elements are the reduced weight, only $3.5g$ per each, in particular with respect to standard metal joints, a very low internal friction and a good range of motion. The chosen IGUS® ball joint is shown in the figure 4.6.

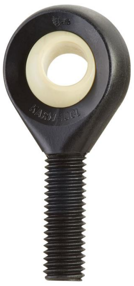


Figure 4.6: IGUS® ball joint

## Fixed Base

The fixed base is the first of the four main components of a Delta robot presented here. The main duty of this element is to secure the robot to the MAV and to provide space for servos mounting. It should be very stiff since it is responsible for holding up the entire Delta robot structure.
After several discarded prototypes, the designed solution is the one shown in the next two figures:

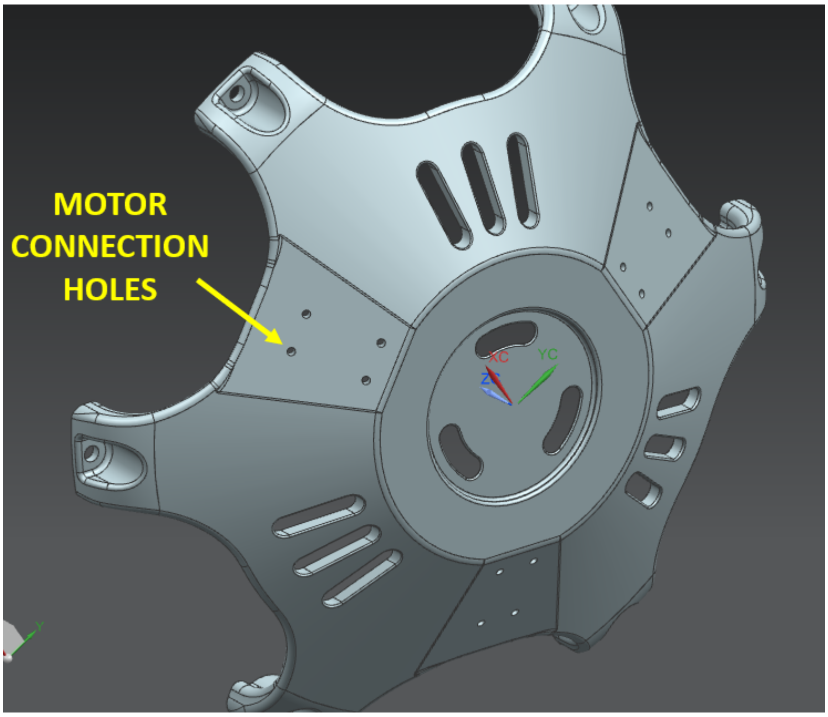


Figure 4.7: Lower view of the mounting base

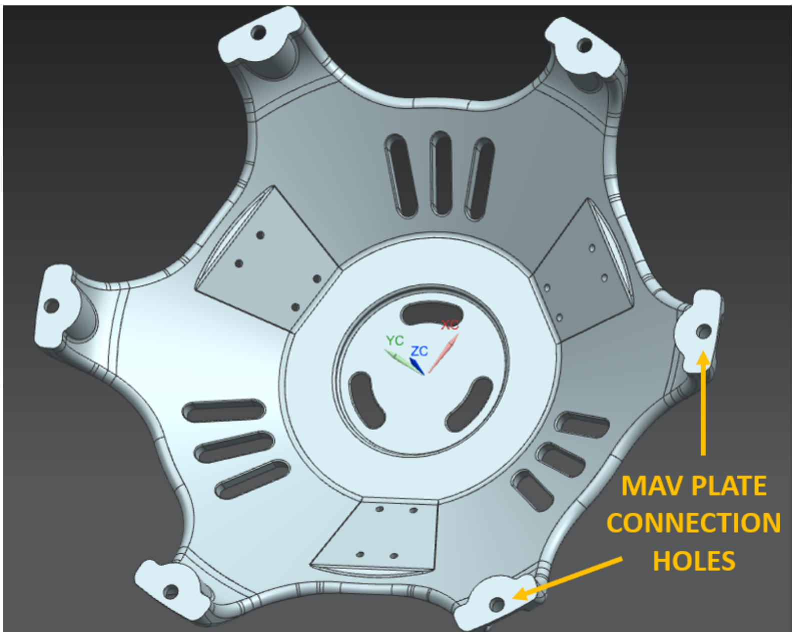


Figure 4.8: Upper view of the mounting base

As can be seen by the two images, the designed base has a curved profile which strengthens its structure against vertical and oblique loads. A robust MAV fixing is ensured by six connection points; three sets of four motor mounting holes are visible and designed specifically for the actuators that will be presented further on.

The fixed base has been manufactured with *3D printing* technology using *nylon*. This material can in fact offer good performance both in term of lightness and stiffness. The total weight of the designed fixed base is then 208 grams.

By taking into account that the mounting base will undergo quite high stress it seemed necessary to exploit finite element analysis for its validation. In this test we decided to simulate the load produced by the weight force of the entire Delta robot acting on the connection points between the actuators and the fixed base, i.e., the three groups of holes that can be seen also in previous images. The value of the force applied at each of these areas is $1.5N$, this value has been obtained by considering a situation of full payload (200 grams) plus the weight of the other components (whose values will be shown in their respective sections). In other words, we are considering here a situation of quite high applied load, even if it is not the most stressed one. It is in fact very likely that some peak forces could easily overcome the one computed here, especially during acceleration phases of the robot. Anyway, it is very difficult to compute an exact value for these peak forces and thus, by considering that the time interval in which they are applied is in general very short and also that other Delta components will be shown to be very light, it seemed to be reasonable to adopt the aforementioned value of $1.5N$. Moreover, the direction of these forces has been chosen parallel to the negative $z$

axis to simulate a standard situation with the MAV horizontal plane parallel to the ground.
FEA deformation results can be seen in figure 4.9 while stress results in figure 4.10.

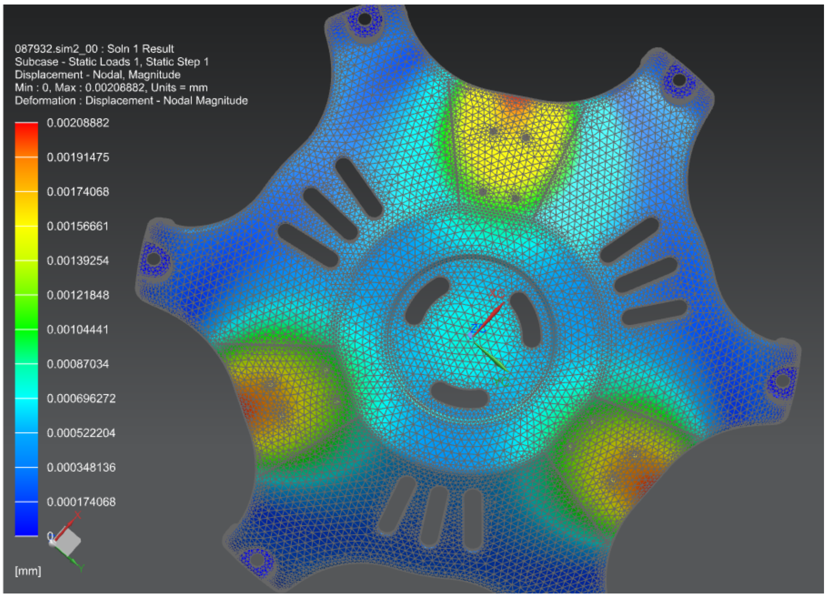

087932.sim2_00 : Soln 1 Result
Subcase - Static Loads 1, Static Step 1
Displacement - Nodal, Magnitude
Min : 0, Max : 0.00208882, Units = mm
Deformation : Displacement - Nodal Magnitude
0.00208882
0.00191475
0.00174068
0.00156661
0.00139254
0.00121848
0.00104441
0.00087034
0.000696272
0.000522204
0.000348136
0.000174068
0
[mm]

Figure 4.9: FEA deformation results (values are in mm)

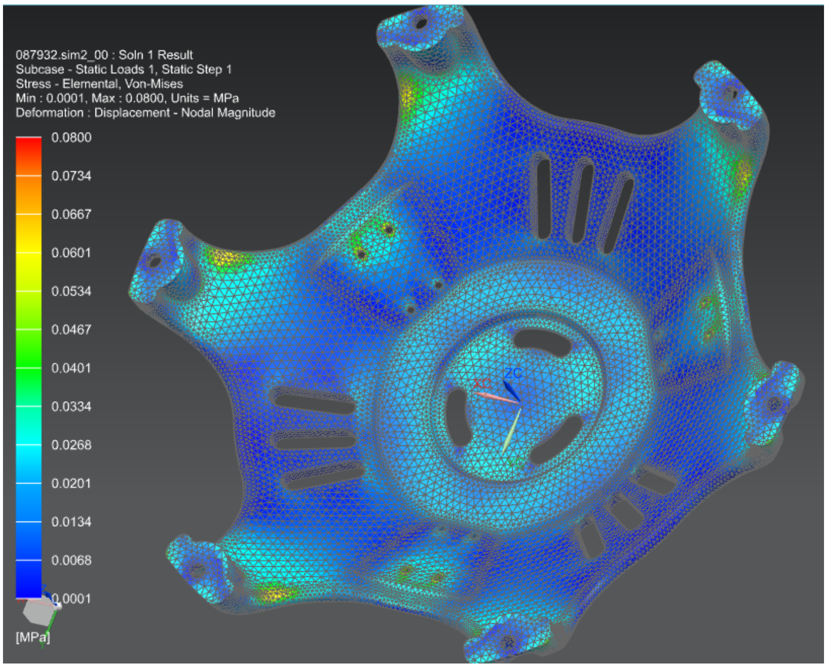

087932.sim2_00 : Soln 1 Result
Subcase - Static Loads 1, Static Step 1
Stress - Elemental, Von-Mises
Min : 0.0001, Max : 0.0800, Units = MPa
Deformation : Displacement - Nodal Magnitude
0.0800
0.0734
0.0667
0.0601
0.0534
0.0467
0.0401
0.0334
0.0268
0.0201
0.0134
0.0068
0.0001
[MPa]

Figure 4.10: FEA stress results (values are in MPa)

From the first image we can see that the maximum deformation is almost null and from the second one we are confirmed that we are very far from the yield strength of Nylon (which is in between 20 and 150 MPa in general).
In conclusion, we are very satisfied with these results which validate the mechanical structure of the designed mounting base.

### Upper Link

The upper link component, like for the fixed base, has been designed in nylon using 3D printing technology. It weights only 29 grams.
The resulting design can be seen in the following figure.

