Università degli Studi di Padova

*CISAS - Centro d'Ateneo di Studi ed Attività Spaziali G. Colombo*

Corso di Dottorato in Scienze, Tecnologie e Misure Spaziali

Curriculum in Sciences and Technologies for Aeronautics and Satellite Applications

---

# Design of a robotic arm for laboratory simulations of spacecraft proximity navigation and docking

Direttore del Corso:
**Ch.mo Prof. Giampiero Naletto**

Supervisore:
**Ch.mo Prof. Alessandro Francesconi**

Dottorando: **Andrea Antonello**

XXIX° Cycle – 2016

*Per te, Elsa.*

# CONTENTS

# ACKNOWLEDGMENTS

This thesis embraces all the efforts of the last three years of my life as a Ph.D. student at the University of Padova. In this time frame I had the opportunity of being "initiated" to research and to get in touch with different cultures and mindsets, and for this I am grateful to my supervisor, prof. Alessandro Francesconi, who made all this possible.

Firstly, I want to thank the SSG team at CISAS "G. Colombo", Dr. Olivieri in particular, with whom I created a proficuous academic and human relationship. Also, I want to thank the Department of Information Engineering, especially Dr. Carli and Dr. Carron, for the exciting collaborations and the solid friendship established in the last years.

This journey gave me the chance to meet international researchers in Boston and in Atlanta, who added value to the way I approach and solve new problems. In particular, the team at Boston University, led by prof. Mac Schwager and the Computer Science and Artificial Intelligence Laboratory at MIT, led by prof. E. H. Adelson, who vastly expanded my knowledge in the fields of robotics and haptics.

In the last two years, my research path has been heavily influenced by the work and knowledge of prof. Panagiotis Tsiotras from Georgia Institute of Technology and the entire team at the Dynamics and Control Systems Laboratory: for the unceasing help and for the support provided, I am grateful to prof. Tsiotras and to Alfredo.

Finally, I want to spend some words of appreciation to the people who made this long journey much more pleasant and a little less stressful: to my Bostonian friends, Rodrigo, Damien, Aws, Nicola, Alice and Morane, with whom I forged a long lasting friendship. To the people I met in Atlanta,

Alfredo and Tom, who helped me relieve the stress of the last months. Finally, but most importantly, to the people of Padova: Anna, Lorenzo, Andrea (3.1), Laura, Lucio and MemmoHouse, who made me appreciate this city little by little.

Lastly, I want to thank my family and my parents for being a fixed point in my life, in good and bad times.

And thanks to you, Elsa, for still being a part of who I am.

Padova,
January 2017

# ABSTRACT

The increasing number of human objects in space has laid the foundation of a novel class of orbital missions for servicing and maintenance. The main goal of this thesis is the development, building and testing of a robotic manipulator for the simulation of orbital maneuvers, with particular attention to Active Debris Removal (ADR) and On-Orbit Servicing (OOS).

There are currently very few ways to reproduce microgravity in a non-orbital environment: among the main techniques, it is worth mentioning parabolic flights, pool simulations and robotic facilities. Parabolic flights allow to reproduce orbital conditions quite faithfully, but simulation conditions are very constraining. Pool simulations, on the other hand, have fewer constrictions in terms of cost, but the drag induced by the water negatively affects the simulated microgravity. Robotic facilities, finally, permit to reproduce indirectly (that is, with an appropriate control system) the physics of microgravity. State of the art on 3D robotic simulations is nowadays limited to industrial robots facilities, that bear conspicuous costs, both in terms of hardware and maintenance.

This project proposes a viable alternative to these costly structures. Through dedicated algorithms, the system is able to compute in real time the consequences of these contacts in terms of trajectory modifications, which are then fed to the hardware in the loop (HIL) control system. Moreover, the governing software can be commanded to perform active maneuvers and relocations: as a consequence, the manipulator can be used as the testing bench not only for orbital servicing operations but also for attitude control systems, providing a faithful, real-time simulation of the zero-gravity behavior.

Furthermore, with the aid of dynamic scaling laws, the potentialities of the facility can be exponentially increased: the simulation environment is not longer bounded to be as big as the robot workspace, but could be several orders of magnitude bigger, allowing for the reproduction of otherwise preposterous scenarios.

The thesis describes the detailed mechanical design of the facility, corroborated by structural modeling, static and vibrational finite element verification. A strategy for the simulation of impedance-matched contacts is presented and an analytical control analysis defines the set of allowable inertial properties of the simulated entities. Focusing on the simulation scenarios, an innovative information theoretic approach for simultaneous localization and docking has been designed and applied for the first time to a 3D rendezvous scenario.

Finally, in order to instrument the facility's end effector with a consistent sensor suite, the design and manufacturing of an innovative Sun sensor is proposed.

1

# INTRODUCTION

## 1.1 THE HISTORY OF ROBOTICS

The image of the robot as a mechanical artifact has its origins in the 1940s, when writer Isaac Asimov conceived the robot as an *automaton* of human looks but without emotions. Asimov describes the term **robotics** as the science committed to the study of robots, founded on three fundamental laws [1]:

1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.
2. A robot must obey the orders given by human beings, except when such orders would conflict with the first law.
3. A robot must protect its own existence, as long as such protection does not conflict with the first or second law.

It is not until the 1960s that robots started to be seen as viable manufacturing devices for the industry, along with CAD and CAM systems, which later influenced conspicuously the advances of technology. The principal milestones of modern automation can be summarized as follows [2]:

1947 - first electric powered teleoperator is developed

1956 - George Devol and Joseph Engelberger form the worlds first robotics company, Unimation.

1961 - Unimate, the worlds first industrial robot, goes to work on a General Motors assembly line

1966 - the Stanford Research Center begins development of Shakey, the first mobile robot

1978 - Unimation develops the PUMA robot

1979 - the SCARA robot design is introduced in Japan

1989- chess playing robot HiTech defeats chess master Arnold Denker

1997 - Sojourner rover performs semi-autonomous operations on Mars

2000 - Hondas humanoid robot ASIMO steps onto the stage.

2001 - Canadarm2 was launched into orbit and attached to the ISS

2004 - Cornell University revealed a robot capable of self-replication

## 1.2 STATE OF THE ART

In the aerospace industry, robotics has its leading edge applications. The two principal macroareas of interest are the Orbital Robotics and the Planetary Rovers [3].

Orbital Robotics consists in the implementation of manipulation and mobility for orbital operations and servicing scenarios. Planetary Rovers, on the other hand, address planetary exploration and surface manipulation.

Orbital robotics, due to space environment (radiation, micro-gravity, thermal stresses, etc.) poses unique challenges to robot and robot algorithms, and sets the need for new and innovative autonomous systems.

The design of servicing operations and devices is probably one of the most important research field in space robotics. Servicing operations range from simple inspection to upgrade of components and refueling [4]. Historical analysis indicates that the combination of the 5% failure rate of launch vehicles coupled with 9% failure rate of satellites during their operational lives will cause the failing of 1/7 of the satellites before the expected end of life (EOL) [5]. Nowadays, the usual approach in trying to avoid these failures is to use proven (usually a synonymous for obsolete) technology and to incorporate massive redundancy. Although the use of consolidated technology helps to mitigate mission risk, it also has the negative effect of limiting satellite performances.