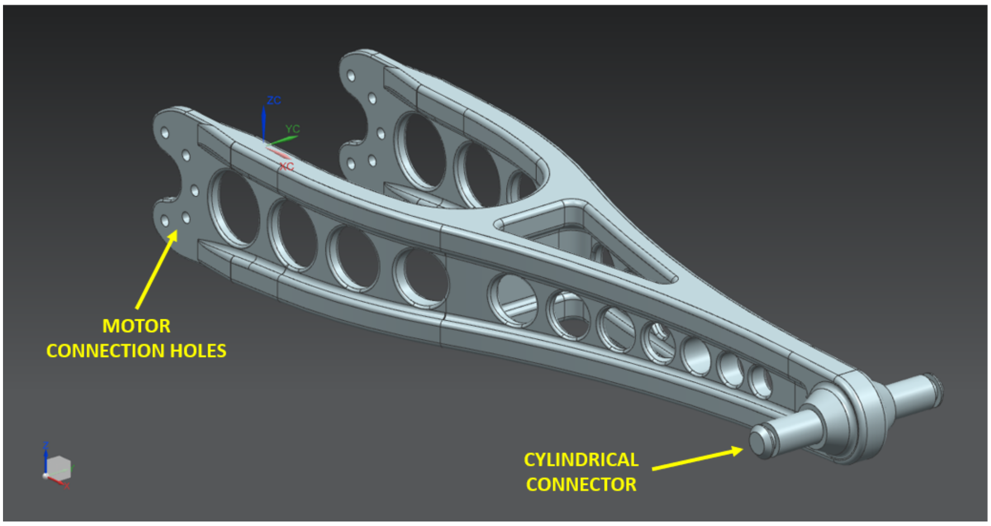
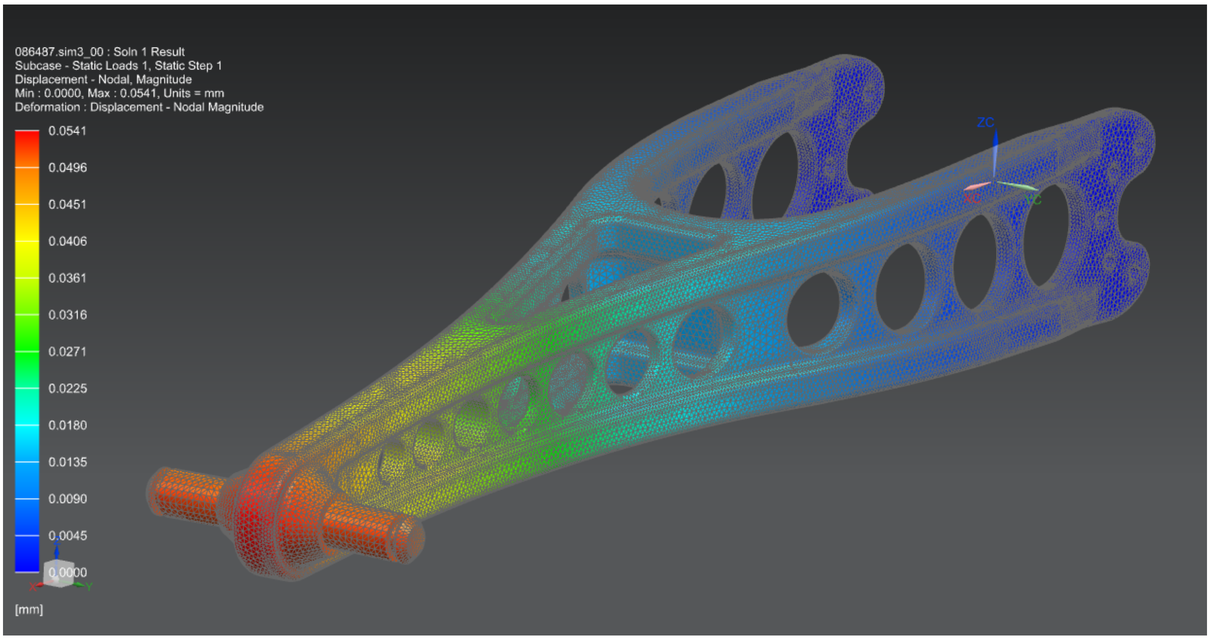


Figure 4.11: Upper link design

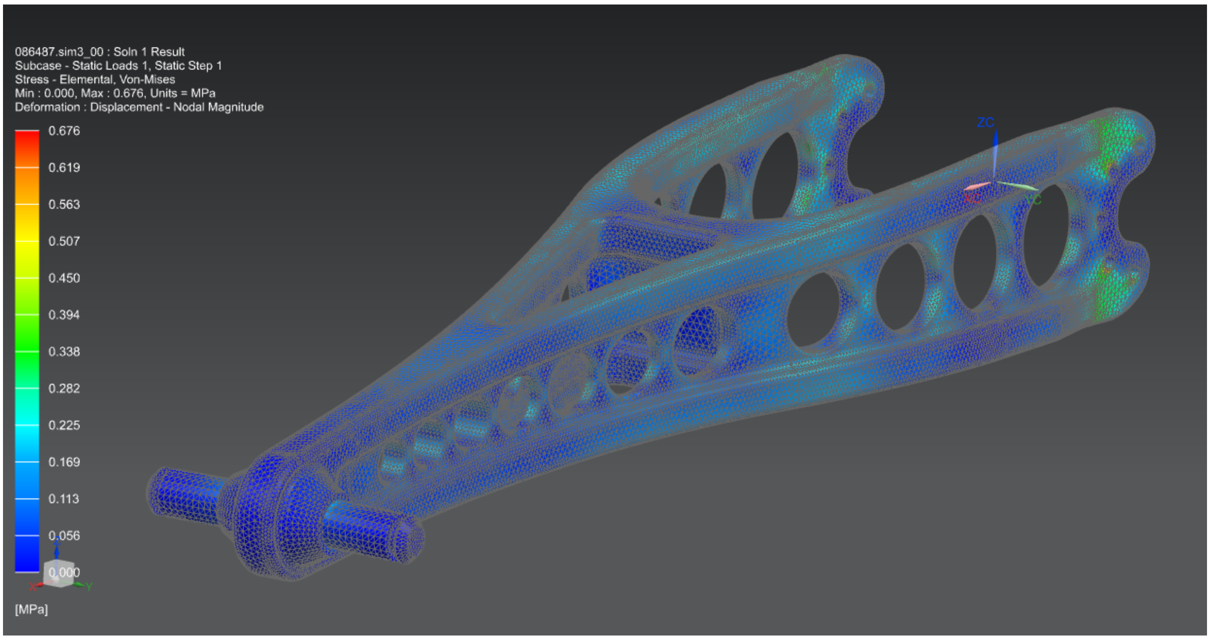
The upper link can be seen as a mechanical *beam* which undergoes both vertical and lateral loads. In order to strengthen its structure against vertical forces we endowed it with a sort of $H$ cross section, while to counteract horizontal bending we decided to fix the upper link to both sides of the motors and thus we added two *legs* to the upper part of this component.
As labelled also in the figure, it should be immediately noticed that the coupling with the ball joints heads is ensured by the integration of a dedicated couple of small cylinders directly on the structure. Thanks to 3D printing technology we do not need any extra connection element since we can overcome the problem of coupling with other components by adding specific shapes to the system. A similar idea has been adopted for the fixing of the upper link to the motor: we added specifically sized holes at both sides in order to guarantee robust coupling with the actuators.
Since also this component will undergo quite high forces, it has been decided to run FEA tests also on the upper link. We simulated standard working conditions

by applying a force with negative $z$ direction and magnitude of $1N$ at both cylindrical connectors. FEA results for deformation can be seen in figure 4.12 and stress results can be found in 4.13.



086487.sim3_00 : Soln 1 Result
Subcase - Static Loads 1, Static Step 1
Displacement - Nodal, Magnitude
Min : 0.0000, Max : 0.0541, Units = mm
Deformation : Displacement - Nodal Magnitude

Figure 4.12: FEA deformation results (values are in mm)



086487.sim3_00 : Soln 1 Result
Subcase - Static Loads 1, Static Step 1
Stress - Elemental, Von-Mises
Min : 0.000, Max : 0.676, Units = MPa
Deformation : Displacement - Nodal Magnitude

Figure 4.13: FEA stress results (values are in MPa)

As for the fixed base, we are satisfied with the results obtained which validate also upper link design.

## Lower Link

As mentioned in the presentation of Delta robot, it has been said that the lower link component is in general made of two parallel thin rods properly coupled with the upper link and the movable plate to form a parallelogram. These connections has been managed by mounting the IGUS® ball joints at both tips of the rods. One of the two rods forming together the lower link is now shown in figure 4.14.

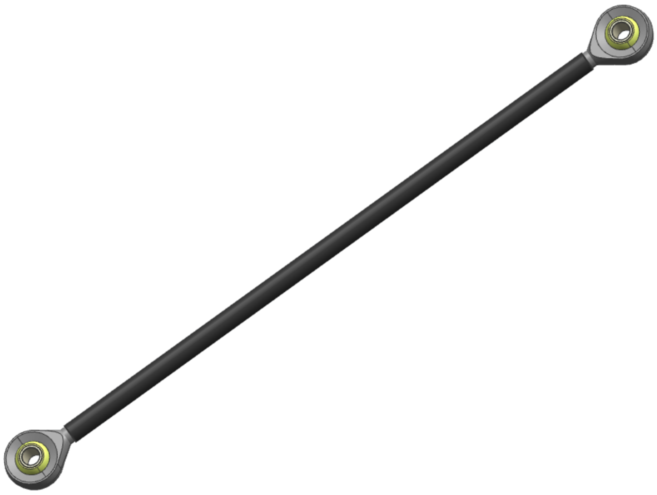


Figure 4.14: A CFRP lower rod coupled with ball joints at both tips

The selected material for these thin rods is Carbon Fiber Reinforced Polymer (CFRP) which has incredible property in terms of both lightness and stiffness. The total weight of a single rod is only $3.5g$, this values arrives to $10.5g$ if we consider also ball joints.

Since in general the lower link of a Delta robot does not undergo to high stress, it has been decided that it was not necessary to provide any kind of structural test for this particular component.

It seemed also useful to provide a dedicated image to highlight the smart coupling between the two lower rods with the upper link by means of the already mentioned integrated cylindrical adapters. This connection can be seen in figure 4.15

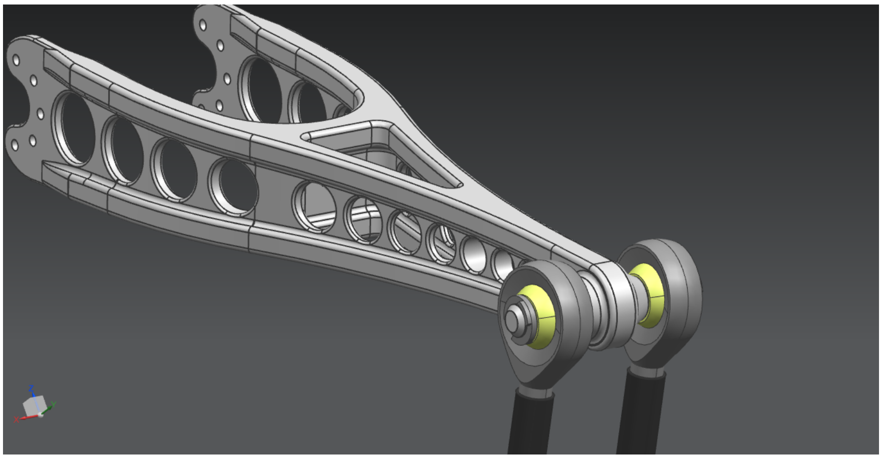


Figure 4.15: Coupling between upper link and the two rods forming lower link

The same concept has been applied to the coupling between the opposite tip of each rod and the movable plate, whose design will be shown in the next section.

## Movable Plate

The movable plate is the end point of the three kinematic chains of a Delta robot and its main goal is to provide space for the mounting of a final element like an end-effector or a sensor to provide the robot with sensing or manipulation capabilities. The presented design has been endowed with the holes for the coupling with the Rokubi EtherCAT force/torque sensor shown in next figure.

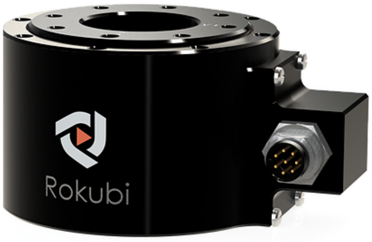


Figure 4.16: Rokubi EtherCAT force/torque sensor from Bota systems

The movable plate design is instead shown in the two different views presented in figures 4.17 and 4.18.

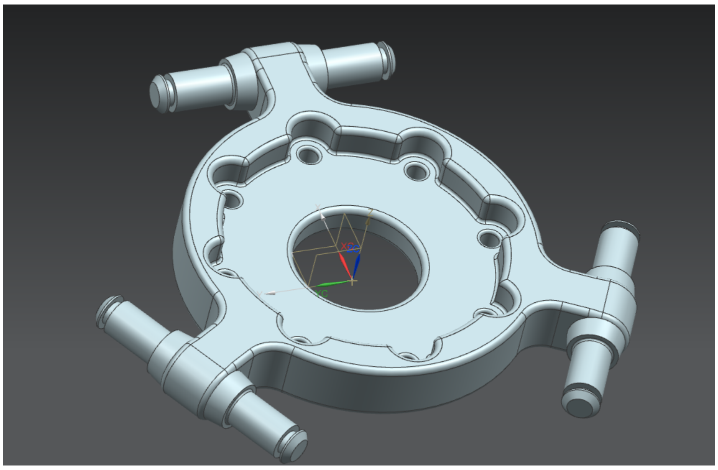


Figure 4.17: Movable plate upper view

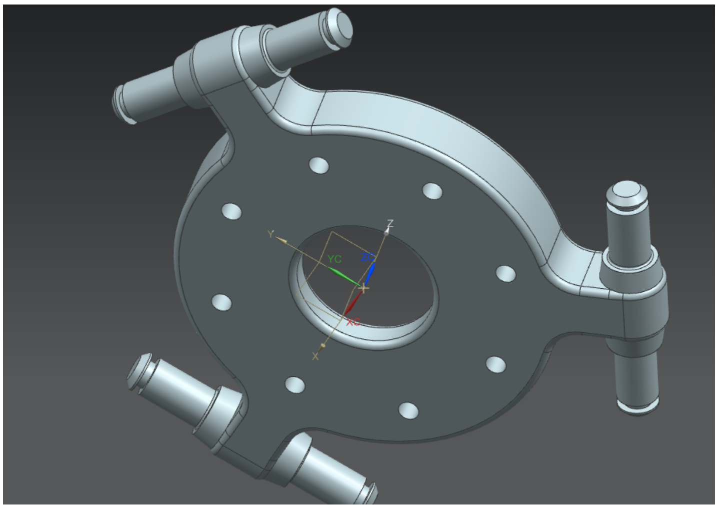


Figure 4.18: Movable plate lower view

As can be seen by the previous images, also the movable plate has been endowed with the cylindrical adapters for the coupling with the ball joints of the three pairs of CFRP rods.
Moreover, also this component has been manufactured in nylon using 3D printing technology and its total weight is 25 grams.
We decided to run FEA test to validate the robustness of the mechanical structure, in particular the resistance against bending of the coupling elements. In this final case we decided to test the component in a very stressed condition, hence we simulated a force of $5N$ with vertical direction applied at the cylindrical connectors. The results of this test, respectively for deformation and stress, are shown in the next two figures 4.19 and 4.20.

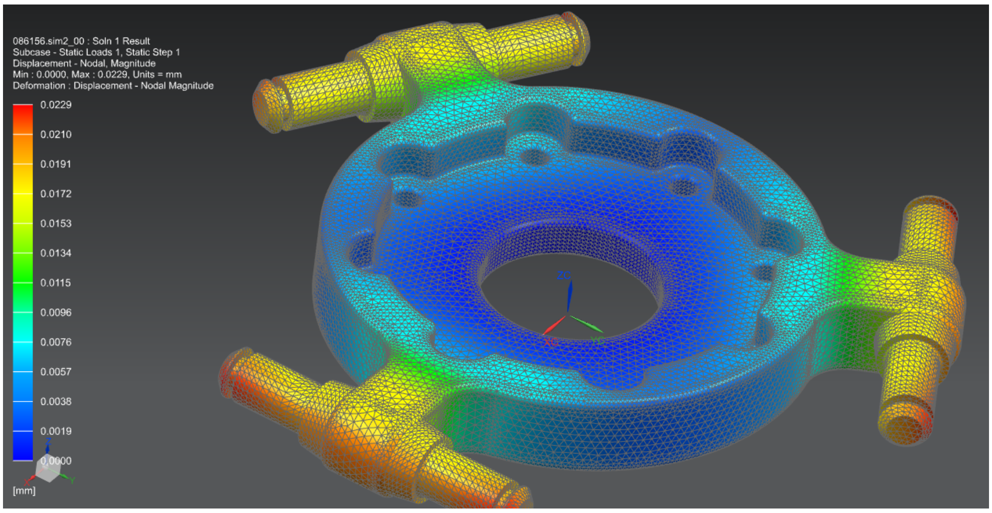

086156.sim2_00 : Soln 1 Result
Subcase - Static Loads 1, Static Step 1
Displacement - Nodal, Magnitude
Min : 0.0000, Max : 0.0229, Units = mm
Deformation : Displacement - Nodal Magnitude
0.0229
0.0210
0.0191
0.0172
0.0153
0.0134
0.0115
0.0096
0.0076
0.0057
0.0038
0.0019
0.0000
[mm]
Z
X
Y

Figure 4.19: FEA deformation results for an applied load of $5N$ at cylindrical connectors (values are in mm)

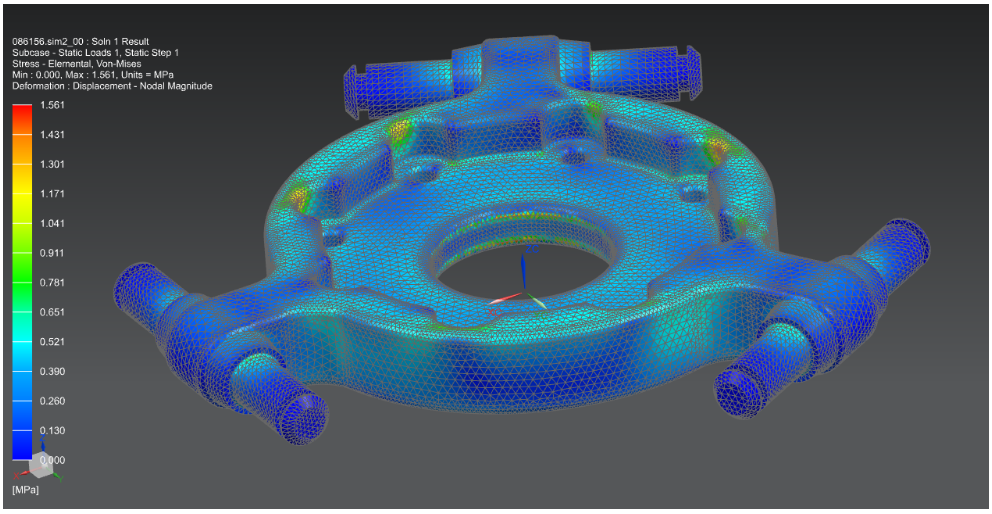

086156.sim2_00 : Soln 1 Result
Subcase - Static Loads 1, Static Step 1
Stress - Elemental, Von-Mises
Min : 0.000, Max : 1.561, Units = MPa
Deformation : Displacement - Nodal Magnitude
1.561
1.431
1.301
1.171
1.041
0.911
0.781
0.651
0.521
0.390
0.260
0.130
0.000
[MPa]
Z
X
Y

Figure 4.20: FEA stress results for an applied load of $5N$ at cylindrical connectors (values are in MPa)

As for the previous components, we are very satisfied with the FEA results for the movable plate; despite the high force applied, the deformation results are still almost null, the stress analysis highlights higher values with respect to previous tests but it should be reminded that we applied much higher force than before, we can in fact conclude that the obtained values are completely acceptable; nonetheless, we are still very far from nylon yield strength and we can thus consider these results as very positive.

## Motor Selection and Sizing

As mentioned at the beginning of the chapter, motor selection and sizing is a process which requires a complete knowledge about system geometric and inertial parameters. Having obtained all single components it is now possible to look for an actuation system able to guarantee desired performance.
For the sizing of the motors it has been decided to take into account three different motion configurations. First of all we computed the maximal requested torque in a static scenario (i.e., the torque necessary to counteract gravity action): the computed value is then 0.656 $N \cdot m$ and it corresponds to a configuration in which the orientation of MAV's horizontal plane is almost vertical.
Then we computed the *peak torque*, which can be seen as the maximum instantaneous torque that can be required to one or more of the three Delta motors. To compute a precise value, we imposed to the movable plate an acceleration of $4\,m/s^2$, which has been chosen as maximum desirable acceleration; the resulting peak torque is then $0.905\,N \cdot m$.
Lastly, we decided to impose a condition on velocity too: by selecting as maximum movable plate velocity $0.5\,m/s$ and deriving the corresponding velocity value at motor side, we obtained a maximum required angular velocity of $48rpm$.

By taking into account all these constraints, and after comparison of several different actuation devices, it has been decided to choose the Dynamixel XM430-W210 servo motor from Robotis. This device can in fact satisfy all motion requirements and offers excellent performance both in term of accuracy (0.088° as resolution) and torque/velocity profile. It is able to reach a maximum torque of 2 $N \cdot m$ and a no-load speed of $72\,rpm$, while the weight is only 82 grams per each servo. The XM430-W210 servo motor and iis shown in figures 4.21.

Figure 4.21: XM430-W210 servo motor (image from Robotis website)

### Complete Design and Manufactured Prototype

Having presented all single Delta components in details, we are now ready to show the CAD assembly of the complete robot, which can be seen in figure 4.22.