The increase of costs associated with growing complexity of payloads have led to the need of augmenting satellite design lifetimes in order to obtain a sufficient investment return. One downside of this increased lifetime is the inability to update the hardware and software with modern avionics, in an era governed by Moores law[1]. This slowdown limits the agility of satellite operators in capturing emergent terrestrial markets [6].

All these limitations and the substantial absence of a maintenance program for satellites are pushing hard for the development of on-orbit servicing (OOS). Among the main operations that fall under the acronym OOS, the most important are:

- *Inspection*: the observation of a space object in order to gather information about its status.
- *Relocation*: the external movimentation of an object that has attitude problems and the on-board systems are not able to finalize the correct operational configuration.
- *Augmentation*: in the case of a modular satellite, it is the upgrading of the obsolete hardware in favor of state of art technology.

[1] Moore law's affirms that there is a doubling of the processing speed of new computer chips every 18÷24 months.

- *Assembling*: the merging of mating modules to construct space systems that wouldn't be possible otherwise.
- *Restoration*: the refueling, docking, station keeping providing, repairing and replacing hardware.

Being able to fix and/or refurbish an out-of-order satellite with unmanned vessels might give rise to a multi millionaire business: NASA estimated the costs of a single Hubble servicing mission at $2 billion. If a robotic servicing satellite was to be sent instead, the economic savings would be enormous, not to mention the avoidance of human losses (which is not an unlikely scenario in a manned mission). Nowadays, a lot of space agencies and private companies are pushing in this direction.

The Canadian aerospace firm MacDonald, Dettwiler and Associates, for example, is developing the Space Infrastructure Servicing (SIS), a spacecraft for refueling of communication satellites in GEO orbits [7]. SIS is being designed to carry a toolkit able to open most of the ∼40 types of on-orbit fueling systems. Intelsat, which owns a 52 communications satellites fleet as of November 2016 [8], has shown a keen interest on the project, founding and sponsoring the inaugural mission with an investment of $280 millions [9].

NASA, on the other hand, has already developed and launched a demonstration technology named Robotic Refueling Mission (RRM). The servicing satellite has successfully performed an extensive series of robotically actuated fuel transfer on the ISS (2011) with the aid of the Canadarm manipulator. The long term goal of NASA for this project is to transfer this technology to the commercial market.

### 1.2.1 The need for relative attitude operations

Close-approach operations fall under the acronym of OOS: regardless of the operation to be carried out, it is necessary to be able to predict and control the chaser-target relative motion. The simulation of orbital operations presents a substantial difficulty in the research community, since the dynamics to which an object is subjected in space can not be fully reproduced on earth.

The are different types of facilities that focus on the reproduction of a micro-gravity condition in a laboratory setting, and each one has its advantages and disadvantages according to the phenomena that are under analysis. Among these facilities there are water pools, low friction tables, parabolic flights and robotic manipulators.

The focus of this thesis is on the latter category, since it is the only one among the ones mentioned that allow to have 6 degrees of freedom and an imposed microgravity behavior. Water pools, for example, take advantage of neutral buoyancy to achieve a pseudo weightlessness condition; this, however, can be heavily disturbed by drag force that the water exerts on the object.

Low friction table, on the other hand, if the setup is adequate (balanced platform and controlled planarity of the surface), allow to simulate microgravity; the only caveat is constituted by the limited dexterity of the system, which guarantees only 2 of the 3 translational degrees of freedom.

Parabolic flights enable to reproduce orbital conditions quite faithfully, but simulation conditions are very constraining. Moreover, this is clearly an expensive solution, not suitable for every-day testing campaigns.

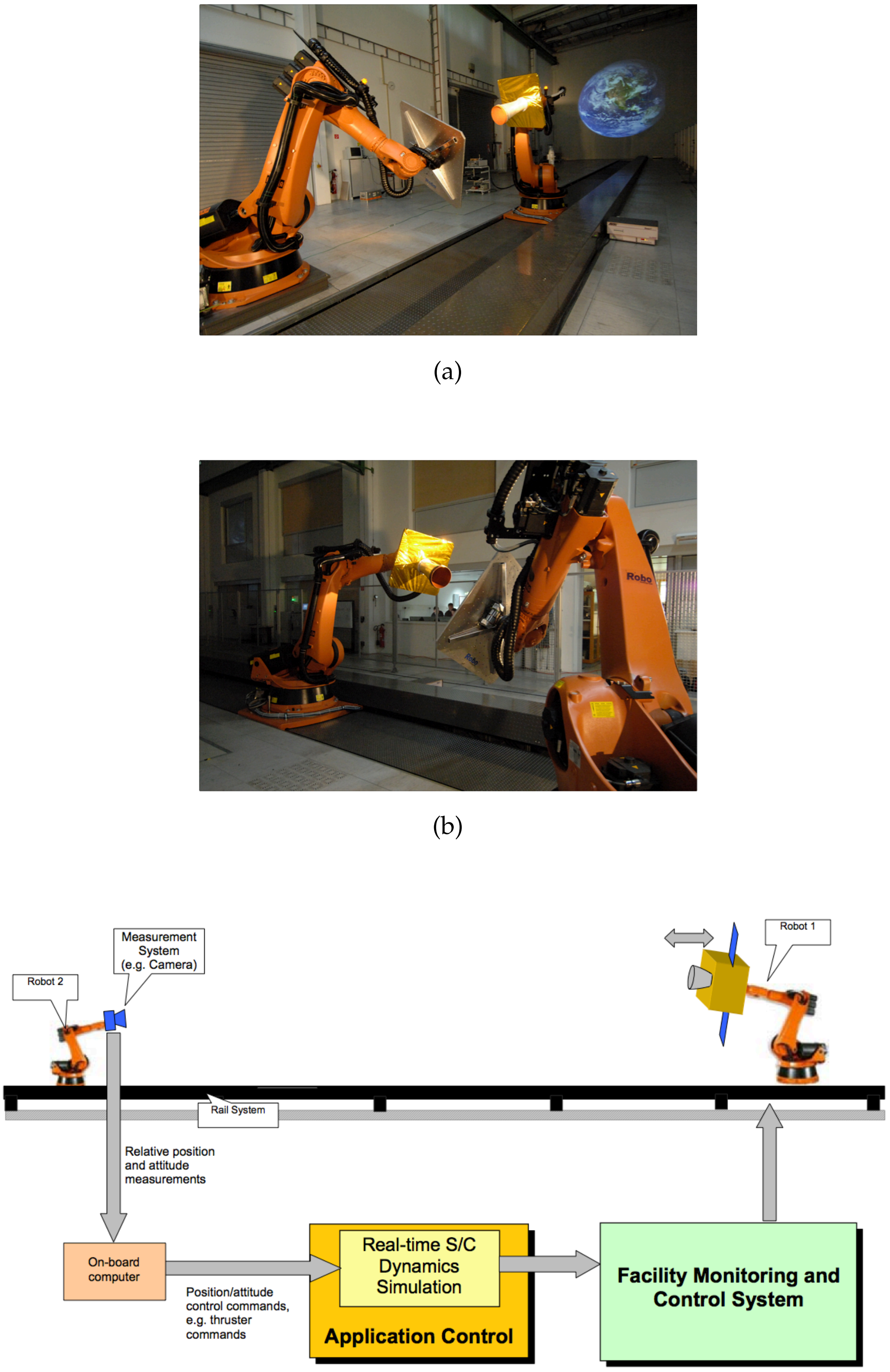
Robotic manipulators, ultimately, can be controlled to dynamically behave as an object in space. That is, the zero-gravity condition is obtained via software by imposing the motion characterized by the desired dynamics; the software can also perform orbital operations in which contact dynamics is present (docking, berthing, manipulation, etc.) and react dynamically with a compliant system [10, 11].

As of 2016, there exist very few facilities that are capable of performing microgravity experiments. One the most important is the EPOS *European Proximity Operations Simulator* experiment conducted by the Deutsches Zentrum für Luft und Raumfahrt (DLR) [12].

The original EPOS was designed as a joint-venture between DLR and ESA in the late 1980s, as the need for a rendezvous and docking (RvD) testing facility arose. In 1991, the facility began operations, and was constituted by three subsystems: a 6 DOF gantry, able to host a 100 kg payload at the end effector, a structure carrying the target object and an auxiliary illumination system to achieve realistic lighting conditions.

This system served for testing for almost 20 years and was renewed due to the demand for better RvD simulation accuracy. The current facility was built in 2009 and it's a joint effort between the DLR's GSOC and DLR's Robotics and Mechatronics Institute, which contributed to the robotic technology, on behalf of their solid background on the subject. The approaching vehicles are simulated via two anthropomorphic industrial robots, with the target robot fixed on the ground and the chaser mounted on a 25 m rail for extra mobility.

An industrial PC feeds in synchronous trajectories via a Simulink® interface and the control and measuring systems allow for a position and angular accuracy of respectively 2 mm and 0.2ą. All the trajectory simulation are carried out via an implementation of Clohessy-Wiltshire coordinate system.



(a)

(b)

Robot 2
Measurement System (e.g. Camera)
Robot 1
Rail System
Relative position and attitude measurements
On-board computer
Position/attitude control commands, e.g. thruster commands
Real-time S/C Dynamics Simulation
Application Control
Facility Monitoring and Control System

(c)

Figure 1: EPOS RvD simulation facility: laboratory configurations (a), (b) and conceptual operating diagram (c)

## 1.3 THESIS MOTIVATION

This project proposes a viable alternative to the huge and costly structures described in the pervious paragraph; through dedicated algorithms, the system will be able to simulate microgravity in a laboratory setting. Moreover, the governing software can be commanded to perform active maneuvers and relocations: as a consequence, the manipulator can be used as the testing bench for rendezvous scenario and orbital operations. Furthermore, with the aid of dynamic scaling laws, the potentialities of the facility can be exponentially increased: the simulation environment is not longer bounded to be as big as the robot workspace, but could be several orders of magnitude bigger, allowing for the reproduction of otherwise preposterous scenarios in a laboratory environment.

Finally, the robot itself can be used as part of the simulated maneuvers. Berthing operations and uncooperative target docking, for example, can be performed. This latter research field, uncooperative docking, as long with RvD rendezvous and docking operations, are under study at CISAS research center for Active Debris Removal as well as for on-orbit servicing: the manipulator presented in this thesis could serve as the main testing facility for the simulation and the verification of theoretical and numerical analysis.

The work done in the doctorate years will comprise the development and design of the robotic facility, the software simulation of collision and contacts, the detailed modeling of the arm mechanics and vibrational modes, the design and manufacturing of a novel attitude sun sensor and the development of an innovative information theoretic scenario for close range inspection and localization.

# 2

# PRELIMINARIES

### 2.0.1 Overview

Before embarking in the kinematic and dynamic analysis, it is necessary to identify the main components of a robotic system. Even for a complex architecture, it is always possible to identify a general block diagram [13]:

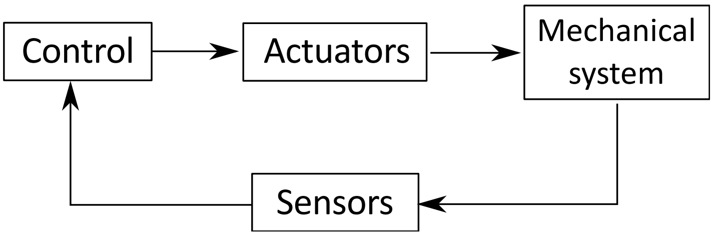


Figure 2: Robotic system components.

The core component is the mechanical system, made up of a manipulation apparatus (arms, links, end effectors, artificial hands) and a movement apparatus (wheels, crawlers, legs). The capability to execute a task is made possible by the actuators block, which provides motion to the manipulation and movement apparatus.

The connection with the outside world is made possible by the presence of sensors, enabling the acquirement of data on the internal status (*proprioceptive* sensors, such as encoders) and on the external status (*exteroceptive* sensors, such as force sensors or vision system)

Finally, the control block permits to make the whole system an harmonic working machine, reading data from the sensors and commanding the actuators with well-tuned control laws.

### 2.0.2 Mechanical structure

The main distinction between different robots concerns their mechanical structure. That is, the way in which the links are connected and the way in which they move with respect to each other. A robot manipulator is a sequence of rigid bodies (called *links*) which are connected by joints. The configuration is most of the times that of an open kinematic chain; usually, at the end of the manipulator, there is the end-effector, which provides the needed dexterity for the execution of tasks.

The mobility is ensured by the presence of joints, which can be of different type and can introduce one or multiples degrees of freedom[1].

Mechanical design considerations when building robots have narrowed the joint choices to two main types: revolute or prismatic. In a revolute joint, the connected bodies rotate with respect to a common axis, whereas in a prismatic joint they slide without rotation. Both of these configurations have a single degree of freedom. When more than one degree of freedom is needed, other less used joint options are available (Fig. 3).

For simplicity, industrial robots have usually single degree of freedom links. The number of DOF characterizes the mobility of the robot in the operational space: in order to arbitrarily position the end effector in 3D space, 6 DOF are required (excluding for the moment the singularities), 3 being translational and 3 rotational.

When a robot has less than 6 DOF, it will have some limitations on the end effector orientation in his working space; when, on the other hand,

[1] Note that, in the special case of singularity, they do not provide any contribution to the overall number of degrees of freedom.

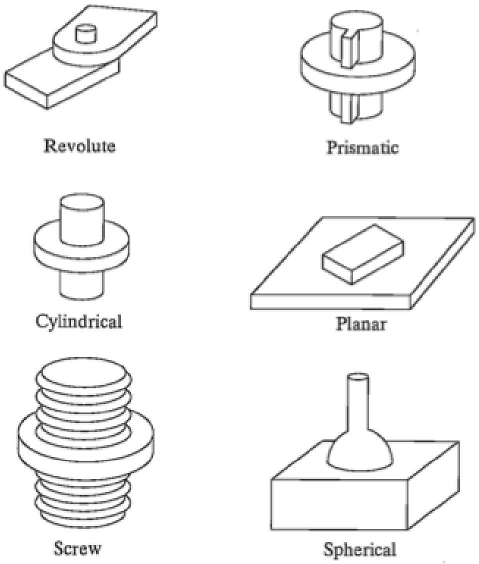


Figure 3: Joint configuration types

there are more than 6 DOF, the robot is kinematically redundant, and the same position in space can be obtained via several configurations.

Among the main robot configurations choices, there exist cartesian, cylindrical, spherical, SCARA, anthropomorphic [13]. In our case, since we are looking for the maximum dexterity, the anthropomorphic manipulator seems to be the best choice.