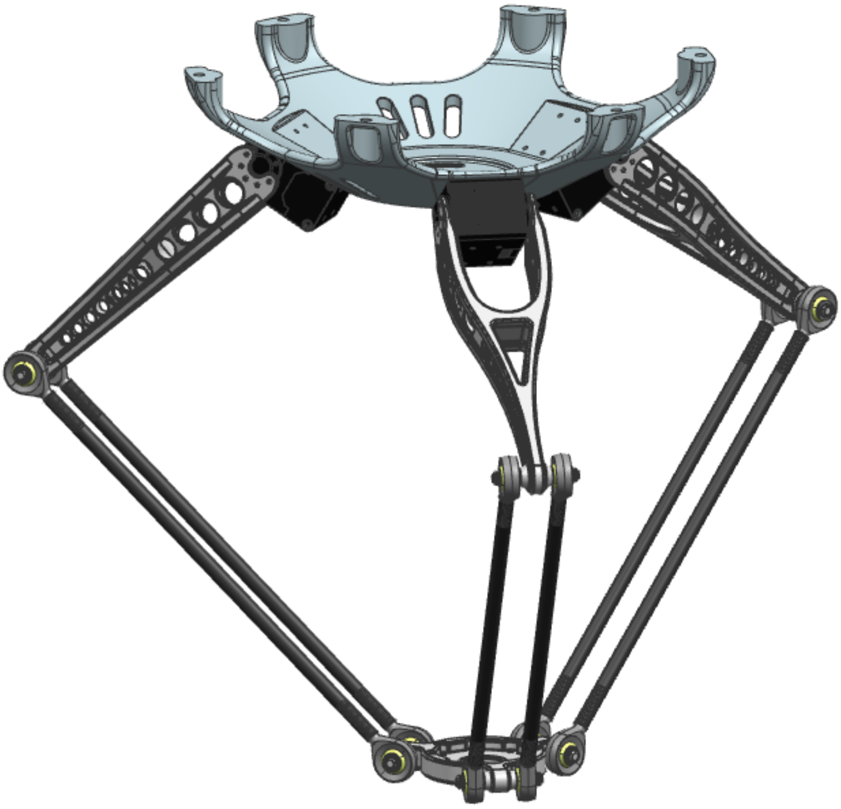


Figure 4.22: Overall Delta robot design

The couplings between the mounting base and the servos, as well as the ones between servos and upper links, have been secured by specific screws; while the fixing between upper links, lower links and movable plate have all been made by the already mentioned integrated cylindrical connectors. One extra circlip has been adopted for each coupling to prevent the ball joint from sliding away and thus to ensure a close fixing.

In conclusion, we are very satisfied with the resulting robot, in particular for its intrinsic simplicity thanks to the flexibility provided by 3D printing technology. Moreover, also mass minimization has been achieved with success: the entire robot, considering also servos, weights only 630 grams. But if we do not consider servos contribution, the weight gets reduced to 380 grams. This last result confirms also that we succeeded in minimizing both mass dispersion and moving mass; it is in fact true that around 70% of the mass (servos plus mounting base) is not moving and concentrated at the center of the robot.

The corresponding manufactured prototype is instead presented in figure 4.23: it can be seen that the robot is upside down with respect to previous image, this temporary solution was adopted for easier fixing. Moreover, also the mounting base is not the same of the CAD model, we had to adopt another wood plate for correct fixing to that temporary support while the originale mounting base was instead specifically designed to be used with the MAV.

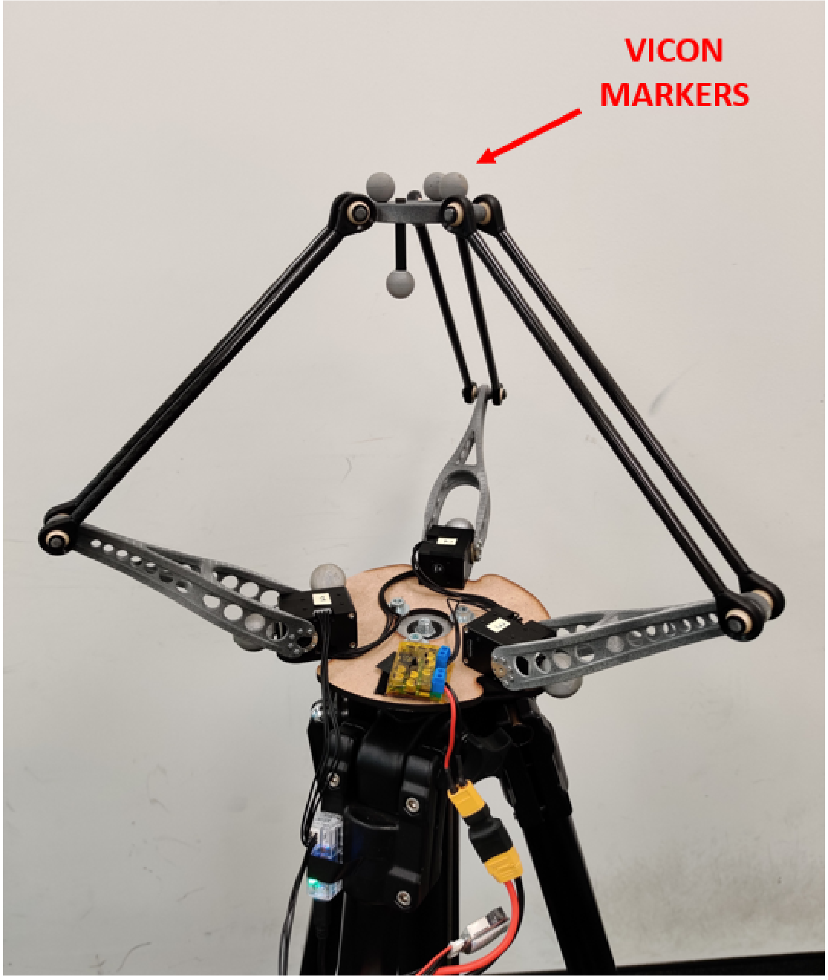


Figure 4.23: Resulting Delta robot prototype

From this image it is possible to see also VICON tracking markers which will be used further on to run final performance tests. A part from these and the mounting base, the presented prototype is equal to the one previously shown in the CAD assembly.

# Chapter 5

# Control

The present chapter will describe all the work related to the control of the manufactured Delta robot. Firstly, theory and concepts behind the implemented control technique will be presented; secondly, it will be given an overview of the Dynamixel workbench ROS package provided by Robotis and its corresponing APIs; lastly, we will focus on the developed ROS structure and code for controlling our system.

As mentioned, all the control work has been based on the very famous *Robot Operating System* (ROS). As described in the official web page, ROS is a set of software libraries and tools to help software developers create robot applications. It provides hardware abstraction, device drivers, libraries, visualizers, message-passing, package management, and more. It is one of the most popular and used software tool in the robotics community.

## 5.1 The Inverse Kinematics-Based Control

As highlighted in the introduction of this thesis, one of the key requirements for the developed robot is the ability to compensate for MAV's position errors and thus maintain the end-effector position to millimetric accuracy.
To successfully achieve this goal, it has been decided to implement an inverse kinematics-base control inspired to the one proposed by Chermprayong et al. in [8] for their Delta manipulator aerial system.
The main idea behind this approach consists in computing the target joint angles corresponding to a Delta end-effector goal position by exploiting the knowledge of MAV's pose (and thus Delta's pose). The information about MAV pose could come either from an external odometry system, like a VICON room, or from an internal measurement system, like IMU. The first solution is feasible only for in-door applications while the second one could be adopted also for out-door scenarios.

As already said, the presented approach follows the reasoning made by Chermprayong et al. in their paper and it is schematized in figure 5.1.

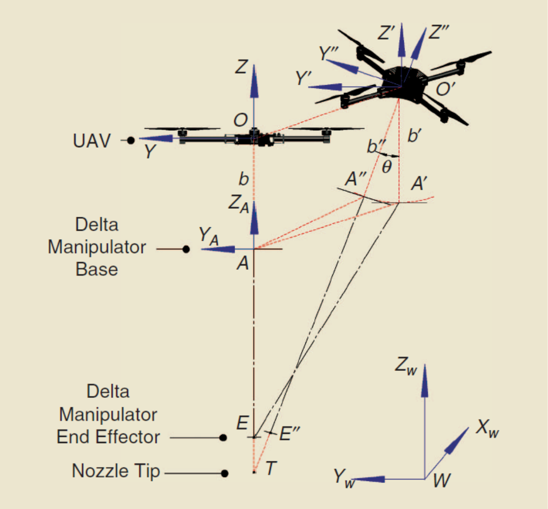


Figure 5.1: Inverse kinematics-based control schematic concept (image from [8])

Three Cartesian frames are used for the control of the system: the first one is the world frame $W$, which is fixed and it is the one generally adopted by VICON system to express position and orientation in space of both MAV and Delta robot, as well as for target points too. The two other reference frames are the ones attached to MAV and Delta base and are defined, respectively, as $M$ and $D$. They are used to transform Delta end-effector goal points into local coordinates, that are the suitable ones for computing corresponding joint angles.
Regarding frames relative position, it should be underlined that MAV frame is fixed at the center of the MAV and it is parallel but just shifted along $Z$ axis with respect to Delta frame, whose $X - Y$ plane lays on the plane identified by the centers of the servos. The pose of the MAV (and thus of Delta too) is constantly computed by an odometry system (VICON or IMU) and stored for future calculations.

From a practical point of view, the implemented control system works in the following way: a desired goal point $\{\mathbf{E}\}$ for the end-effector is specified to the control unit, this value could be expressed both in world frame $W$ or in Delta base frame $B$, for now we will take into account the case in which goal position is specified in world coordinates. Then, by the knowledge of MAV position in world frame $\{\mathbf{O}_M - \mathbf{O}_W\}$ coming from VICON/IMU and by exploiting basic geometric considerations shown also in figure 5.2, the control system computes the end-effector goal position with respect to MAV origin, which is vector $\{\mathbf{E} - \mathbf{O}_M\}_W$ in the cited image. In notation:

$$\{\mathbf{E} - \mathbf{O}_M\}_W = \{\mathbf{E} - \mathbf{O}_W\}_W - \{\mathbf{O}_M - \mathbf{O}_W\}_W \; . \tag{5.1}$$

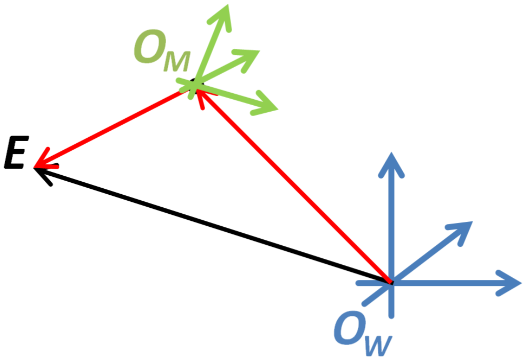


Figure 5.2: World frame, MAV frame and corresponding vectors with respect to end-effector goal **E**

It is thus the moment to exploit odometry data to transform the coordinates of this last vector into MAV/Delta local frame: this can be easily done by exploiting quaternions geometry. It is in fact typical in robotics applications that the relative orientation of a body is expressed using quaternion, since these tools do not suffer from computational issues like standard rotation matrices.
Thus, by defining as $\mathbf{q}_M$ the quaternion whose vector part is $\{\mathbf{E}-\mathbf{O}_M\}_M$ and with null scalar part, and $\mathbf{q}_W$ the corresponding quaternion derived from $\{\mathbf{E}-\mathbf{O}_M\}_W$, i.e.:

$$\begin{aligned}\mathbf{q}_M &= (0\,, \{\mathbf{E} - \mathbf{O}_M\}_M)\\ \mathbf{q}_W &= (0\,, \{\mathbf{E} - \mathbf{O}_M\}_W)\end{aligned}$$

and by defining as $\overline{\mathbf{q}}$ the rotation quaternion coming from odometry, we can obtain $\mathbf{q}_M$ by standard quaternion rotations, i.e.:

$$\mathbf{q}_M = \overline{\mathbf{q}} \;\; \mathbf{q}_W \; \overline{\mathbf{q}}^* \tag{5.2}$$

where $\overline{\mathbf{q}}^*$ is the conjugate of $\overline{\mathbf{q}}$.

We have thus obtained the coordinates of the end-effector goal in local frame, which are exactly the vectorial part of $\mathbf{q}_M$. If the computed vector is in MAV frame, it can be easily transformed into Delta frame by a simple translation along $Z$ equal to the distance between the centers of the two frames.
At this point, we just need to calculate the corresponding joint angles for the given end-effector goal position: this can be done by exploiting the already presented inverse kinematics routine, suitably adjusted to run inside the control unit.

We then find the target joint angles $\theta^* = [\theta_1, \theta_2, \theta_3]$ which have to be sent as a command to the corresponding servo motors.

In conclusion, it has been presented a method based on [8] to compute in real time joints goal angles for a given end-effector goal position in world frame, which in general could vary during time. MAV's pose is provided by a dedicated odometry system based either on VICON or IMU and exploited to compute both end-effector position with respect to MAV origin and to transform this vector into MAV local coordinates by using quaternions algebra. As a remark, it should be reminded that this routine could work as long as the end-effector goal position is *inside* Delta workspace, it is thus now even more evident the importance of designing a sufficiently sized robot.
A validation test will be presented further on to show system capabilities in maintaining the end-effector position fixed even in presence of external disturbances and MAV position errrors.

## 5.2 ROS Framework

As mentioned in the introduction, all the control framework has been developed using ROS. Robotis, the supplier of Dynamixel servos, has also published a ROS package called *Dynamixel Workbench* which is a software library providing some functionalities to start working with Dynamixel servos. In particular, Dynamixel workbench gives a C++ ROS node which can communicate to one or more servos by writing on their control table items using Dynamixel SDK API, which are the low level libraries providing very basic functionalities to allow the user interface with their servos.
Dynamixel servos can in fact be controlled by writing to and reading from values stored in the control table that each servo is endowed with; by manipulating these values several different functions can be performed providing a very extensive control over Dynamixel servos. In particular, the XM430-W210 model can be controlled in several different ways, for example: we decided to use "Position Control Mode" in which target angles are specified to control the servos, but it is possible also to adopt other typologies of control, like "Velocity Control Mode" or also "Current Control Mode". Moreover, also servos PID parameters can be suitably tuned by playing with specific control table items values. Much more items are present in the control table of the XM430-W210 servo motor: for a complete overview of these ones, the reader is remanded to the Robotis e-manual of this specific model.

As already said, Dynamixel workbench can be seen as a sort of interface between the servos hardware and the higher-level control. Originally it provided some very basic functionalities which have been increased and adapted to our specific

needs.
On the top of the modified Dyanmixel workbench package, a dedicated ROS package for Delta control has been developed. This control unit can provide several functions to control our Delta robot and it has been designed to stay in between Dynamixel workbench and the user. Moreover, it is also responsible for processing incoming data from the odometry system and it computes servos target angles as shown in previous section. A schematic representation of ROS organization for Delta aerial manipulator control can be see in figure 5.3.

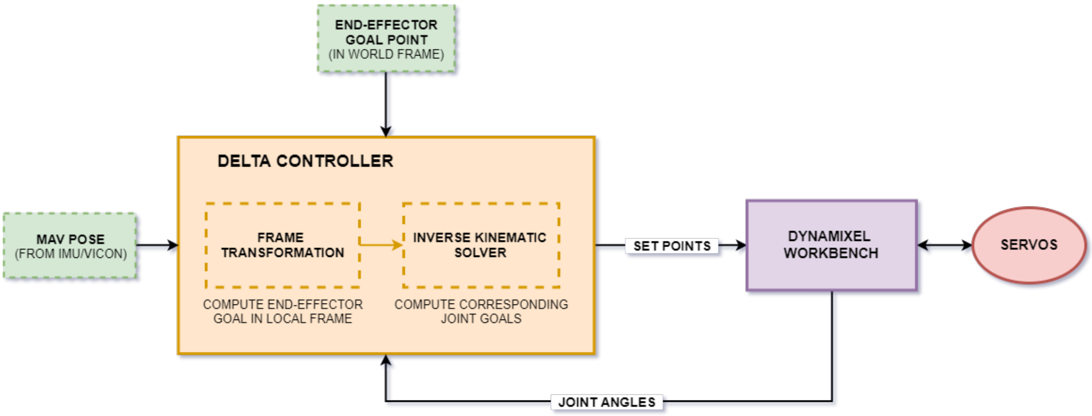


Figure 5.3: ROS organization for Delta aerial manipulator control

As can be seen from figure 5.3, end-effector goal position is sent to the Delta controller ROS node which firstly transforms it into local frame by exploiting odometry data published in a specific ROS topic by VICON or IMU; then, a dedicated inverse kinematics routine is called to computed corresponding joint angles which are immediately sent as a ROS message to a topic subscribed by Dynamixel workbench node. Once joints goals arrived to Dynamixel workbench, they get immediately written into corresponding control table items of the Dynamixel servos. Suitably tuned PID internal parameters are then used to compute torque at the servos and to provide the system with a good responsive behaviour. As shown also in the figure, real time joint angles are continuously sent by Dynamixel workbench to the Delta controller to keep track of the current state of the joints.

In figure 5.4 it is presented a graph of the implemented ROS communication. Besides the inverse kinematic-based approach for end-effector position maintenance, some extra functionalities are provided by Delta controller. It is in fact possible to perform also smooth trajectories both in jointspace and in workspace by a dedicated topic provided by Delta controller: this solution, which is based on a trajectory generator unit inside Dynamixel controller, is thought for tasks in which smoothness is more important than speed and could be used in future ap-

plications of the Delta robot when used both as a standalone system or mounted on the MAV. Specific ROS services are also available to compute inverse and forward kinematics either for current Delta configuration or for a desired input one.
It should also be mentioned that all these ROS topics have been set to run at 200 Hz frequency to provide high reactivity and fast errors compensation.

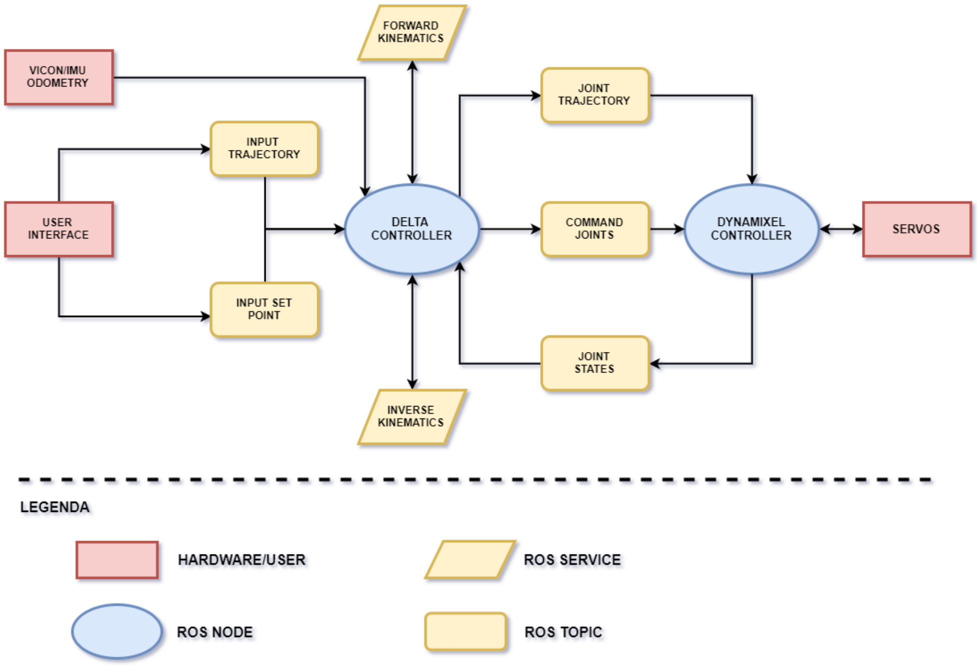


Figure 5.4: ROS communication scheme

As a final remark, it should be recalled that at the beginning of this text we mentioned two different control approaches that could have been suitable for our needs: the *independent* approach, in which MAV and manipulator are controlled by two separate control units, and the *overall* approach, in which an unique control unit handles the entire system. As just shown, it should be clear that we decided to go for the independent solution; this has been done for a couple of reasons: first of all, remaining project time did not seem to be enough to try implementing an overall approach which is undoubtedly more complicated; secondly, because the adopted solution seemed to provide quite good results too and the difference in control complexity seemed to be too high with respect to the difference in achievable results. Anyway, it would be surely interesting to try implementing the overall approach too: this would allow also to directly measure the differences in terms of performances provided by the two approaches; this work is therefore left to future developments.

# Chapter 6

# Final Tests and Results

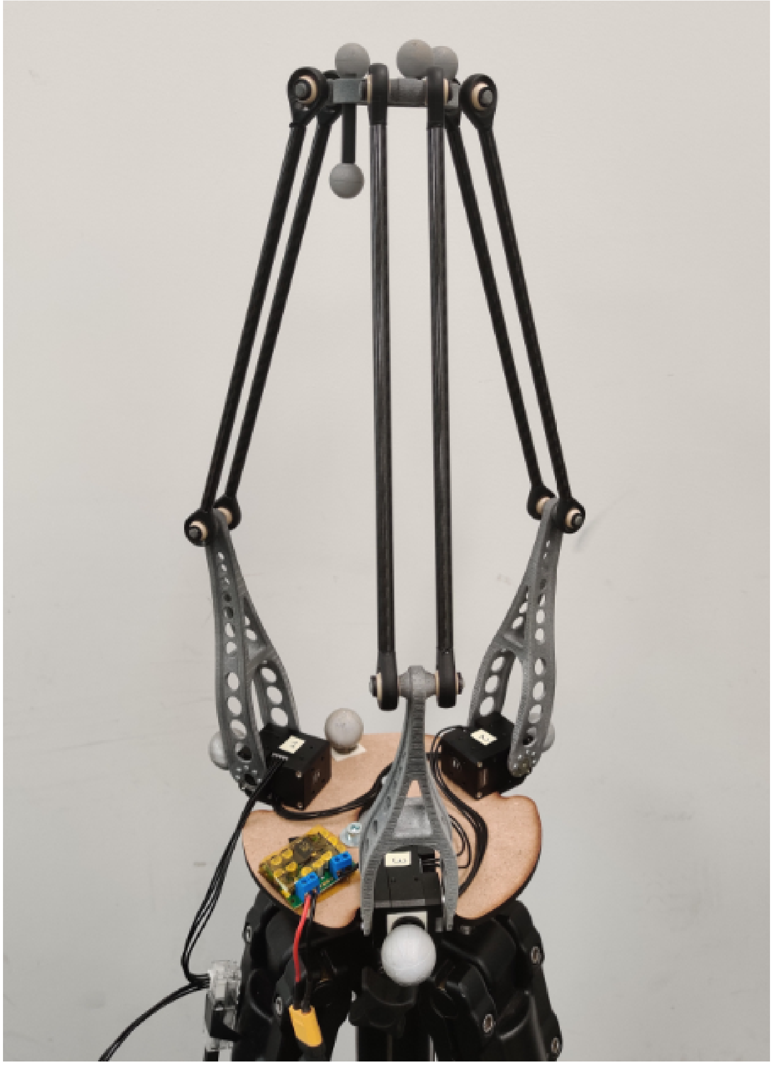


Figure 6.1: Our Delta robot in an extended configuration

This chapter will present the results provided by different field tests that have been run and then discuss them to identify positive and negative aspects of the designed system.
The first part of the chapter will individually describe the features of the different tests and explain goals of each one. Plots and results will then be presented and discussed. Lastly, final conclusions about system performances will be proposed to highlight achieved goals and possible future work for further improvements.