Among the requirements that need to be satisfied in this project, there is the workspace: the manipulator, in fact, has to have sufficient dexterity in a cube whose volume is at least 0.5 m×0.5 m×0.5 m. In the sizing analysis, the link lengths will be chosen in order to fulfill this requirement.

We summarize in the following sections the main results obtained in the kinematics and dynamics of the arm from a previous work by this author [10], which will serve as the starting point for all the further analysis.

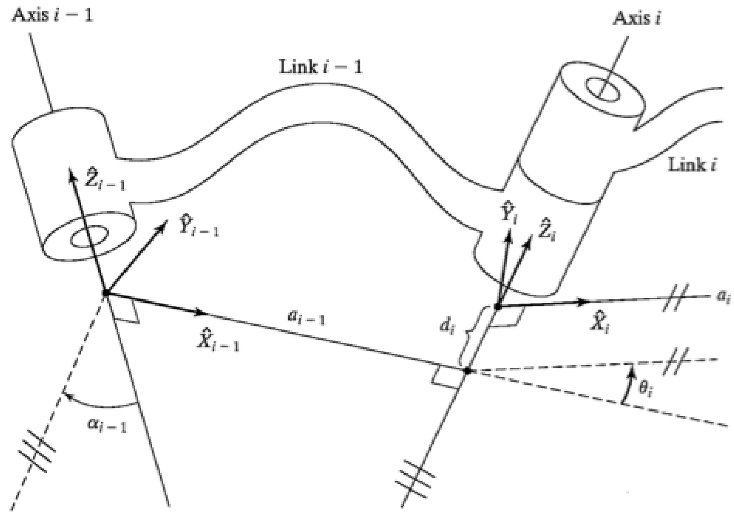


Figure 4: Link and joint notation schematic

## 2.1 KINEMATICS REVIEW

Kinematics studies of the motion of a body that considers the object without taking into account the dynamics causing the movement. This branch of robotics accounts for the study of the position and its higher order derivatives[2] (velocity, acceleration, jerk etc). The links are numbered starting from the base of the arm, which is fixed and is numbered as *link 0*. The first moving link is *link 1*, and so on, until the last link, which is *link n*.

Each link presents several characteristics that need to be considered during the design process, but as long as kinematics is concerned, we only need information about the relationship between the two neighboring joint axes. Here, the links will be treated as rigid bodies.

[2] Taken with respect to time or other variables.

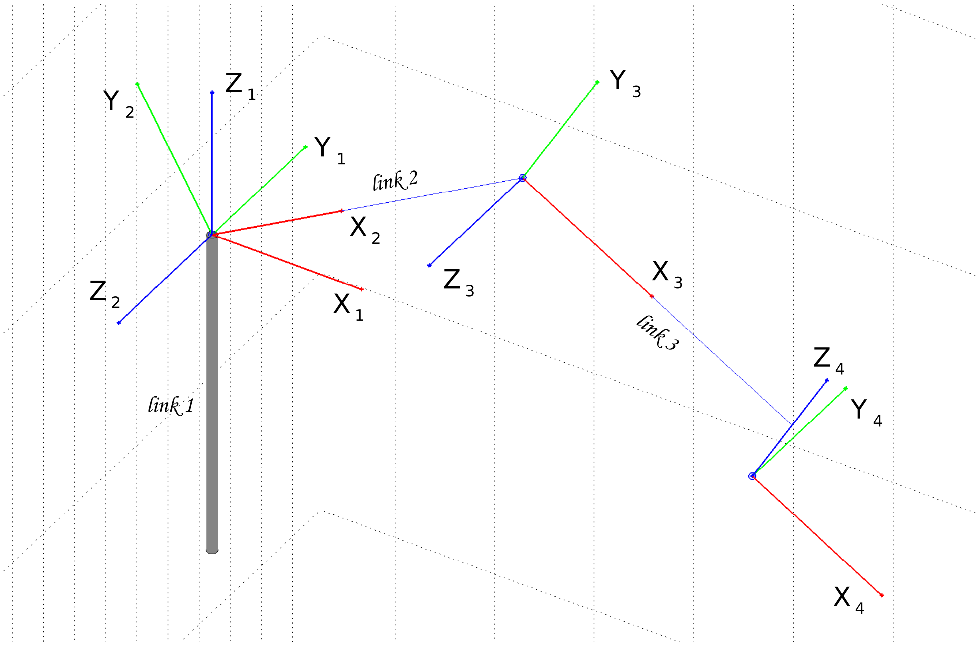


Figure 5: Frame configuration obtained via DH procedure.

## 2.2 DENAVIT-HARTENBERG CONVENTION

In order to describe the manipulator and to accomplish the kinematic and dynamic analysis it is necessary to implement a solid and recursive notation. The Denavit-Hartenberg convention defines the relative position and orientation of two consecutive links by determining the reference frames attached to each link and computing the coordinate transformations among them. The notation used in this work is presented in Fig. 4; for the complete description of the notation, the reader should refer to [10].

The first three frames of the robot, using the DH framework, can be visualized for a random configuration in Fig 5. As far as the end effector is concerned, the frames will have the same origin, and they are oriented as shown in Fig 6.

Once all the frames are defined, their characteristic parameters are

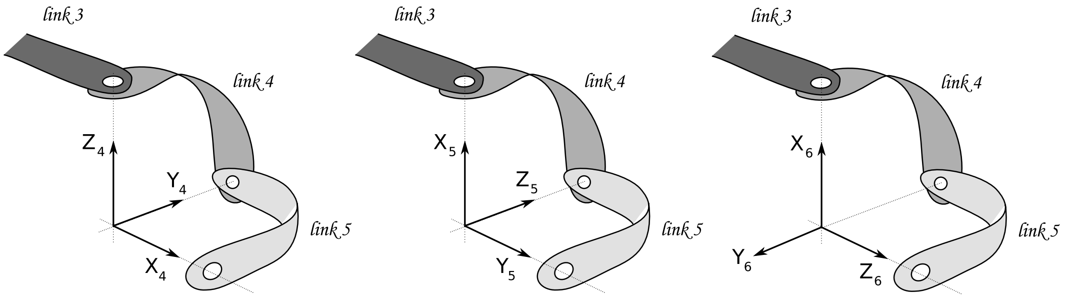


Figure 6: Frame configuration for end-effector structure.

stored in a matrix. For this application, the parameters are summarized in Tab. 1.

| Joint ($i$) | $\alpha_{i-1}$ | $a_{i-1}$ | $d_i$ | $\theta_i$ |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | $\theta_1$ |
| 2 | $\frac{\pi}{2}$ | 0 | -$d_2$ | $\theta_2$ |
| 3 | 0 | $l_2$ | -$d_3$ | $\theta_3$ |
| 4 | -$\frac{\pi}{2}$ | $l_3$ | -$d_4$ | $\theta_4$ |
| 5 | -$\frac{\pi}{2}$ | 0 | 0 | $\theta_5$ |
| 6 | -$\frac{\pi}{2}$ | 0 | 0 | $\theta_6$ |

Table 1: DH matrix containing the parameters for the frame definition.

Concluding the DH framework definition, the roto-translation matrices for the manipulator are:

$$
{}^0_1\mathbf{T} = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^1_2\mathbf{T} = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & 0 \\ 0 & 0 & -1 & d_2 \\ s\theta_2 & c\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$ {}^2_3\mathbf{T} = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & l_2 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 1 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^3_4\mathbf{T} = \begin{bmatrix} c\theta_4 & -s\theta_4 & 0 & l_3 \\ 0 & 0 & 1 & -d_4 \\ -s\theta_4 & -c\theta_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^4_5\mathbf{T} = \begin{bmatrix} c(\theta_5 - \frac{\pi}{2}) & -s(\theta_5 - \frac{\pi}{2}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s(\theta_5 - \frac{\pi}{2}) & -c(\theta_5 - \frac{\pi}{2}) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^5_6\mathbf{T} = \begin{bmatrix} c\theta_6 & -s\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_6 & -c\theta_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

These matrices are a function of the joint variables only. Note that the last two transform-matrices present the same $4^{th}$ column: this means that the translation with respect to the previous frame is zero, and there is only a rotational transformation. This is due to the fact that the same origin was chosen for these frames (Pieper's hypothesis [10, 14]).

## 2.3 DIRECT KINEMATICS

With these matrices computed, we can introduce the direct kinematics problem (DK). Direct kinematics allows for the knowledge of the cartesian position of each link of a kinematic chain once the joint variables $\mathbf{q} = [q_1 \dots q_n]$ are know.

In a manipulator, the most important result that the DK procedure provides is certainly the knowledge of the Cartesian position and orientation of the end effector.

This is done by simply taking the product of the transforms:

$$ {}^0_N\mathbf{T}(\mathbf{q}) = {}^0_1\mathbf{T}(q_1)\,{}^1_2\mathbf{T}(q_2)\ \dots {}^{N-1}_{\ \ N}\mathbf{T}(q_N) \tag{1} $$

If the cartesian position is needed (in terms of $[p_x, p_y, p_z]$), we recall the general expression of a rototransitional matrix:

$$
{}^{0}_{N}\mathbf{T}(\mathbf{q}) = \left[\begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & {}^{0}p_{x,N} \\ r_{21} & r_{22} & r_{23} & {}^{0}p_{y,N} \\ r_{31} & r_{32} & r_{33} & {}^{0}p_{z,N} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \tag{2}
$$

In this fashion, it is possible to instantly know the position of each joint in the Cartesian space:

$$
{}^{0}\mathbf{p_j} = {}^{0}_{j}\mathbf{T}(1:3,4) \tag{3}
$$

Where ${}^{0}\mathbf{p_j}$ is the position of the $j$-th joint with respect to the origin. The orientation can be obtained in a similar way from from Eq 2:

$$
{}^{0}_{j}\mathbf{R}(\mathbf{q}) = {}^{0}_{j}\mathbf{T}(1:3,1:3) \tag{4}
$$

## 2.4 INVERSE KINEMATICS

Inverse kinematics (IK) consists in the solution of the Cartesian-to-joint variables problem. The solution to this problem is less straightforward than the direct kinematics case, and it is strictly linked to the geometrical configuration of the manipulator. Not for all cases, in fact, there exists an analytical solution; moreover, for those cases whose analytical solution is available, this is usually difficult and time consuming. However, some configurations might provide large simplifications for the inverse kinematics problem. A 6 DOF robot, for example, does not have a closed form solution in general. However, if three consecutive axes intersect at a point, then Piepers solution can be applied [14, 13].

In this thesis, the last three axes of the manipulator intersec: the origins of frames ${}^{3}_{4}\mathbf{T}$, ${}^{4}_{5}\mathbf{T}$, ${}^{5}_{6}\mathbf{T}$, in fact, are coincident. The merging point can be calculated in base coordinates as:

$$ {}^{0}\mathbf{p_4} = {}^{0}_{1}\mathbf{T}\,{}^{1}_{2}\mathbf{T}\,{}^{2}_{3}\mathbf{T}\ {}^{3}\mathbf{p_4} = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} \tag{5} $$

from which:

$$ {}^{0}\mathbf{p_4} = {}^{0}_{1}\mathbf{T}\,{}^{1}_{2}\mathbf{T}\,{}^{2}_{3}\mathbf{T} \begin{bmatrix} a_3 \\ -d_4 s\alpha_3 \\ d_4 c\alpha_3 \\ 1 \end{bmatrix} \tag{6} $$

we can also state that:

$$ {}^{0}\mathbf{p_4} = {}^{0}_{1}\mathbf{T}\,{}^{1}_{2}\mathbf{T} \begin{bmatrix} f_1(\theta_3) \\ f_2(\theta_3) \\ f_3(\theta_3) \\ 1 \end{bmatrix} \tag{7} $$

where we defined:

$$ \begin{bmatrix} f_1(\theta_3) \\ f_2(\theta_3) \\ f_3(\theta_3) \\ 1 \end{bmatrix} = {}^{2}_{3}\mathbf{T} \begin{bmatrix} a_3 \\ -d_4 s\alpha_3 \\ d_4 c\alpha_3 \\ 1 \end{bmatrix} \tag{8} $$

Using ${}^{2}_{3}\mathbf{T}$, the following expressions for $f$ can be obtained:

$$ \begin{cases} f_1 = a_3 c_3 + d_4 s\alpha_3 s\alpha_3 + a_2 \\ f_2 = a_3 c\alpha_2 s_3 - d_4 s\alpha_3 c\alpha_2 c_3 - d_4 s\alpha_2 c\alpha_3 - d_3 s\alpha_2 \\ f_3 = a_3 s\alpha_2 s_3 - d_4 s\alpha_3 s\alpha_2 c_3 + d_4 c\alpha_2 c\alpha_3 + d_3 c\alpha_2 \end{cases} $$

We define also the following parameters:

$$\begin{cases} g_1 = c_2 f_1 - s_2 f_2 + a_1 \\ g_2 = f_1 c\alpha_1 s_2 + f_2 c\alpha_1 c_2 - f_3 s\alpha_1 - d_2 s\alpha_1 \\ g_3 = f_1 s\alpha_1 s_2 + f_2 s\alpha_1 c_2 + f_3 c\alpha_1 + d_2 c\alpha_1 \end{cases}$$

And we can write, with some algebra:

$$^0\mathbf{p_4} = \begin{bmatrix} c_1 g_1 - s_1 g_2 \\ s_1 g_1 + c_1 g_2 \\ g_3 \\ 1 \end{bmatrix} \tag{9}$$

The square magnitude of $^0\mathbf{p_4}$, using Eq. 9, is:

$$r = f_1^2 + f_2^2 + f_3^2 + a_1^2 + d_2^2 + 2d_2 f_3 + 2a_1(c_2 f_1 - s_2 f_2) \tag{10}$$

We define some simplifying parameters:

$$\begin{cases} k_1 = f_1 \\ k_2 = -f_2 \\ k_3 = f_1^2 + f_2^2 + f_3^2 + a_1^2 + d_2^2 + 2d_2 f_3 \\ k_4 = f_3 c\alpha_1 + d_2 c\alpha_1 \end{cases}$$

And we finally state:

$$\begin{cases} r = (k_1 c_2 + k_2 s_2) 2a_1 + k_3 \\ z = (k_1 s_2 - k_2 c_2) s\alpha_1 + k_4 \end{cases}$$

It can be noted that the dependence on $\theta_1$ has been eliminated and the dependence from $\theta_2$ has been drastically simplified. The first step is to consider the solution for $\theta_3$. We distinguish three cases:

1. If $a_1 = 0$, then $r = k_3$. Since $k_3$ is a function of $\theta_3$ only, we can obtain a quadratic equation in $tan\frac{\theta_3}{2}$ which yields the solution for $\theta_3$

2. If $s\alpha_1 = 0$, then $z = k_4$. We can obtain a quadratic equation and solve for $\theta_3$

3. If $a_1 \neq 0$ and $s\alpha_1 \neq 0$, we can eliminate with an auxiliary equation $s_2$ and $c_2$, and we end up with a $4^{th}$ degree equation, which will be solved for $\theta_3$

In our particular case, $a_1 = 0$ and we can compute $\theta_3$ referring to the first bullet point. We then focus on the solution of $\theta_2$ and $\theta_1$ which is trivial. At this point, we know $\theta_1$, $\theta_2$ and $\theta_3$. Since the three last axis are intersecting, it is possible to compute the remaining angles with the aid of elementary matrix transform algebra.