## 6.1 Sinusoidal Trajectory Tracking Tests

The first two tests that have been run are both based on a three-dimensional *sinusoidal* trajectory in workspace with the following characteristics:

- $X$ direction:
  - Amplitude: 5 cm
  - Frequency: 1 Hz
  - Offset: 0 cm
- $Y$ direction:
  - Amplitude: 5 cm
  - Frequency: 1 Hz
  - Offset: 0 cm
- $Z$ direction:
  - Amplitude: 8 cm
  - Frequency: 2 Hz
  - Offset: -24 cm

where all the offsets are expressed with respect to Delta base frame origin.

### End-Effector Tracking by Forward Kinematics

In this first test the objective is to evaluate the ability of the system at following the just introduced trajectory and then to measure corresponding errors.
End-effector position is computed by exploiting the forward kinematics service provided by Delta controller, which is based on the real time servos angles coming from Dynamixel workbench.

Results are now shown in the next three plots which are referred to, respectively, $X$, $Y$ and $Z$ end-effector coordinates, error trajectory is also reported in the lower plots.

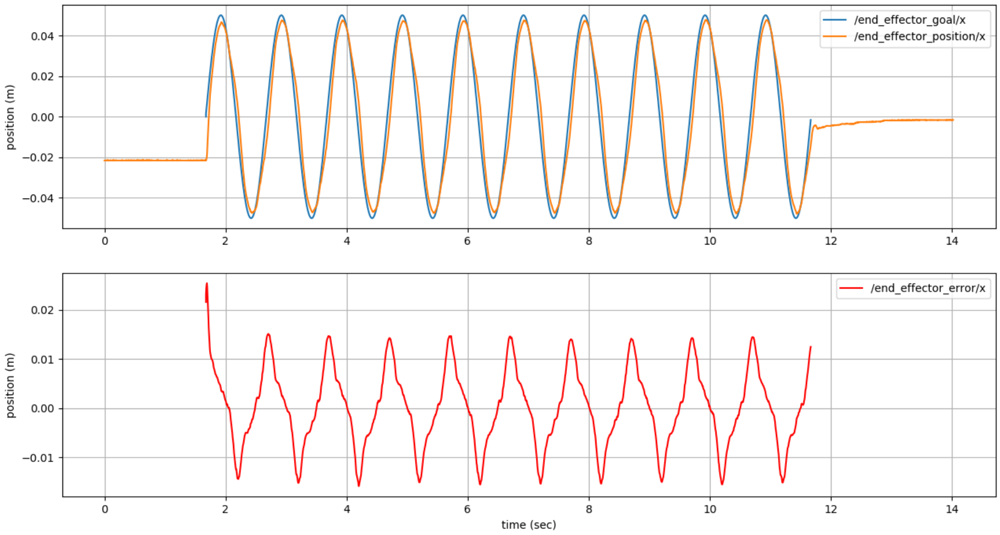


Figure 6.2: End-effector tracking and corresponding error in $X$-coordinate

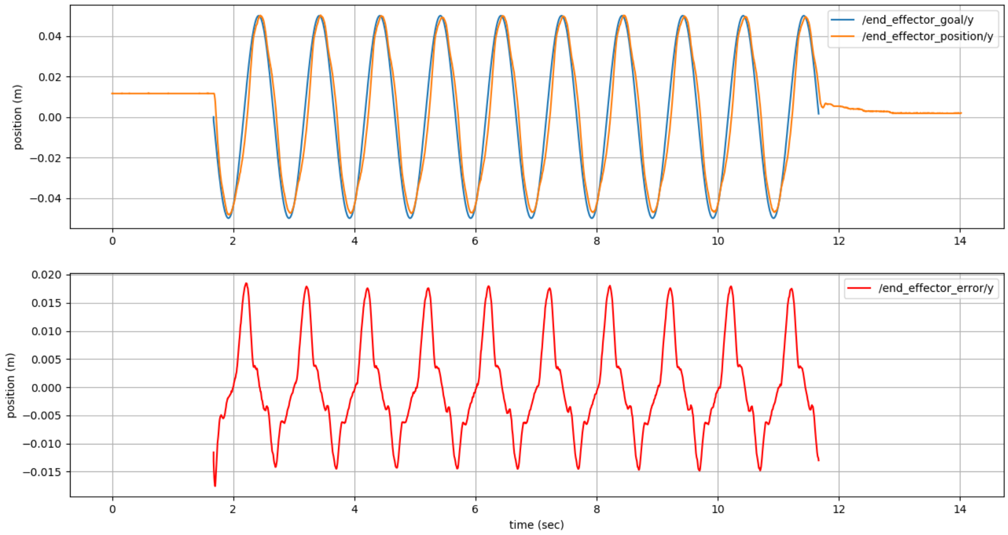


Figure 6.3: End-effector tracking and corresponding error in $Y$-coordinate

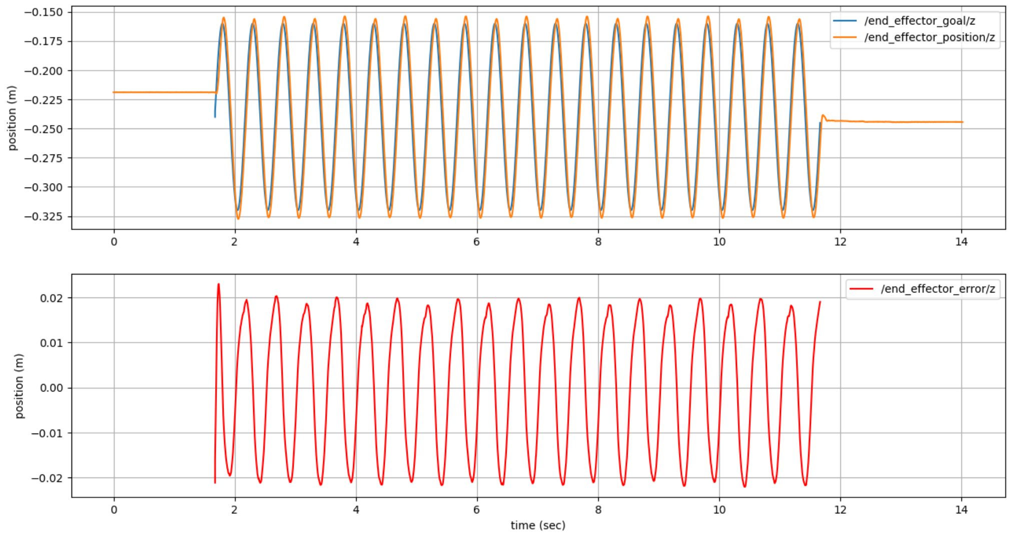


Figure 6.4: End-effector tracking and corresponding error in $Z$-coordinate

Some errors statistics have also been computed in order to get more useful values for the just seen trajectories, in particular:

| Error | $X$ (m) | $Y$ (m) | $Z$ (m) |
|---|---|---|---|
| Mean Value | 1.61$e$-04 | 7.29$e$-05 | 1.49$e$-04 |
| RMSE | 0.008081 | 0.008716 | 0.014721 |
| Standard Deviation | 0.008079 | 0.008716 | 0.014720 |

Table 6.1: Error statistics for trajectory tracking test

These results confirm that the system is able to follow the trajectory quite well, as can be seen also from the three plots. $Z$-direction data are subject of higher errors with respect to trajectories along $X$ and $Y$ but this comes from the fact that the sinusoidal trajectory along $Z$ has doubled frequency and almost doubled amplitude, so this errors difference was clearly foreseeable. It should also be considered that the tested trajectory is quite challenging for our system since it covers quite a large area of the robot workspace with satisfying velocity: this confirms us even more the goodness of these results.

## Vicon Tracking Comparison

The second test, as already said, is still based on the same sinusoidal trajectory seen before. Anyway, this time we are interested in evaluating the difference in end-effector position tracking between the values computed by the already seen forward kinematics service of Delta controller and the ones provided by VICON estimation. As can be seen from figure 6.1 at the beginning of this chapter, four VICON markers have been placed both in the base plate of the robot as well as in the movable plate. These markers are properly designed to be constantly tracked by VICON cameras and, after preliminary calibration, they allow us to get real time value of the end-effector position, i.e., the center of the movable plate in our test.
The main idea behind this test is to *validate* the end-effector position computed internally by Delta controller and thus to get a precise measure of the mismatch between the two results. The reasons for these differences will be discussed further on because deserved to be properly described.

As before, in the next three plots it is shown the end-effector position in $X$, $Y$ and $Z$ coordinates; in light blue it is represented the trajectory comuputed by forward kinematics service from Delta controller and in orange the one from VICON system. Differences are highlighted in the lower plots in red.