In the real case scenario, a required attitude will have to be obtained the end effector with reference to the base frame, which is ${}^0_6\mathbf{T}_{\mathbf{att}}$. From Pieper's solution, ${}^0_3\mathbf{T}$ can be computed:

$${}^0_4\mathbf{T} = {}^0_1\mathbf{T}(q_1)\,{}^1_2\mathbf{T}(q_2)\,{}^2_3\mathbf{T}(q_3) \tag{11}$$

The desired orientation, ${}^0_6\mathbf{T}$, differs from the actual orientation ${}^0_3\mathbf{T}$ only due to the action of the last three joints, whose contribution is described by the following matrix:

$${}^3_6\mathbf{T}(q_4, q_5, q_6) = {}^0_3\mathbf{T}^{-1}\,{}^0_6\mathbf{T} \tag{12}$$

From this matrix, the computation of the angle is straightforward, and we proceed algebraically from the symbolic expression of ${}^3_6\mathbf{T}$, containing the DH parameters and trigonometric functions of $q_4, q_5, q_6$.

## 2.5 DIFFERENTIAL KINEMATICS

The relationship between joint velocities and end effector velocities is provided by the Jacobian matrix. With the knowledge of this matrix and the end effector desired trajectory (expressed in terms of velocities), the kinematic problem can be solved: joint velocities can be directly obtained and then, with a numerical integration, also their instantaneous position.

The computation of the Jacobian is done with the geometric approach [10, 14], by computing the contributions of each joint velocity to the components of the end-effector cartesian linear and angular velocities.

### 2.5.1 Geometric approach

From solid mechanics, we recall that the velocity of point P belonging to a rigid body moving in 3D space, with respect to frame A, can by expressed as [13, 15]:

$$^{A}V_{P} = {}^{A}V_{B} + {}^{A}_{B}R\,{}^{B}V_{P} \tag{13}$$

Where B is a reference matrix fixed to the body. In this case we consider the motion of frame B as a pure translation. If a rotation is present, Eq. 15 becomes:

$$^{A}V_{P} = {}^{A}V_{B} + {}^{A}_{B}R\,{}^{B}V_{P} + {}^{A}\Omega_{B} \times {}^{A}_{B}\,{}^{B}P \tag{14}$$

Where ${}^{A}\Omega_{B}$ is the angular velocity of the body with respect to frame A. By using this equation and its derivatives we can approach the Jacobian matrix derivation as well as the dynamics. For the solution of differential kinematics, the velocities of each link (linear and angular) are needed: a technique called velocity propagation will be used in order to obtain a recursive and implementable sequence. We start from the base: *frame 0* will be considered the fixed, reference frame. We define $v_i$ as the linear velocity of the origin of the frame attached to link $i$; same notation applies to $w_i$. The superscript on the left of a parameter represents the frame in which it is expressed.

We compute the velocities in order, from the base to the end effector.

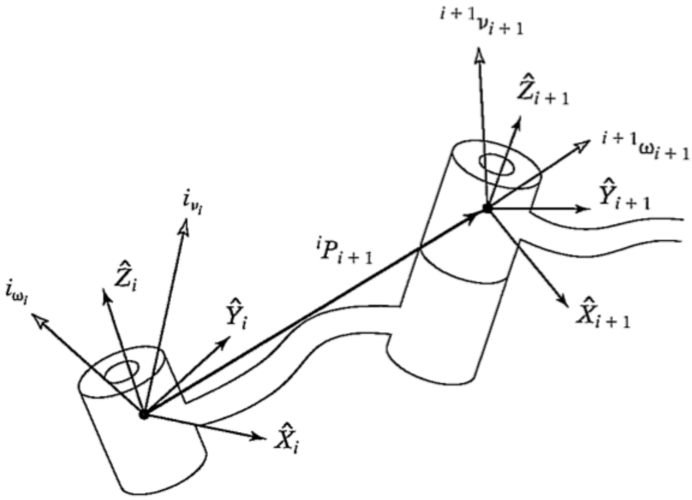


Figure 7: Velocity vectors for two adjoining links.

Referring to Fig. 7, velocity of *link i+1* will be the one of *link i* with the addition of a contribution from *joint i+1*. That is:

$$ {}^{i}v_{i+1} = {}^{i}v_{i} + {}^{i}\omega_{i} \times {}^{i}P_{i+1} \tag{15}$$

Where ${}^{i}P_{i+1}$ is the vector connecting the two links. There is no need to calculate this, since the ${}^{i}_{i+1}\mathbf{T}$ matrices have this information stored in their fourth column. To get the velocity of *link i+1* expressed in *frame i+1*, we rearrange the previous equation:

$$ {}^{i+1}v_{i+1} = {}^{i+1}_{i}R\,({}^{i}v_{i} + {}^{i}\omega_{i} \times {}^{i}P_{i+1}) \tag{16}$$

Adjusting the reference frames, we have:

$$ {}^{i}w_{i+1} = {}^{i}w_{i} + {}^{i}_{i+1}R\,\dot{\theta}_{i+1}\,{}^{i+1}\hat{k}_{i+1} \tag{17}$$

Where:

$$ \dot{\theta}_{i+1}\,{}^{i+1}\hat{k}_{i+1} = {}^{i+1}\begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_{i+1} \end{bmatrix} \tag{18}$$

The same equation expressed in *frame i+1* becomes:

$$^{i+1}w_{i+1} = {}^{i+1}_{i}R\,{}^{i}w_i + \dot{\theta}_{i+1}\,{}^{i+1}\hat{k}_{i+1} \tag{19}$$

We can finally write the system of equations that will be used to propagate the velocities from $i = 0$, the base frame, to $i = N$, which corresponds to the velocities (linear and angular) of the end effector.

$$\begin{cases} ^{i+1}v_{i+1} = {}^{i+1}_{i}R\,({}^{i}v_i + {}^{i}\omega_i \times {}^{i}P_{i+1}) \\ ^{i+1}w_{i+1} = {}^{i+1}_{i}R\,{}^{i}w_i + \dot{\theta}_{i+1}\,{}^{i+1}\hat{k_{i+1}} \end{cases}$$

For a 6 DOF manipulator, the joint and end effector velocities relationship can be written as:

$$\begin{bmatrix} \mathbf{v} \\ \omega \end{bmatrix} = \begin{bmatrix} \mathbf{J}_P \\ \mathbf{J}_O \end{bmatrix} \cdot \begin{bmatrix} \dot{\mathbf{q}} \end{bmatrix} \tag{20}$$

Where the Jacobian can be written as:

$$\begin{bmatrix} \mathbf{J}_P \\ \mathbf{J}_O \end{bmatrix} = \begin{bmatrix} \mathbf{z}_i \times (\mathbf{p}_e - \mathbf{p}_i) \\ \mathbf{z}_i \end{bmatrix} \tag{21}$$

Which, in our case, becomes:

$$\begin{bmatrix} \mathbf{J}_P \\ \mathbf{J}_O \end{bmatrix} = \begin{bmatrix} \mathbf{z}_1 \times \tilde{\mathbf{p}}_\mathbf{1} & \mathbf{z}_2 \times \tilde{\mathbf{p}}_\mathbf{2} & \mathbf{z}_3 \times \tilde{\mathbf{p}}_\mathbf{3} & \mathbf{z}_4 \times \tilde{\mathbf{p}}_\mathbf{4} & \mathbf{z}_5 \times \tilde{\mathbf{p}}_\mathbf{5} & \mathbf{z}_6 \times \tilde{\mathbf{p}}_\mathbf{6} \\ \mathbf{z}_1 & \mathbf{z}_2 & \mathbf{z}_3 & \mathbf{z}_4 & \mathbf{z}_5 & \mathbf{z}_6 \end{bmatrix} \tag{22}$$

Where $\tilde{\mathbf{p}}_\mathbf{i} = \mathbf{p}_e - \mathbf{p}_i$. The elements of the matrix are thoroughly explained in [10, 14]. Equations 22 can be easily solved with the aid of a MATLAB® code; the value of **J** depends on the instantaneous configuration and its symbolic expression is available in the Appendix of a previous publication by this author [10].

### 2.5.2 Inverse differential kinematics

The inverse kinematics problem is:

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}(\bar{q}) \cdot \mathbf{v} \tag{23}$$

From this vector, since $\mathbf{v}$ is known from the trajectory planning, we can obtain the joint variable position using an integration:

$$\mathbf{q}(t) = \int_{0}^{T} \mathbf{q}(t)\,dt \; + \; \mathbf{q}(0) \tag{24}$$

The initial position $\mathbf{q}(t = 0)$ can be obtained with the inverse kinematics method presented in the previous section.

Eq. 24 is implemented in the code as a discrete linear expression, using Heun's numerical integration, which ultimately yields:

$$\mathbf{q}(t_{i+1}) = \mathbf{q}(t_i) + \dot{\mathbf{q}}(t_i)\,\Delta t\,\mathbf{q}(t_{i+1}) \tag{25}$$

At this point, it is possible to summarize the procedure with a block diagram (Fig. 8). If we insert the integration method, then the solution procedure can be represented by the blocks in Fig. 9. In the diagram, Euler's integration method is implemented.

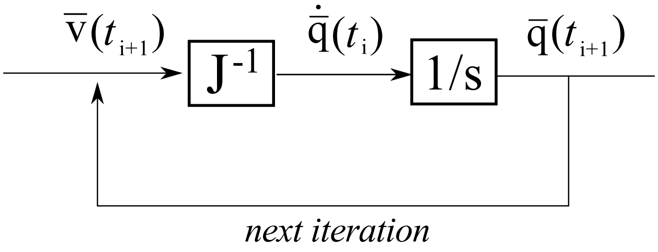


Figure 8: Inverse differential kinematics diagram.

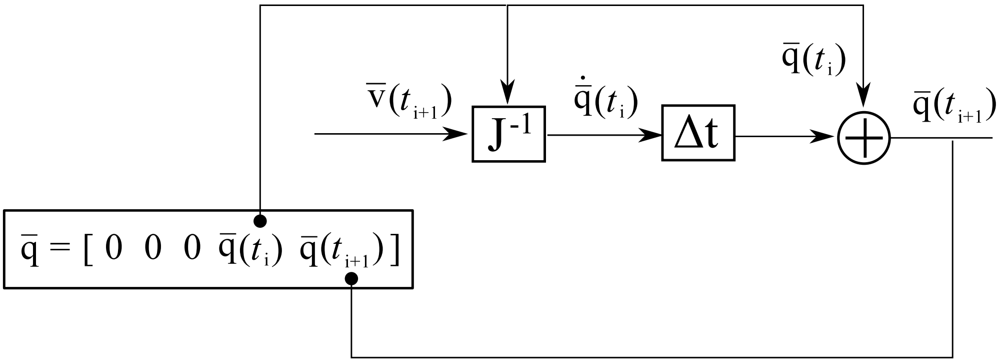


Figure 9: Inverse differential kinematics diagram with integration method.

## 2.6 DYNAMICS

This section deals with the study of the forces required to cause the motion. In the field of robotics, two main approaches are available: the Euler-Newton and the Euler-Lagrange [13, 16]. They both lead to the same results, but they are indeed different, both conceptually and computationally.

### 2.6.1 Euler-Lagrange method

Euler-Lagrange method is an energy based approach. With this technique, the equations of motion can be obtained in a systematic way independently of the reference frame. By choosing a set of generalized coordinates describing the link positions (the $\mathbf{q} = [q_1 \ldots q_n]$ are the natural choice), it is possible to define the Lagrangian of the structure:

$$\mathcal{L} = \mathcal{T} - \mathcal{U} \tag{26}$$

Where $\mathcal{T}$ and $\mathcal{U}$ are the kinetic and potential energy. Lagrange equations is given by:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}}\right) - \left(\frac{\partial \mathcal{L}}{\partial \mathbf{q}}\right) = \tau \tag{27}$$

Where $\tau$ are the generalized forces, or non-conservative forces acting on the links: they are mainly given by the actuator torques and the joint friction torques. From this equation it is possible to examine the relationship between the joint positions and the generalized forces.

However, although the formulation is fairly easy to understand, its implementation is actually troublesome. The equations of the kinetic and the potential energy are, in fact:

$$\mathcal{T} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij}(\mathbf{q})\dot{q}_i\dot{q}_j = \frac{1}{2}\dot{\mathbf{q}}^{\mathbf{T}}\mathbf{B}(\mathbf{q})\dot{\mathbf{q}} \tag{28}$$

Where $\mathbf{B}(\mathbf{q})$ represents:

$$\mathbf{B}(\mathbf{q}) = \sum_{i=1}^{n}(m_{li}\mathbf{J}_{Pi}{}^{T}\mathbf{J}_{Pi} + \mathbf{J}_{Oi}{}^{T}\mathbf{R}_i\mathbf{I}_{li}\mathbf{R}_i{}^{T}\mathbf{J}_{O_i} + m_{m_i}\mathbf{J}_{Pm}{}^{T}\mathbf{J}_{Pm} + \mathbf{J}_{Omi}{}^{T}\mathbf{R}_{mi}\mathbf{I}_{mi}\mathbf{R}_{mi}{}^{T}\mathbf{J}_{Omi}) \tag{29}$$

And, for the potential energy:

$$\mathcal{U} = \sum_{i=1}^{n}(\mathcal{U}_{l_i} + \mathcal{U}_{m_i}) = -\sum_{i=1}^{n}(m_{l_i}\mathbf{g}_0{}^{T}\mathbf{p}_{l_i} + m_{m_i}\mathbf{g}_0{}^{T}\mathbf{p}_{m_i}) \tag{30}$$

These equations do not have an easy solution: Eq. 28, for example, is highly non linear, and the $\mathbf{B}(\mathbf{q})$ matrix is made up of several nested components that are not well suited for a quick, recursive approach. Moreover, the presence of partial derivatives and the fact that we need to deal with symbolic quantities complicates the problem.