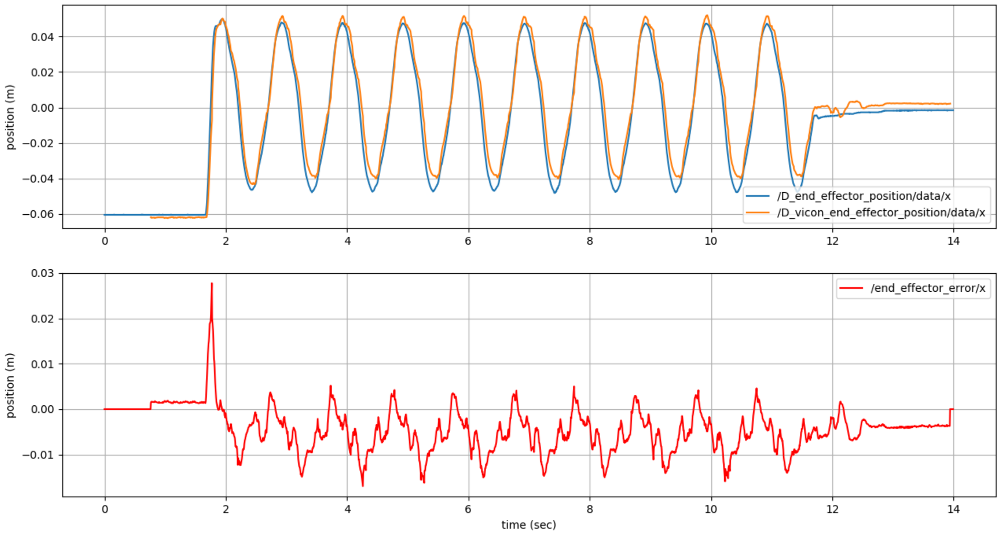


Figure 6.5: End-effector position by FK and VICON in $X$-coordinate

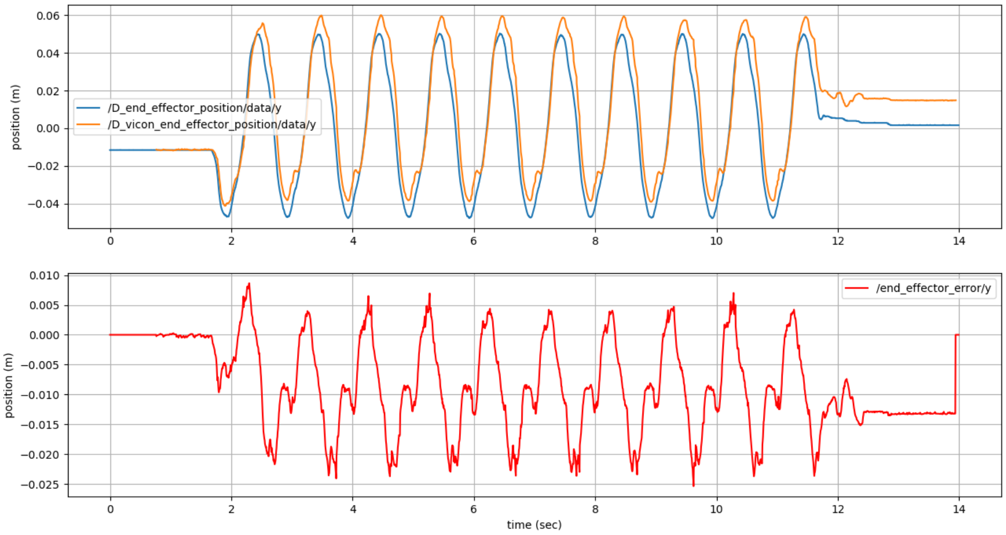


Figure 6.6: End-effector position by FK and VICON in $Y$-coordinate

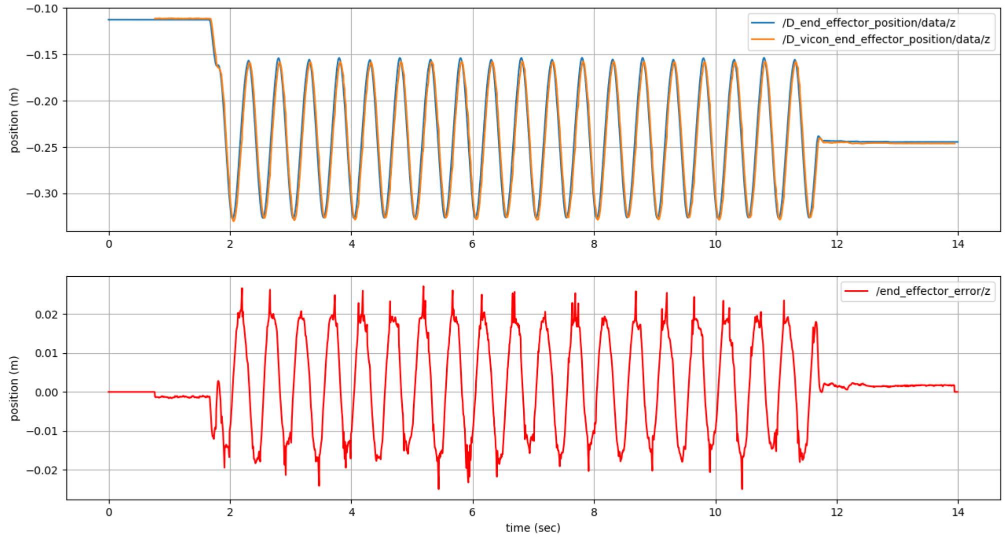


Figure 6.7: End-effector position by FK and VICON in $Z$-coordinate

The same error statistics as before have also been computed and reported in the table 6.2.
By looking at both the above results and the table, it is evident that there are

| Error | $X$ (m) | $Y$ (m) | $Z$ (m) |
|---|---|---|---|
| Mean Value | $-0.004332$ | $-0.008269$ | 0.001647 |
| RMSE | 0.006437 | 0.011279 | 0.011621 |
| Standard Deviation | 0.004761 | 0.007670 | 0.011503 |

Table 6.2: Error statistics for VICON validation test

some differences between the position computed by the forward kinematics service and by VICON. As previously mentioned, the sources of these errors deserve to be properly investigated and explained.

First of all, a possible reason for these errors is very likely to come from a non perfect calibration of the VICON markers, this is also suggested by the presence of a residual mean error, in particular along $X$ and $Y$ axes. By taking into account this, other sources of errors can be reasonably identified in:

- non perfect machining of components (lower rods have been cut by hand)
- low precise coupling between CFRP rods and threaded part of ball joints
- non optimized homing offset for the servo motors.

As can be easily imagined, these three errors sources could be enough to justify most of the resulting differences. Moreover, all these problems could be solved by some small but precise improvements which will be explained in more detail at the conclusion of this work. To conclude also this second test, we can say that we are satisfied with the obtained results since we strongly believe that most of the errors come from the cited problems and thus could be easily canceled out.

## 6.2 MAV Compensation Routine

The third and final test has the goal of evaluating the capabilities of the system at maintaining end-effector position fixed while it is mounted on the MAV and thus subjected to drifts caused by external disturbances like wind or MAV position errors. As already said, in this test we are interested only in the compensation abilities of the system and not in tracking performances; for this reason it has been decided to provide the system with a world frame end-effector goal position constant in time. The routine for controlling our Delta robot is exactly the one presented in chapter 5 and the MAV pose tracking data is directly provided by VICON system. Before proceeding with plots and results, it is important to recall that developed control routine is able to maintain the end-effector position at the target point as long as this is inside the workspace; obtained results are thus directly dependent on MAV errors, which is something unpredictable for us. Moreover, it seems reasonable to assert that further system improvements should be focused at minimizing MAV pose errors since this would directly benefit also Delta performances; following this reasoning, it seems now even more interesting to implement in future some kind of *overall* approach for system control.
Test results are now shown for each specific coordinate in the next three plots:

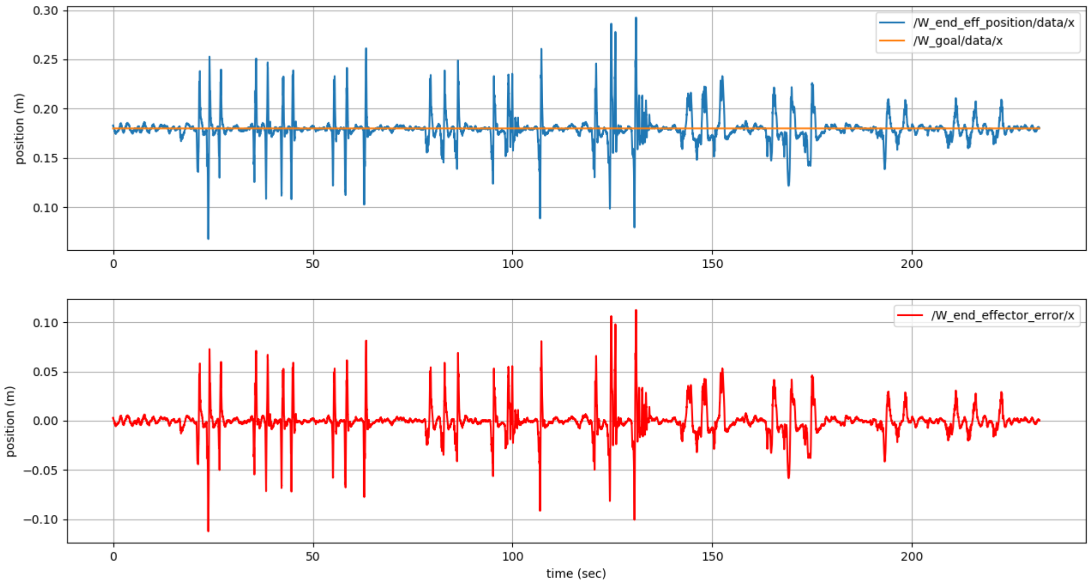


Figure 6.8: MAV's errors compensation routine results in $X$-coordinate

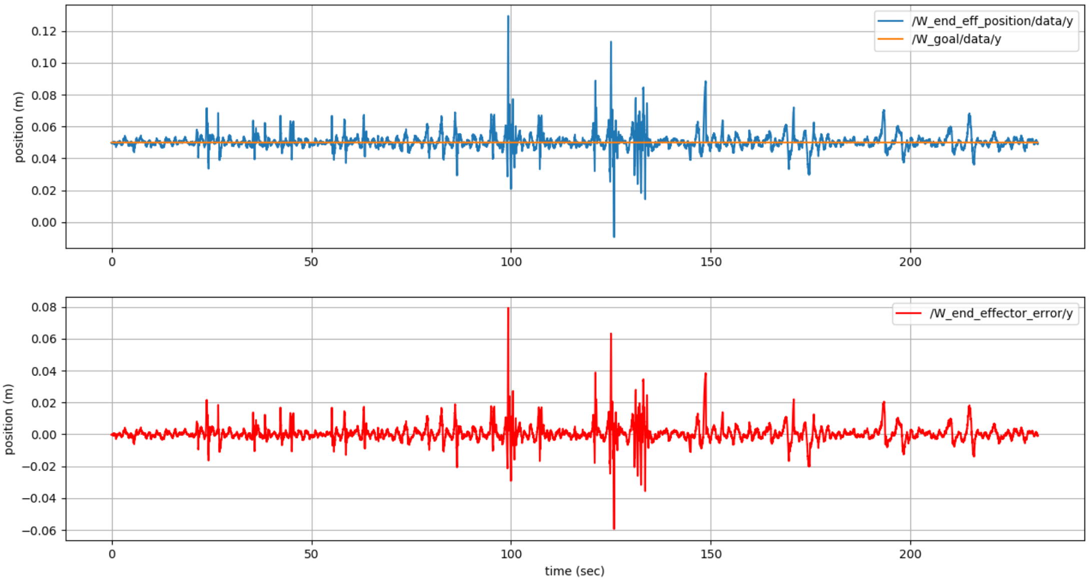


Figure 6.9: MAV's errors compensation routine results in $Y$-coordinate

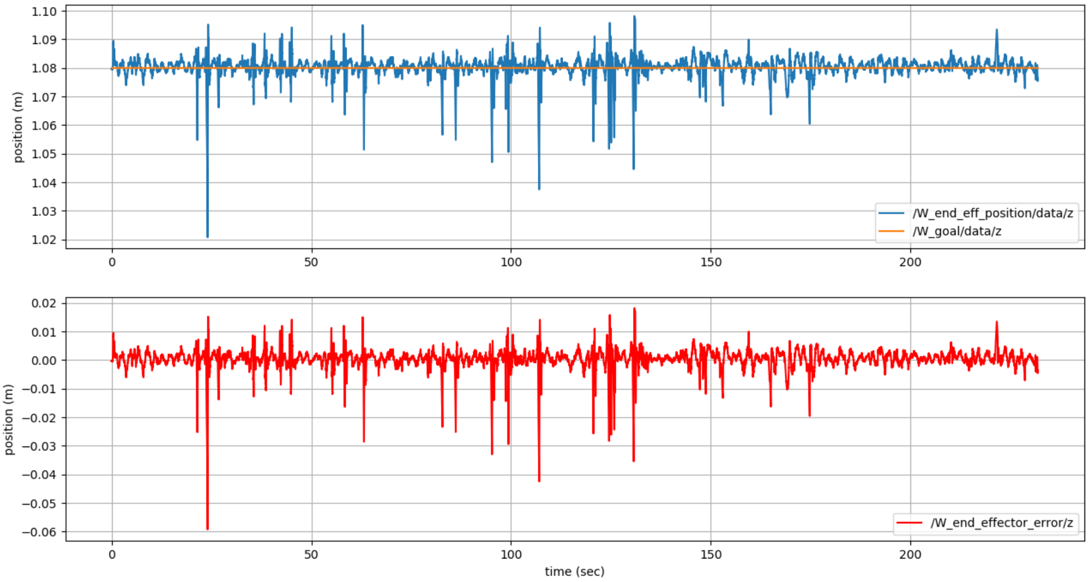


Figure 6.10: MAV's errors compensation routine results in $Z$-coordinate

Also for this last test we have computed error statistics which are now reported as usual in the next table:

| Error | $X$ (m) | $Y$ (m) | $Z$ (m) |
|---|---|---|---|
| Mean Value | $-0.000338$ | 0.000500 | 0.000301 |
| RMSE | 0.014773 | 0.005267 | 0.003776 |
| Standard Deviation | 0.014769 | 0.005244 | 0.003764 |

Table 6.3: Error statistics for VICON validation test

From the table it could be seen that results are very good: only in $X$-direction the error is noteworthy while along $Y$ and $Z$ it is very limited. It should also be noticed the presence of some peaks in all three plots, most of them are in fact caused by end-effector goal position being *outside* the workspace of Delta robot. In fact, if we consider only those time instants in which the goal point in inside the workspace, results are around 20% better, this holds in particular for $X$ axis plot.

In conclusion, by taking into account error statistics and these last considerations we are very satisfied with obtained results and we can state that *most of the time* end-effector position accuracy is guaranteed to be lower than $1\,cm$ as required at the beginning of the project. Moreover, the manipulator is able to compensate quite satisfactorily for MAV pose errors and we strongly believe that results will be even better once previously mentioned problems will be solved.

## 6.3 Conclusions and Future Work

As mentioned also in previous sections, some possible sources of errors causing performance degradation have been already identified. These ones, together with other small imperfections will be now listed for completeness:

- non perfect machining of components
- non perfect coupling between ball joints and carbon-fiber rods
- unoptimized homing offset of Dynamixel servos
- unoptimized servos PID parameters
- non perfect calibration of VICON markers.

Looking at future developments, it will be undoubtedly useful try to solve aforementioned issues; this seems to be quite feasible and it would have been already done in this thesis if some extra time had been available. It seems also interesting trying to develop some kind of overall control for the entire system (MAV +

Delta); anyway, this last task seems to be much more challenging and needs to be properly handled. Last but not least, a decisive further step in terms of attainable applications could be done by endowing our Delta robot with a specifically designed end-effector in order to provide the system with some kind of *manipulation* capabilities.
For the sake of clarity, these and other proposals left for future work are now grouped in the following list:

- improve mechanical machining
- servos homing improvement (could be done by exploiting least square optimization)
- improve PID tuning for servos
- develop overall control approach and compare with obtained results
- design and implement one or more suitable end-effector (based on specific applications) for adding manipulation capabilities
- implement some kind of compliant control for safe interactions.

In conclusion, in this thesis project a 3 DOF Delta manipulator for aerial interaction has been developed. After an initial literature research, the mentioned system has been firstly studied from a kinematics and dynamics point of view; then, the optimal geometric parameters have been found by means of an optimization problem based on a genetic algorithm.
Subsequently, the mechanical design of each single component for our Delta robot has been obtained by using CAD software and then validated by means of finite element analysis. Suitably powerful Dynamixel servo motors have been selected and mounted on the manufactured robot.
Moreover, a control framework based on an inverse kinematic approach has been studied and implemented in order to provide the robot with the ability of compensating for MAV errors. Robot Operating System has been the basic software tool on which the control routine and other low level control features have been developed, two main ROS packages have been implemented and used for controlling our Delta system: Dynamixel workbench, provided by Robotis and then suitably adjusted to our needs, for the control of Dynamixel servos, and Delta controller, which is the high level control unit responsible for the interface with the external inputs and the calculation of corresponding joint angles.
Finally, three different tests have been carried on to evaluate three different aspects of the system, but particularly to measure the ability of our Delta robot at compensating for MAV pose errors. Results have confirmed the goodness of the designed system but highlighted also room for improvements. Plots and statistics have been reported to show system performances and formulate hypothesis about

possible problems sources.
We are thus very satisfied with the resulting Delta robot and the obtained results; anyway, future improvements could be made to bring out the best from this system and are thus strongly encouraged.

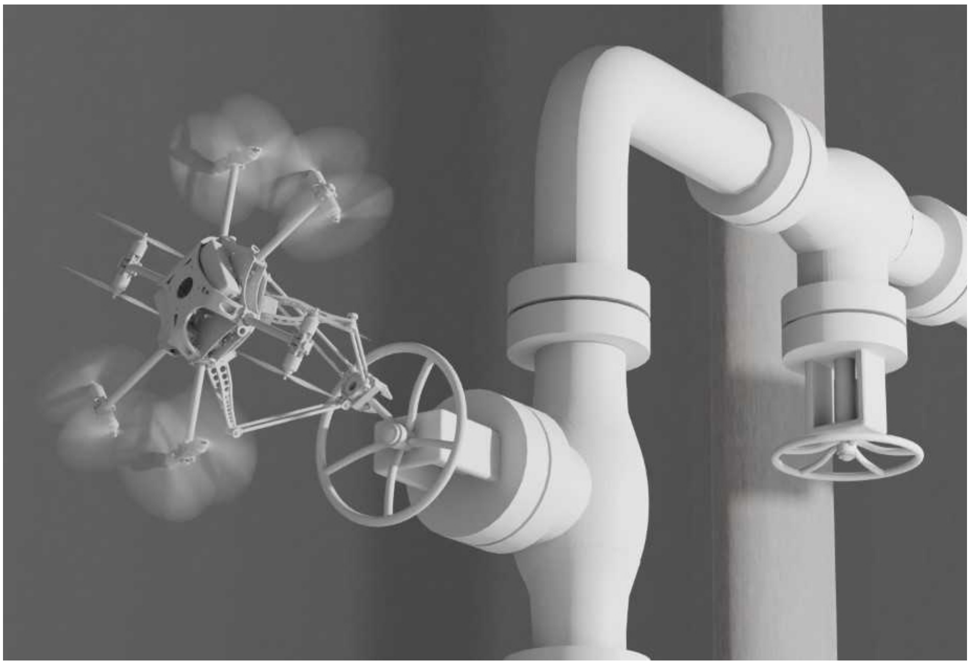


Figure 6.11: Delta robot mounted on MAV performing a manipulation task (rendering by Christian Lanegger)

# Bibliography

[1] (2013) The rise of the micro air vehicle. The Engineer.

[2] A. Ollero and B. Siciliano, *Aerial Robotic Manipulation Research, Development and Applications.* Springer, 2019.

[3] K. Bodie, M. Brunner, M. Pantic, S. Walser, P. Pfändler, U. Angst, R. Siegwart, and J. Nieto, "An omnidirectional aerial manipulation platform for contact-based inspection."

[4] X. Ding, P. Guo, K. Xu, and Y. Yu, "A review of aerial manipulation of small-scale rotorcraft unmanned robotic systems," *Chinese Journal of Aeronautics*, vol. 32, pp. 200–214, 2019.

[5] M. Kamel, K. Alexis, and R. Siegwart, "Design and modeling of dexterous aerial manipulator," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[6] A. E. Jimenez-Cano, J. Braga, G. Heredia, and A. Ollero, "Aerial manipulator for structure inspection by contact from the underside," *International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[7] A. Q. Keemink, M. Fumagalli, S. Stramigioli, and R. Carloni, "Mechanical design of a manipulation system for unmanned aerial vehicles." Institute of Electrical and Electronics Engineers Inc., 2012, pp. 3147–3152.

[8] P. Chermprayong, K. Zhang, F. Xiao, and M. Kovac, "An integrated delta manipulator for aerial repair: A new aerial robotic system," *IEEE Robotics and Automation Magazine*, pp. 54–66, 3 2019.

[9] S. Cho and D. H. Shim, "Development of a vision-enabled aerial manipulator using a parallel robot," *Trans. JSASS Aerospace Tech. Japan*, pp. 27–36, 2016.

[10] Y. D. Patel and P. M. George, "Parallel manipulators applications—a survey," *Modern Mechanical Engineering*, 2012.

[11] T. W. Danko, K. P. Chaney, and P. Y. Oh, “A parallel manipulator for mobile manipulating uavs.” IEEE Computer Society, 2015.

[12] R. Williams, “The delta parallel robot: Kinematics solutions,” 2016.

[13] A. Olsson, “Modeling and control of a delta-3 robot,” Master’s thesis, 2009.

[14] P. Guglielmetti and R. Longchamp, “Task space control of the delta parallel robot,” *IFAC Motion Control for Intelligent Automation*, 1992.

[15] ——, “A closed form inverse dynamics model of the delta parallel robot,” *IFAC Proceedings Volumes*, 1994.

[16] J. Brinker, B. Corves, and M. Wahle, “A comparative study of inverse dynamics based on clavel’s delta robot.” National Taiwan University, 2015.

[17] M. A. Laribi, L. Romdhane, and S. Zeghloul, “Analysis and dimensional synthesis of the delta robot for a prescribed workspace,” *Mechanism and Machine Theory*, vol. 42, pp. 859–870, 7 2007.

[18] V. Mallawaarachchi. (2017) Introduction to genetic algorithms — including example code.

[19] S. A. Conyers, M. J. Rutherford, and K. P. Valavanis, “An empirical evaluation of ground effect for small-scale rotorcraft.” Institute of Electrical and Electronics Engineers Inc., 2018.