Thus, even tough this approach is good for having a sense of the physics involved in the problem, it does not appear to be a viable method for a real time code simulation.

### 2.6.2 Euler-Newton method

Euler-Newton approach is based on the balance of all the forces and torques acting on the generic link of the manipulator. The solution of this problem is well suited for a recursive approach, thus making this our choice for the dynamics simulation. The approach starts from the classic Newton's formula:

$$F = m\,\dot{v}_C \tag{31}$$

From solid mechanics, we can recall that the moment acting on a rotating body of inertia ${}^{C}I$ is given by:

$$N = {}^{C}I\dot{\omega} + \omega \times {}^{C}I\omega \tag{32}$$

Where $\omega$ is the angular velocity and $\dot{\omega}$ is the angular acceleration. By knowing the trajectory to be followed, we then know the position, velocity and acceleration of the joints (that is, $\mathbf{q}$, $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$). With these information and with data about the mass distribution of each joint (mass and inertia tensor), we can calculate the joint torques required at each link. This approach is much more computationally-friendly, and its equations are suited for a simple recursive technique. Thus, in this thesis we analyze dynamics with the Euler-Newton method.

### 2.6.3 The Euler-Newton routine

Equations can be implemented by following Luh-Walker method, developed in 1980 [17]. It is made up of two parts: the *outward* and the *inward* iteration. The first part consists on the calculation of $\omega$, $\dot{\omega}$, $\dot{\mathbf{v}}$ $\dot{\mathbf{v}}_{cm}$ for all the links of structure. These computations are propagated from *link 1* to *link N* of the chain, hence the name *outward*.

*Outward part*

Recalling the expression introduced in the kinematics section, the propagation of the rotational velocity is obtained as:

$${}^{i+1}w_{i+1} = {}^{i+1}_{i}R\,{}^{i}w_i + \dot{\theta}_{i+1}\,{}^{i+1}\hat{k}_{i+1} \tag{33}$$

Derivation of rotational velocity implies the derivation of the trasformation matrix as well. We get:

$$ {}^{i+1}\dot{w}_{i+1} = {}^{i+1}_{i}R\,{}^{i}\dot{w}_i + {}^{i+1}_{i}R\,{}^{i}w_i \times \dot{\theta}_{i+1} + \ddot{\theta}_{i+1}\,{}^{i+1}\hat{k}_{i+1} \tag{34} $$

The linear acceleration is:

$$ {}^{i+1}\dot{v}_{i+1} = {}^{i+1}_{i}R\,[\,{}^{i}\dot{v}_i + {}^{i}\dot{\omega}_i \times {}^{i}P_{i+1} + {}^{i}\omega_i \times ({}^{i}\omega_i \times {}^{i}P_{i+1})\,] \tag{35} $$

The linear acceleration of center of mass of *link i+1*, expressed in *Frame i+1*, is expressed by:

$$ {}^{i}\dot{v}_{C_i} = {}^{i}\dot{v}_i + {}^{i}\dot{\omega}_i \times {}^{i}P_{C_i} + {}^{i}\omega_i \times ({}^{i}\omega_i \times {}^{i}P_{C_i}) \tag{36} $$

Note that ${}^{i}P_{C_i}$ represents the distance from the $i$ joint to the center of mass of *link i*. As far as concerns the forces and torques acting on the link, we can apply Eq. 31 and 32:

$$ F_i = m\,\dot{v}_{C_i} \tag{37} $$

$$ N_i = {}^{C}_{i}I\dot{\omega}_i + \omega_i \times {}^{C}_{i}I\omega i \tag{38} $$

Summing up, the outward part of the solution process is then constituted by solution of the following set of equations, starting from *i=0* and arriving to *i=N-1*:

$$ \left\{ \begin{array}{rcl} {}^{i+1}\dot{w}_{i+1} & = & {}^{i+1}_{i}R\,{}^{i}\dot{w}_i + {}^{i+1}_{i}R\,{}^{i}w_i \times \dot{\theta}_{i+1} + \ddot{\theta}_{i+1}\,{}^{i+1}\hat{k}_{i+1} \\ {}^{i+1}\dot{v}_{i+1} & = & {}^{i+1}_{i}R\,[\,{}^{i}\dot{v}_i + {}^{i}\dot{\omega}_i \times {}^{i}P_{i+1} + {}^{i}\omega_i \times ({}^{i}\omega_i \times {}^{i}P_{i+1})\,] \\ {}^{i}\dot{v}_{C_i} & = & {}^{i}\dot{v}_i + {}^{i}\dot{\omega}_i \times {}^{i}P_{C_i} + {}^{i}\omega_i \times ({}^{i}\omega_i \times {}^{i}P_{C_i}) \\ F_i & = & m\,\dot{v}_{C_i} \\ N_i & = & {}^{C}_{i}I\,\dot{\omega}_i + \omega_i \times {}^{C}_{i}I\,\omega_i \end{array} \right. \tag{39} $$

*Inward part*

The second part comprises the use of Newton Euler equations (Eq. 31 and 32) to obtain the inertial forces and torques acting on the links' centers of mass. Then, referring to the free body diagram of Fig. 10, the force and

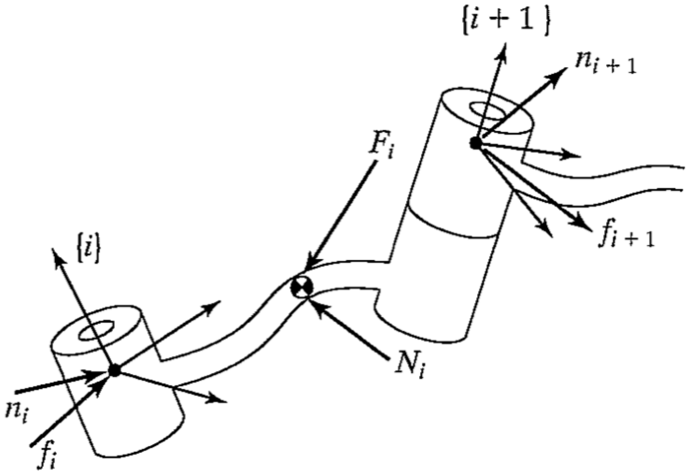


Figure 10: Free body diagram of *link i*, with force balance

moment balance equations need to be considered in order to extract the joint torques.

Every link experiences inertial force and torque in addition to forces and torques exerted on it by the adjoining links. From the free body diagram, the force and torque equilibrium yield the following balance equations:

$$ {}^{i}F_i = {}^{i}f_i - {}^{i}_{i+1}R^{i+1}f_{i+1} \tag{40} $$

$$ {}^{i}N_i = {}^{i}n_i - {}^{i}n_{i+1} + (-{}^{i}P_{C_i}) \times {}^{i}f_i - ({}^{i}P_{i+1} - {}^{i}P_i) \times {}^{i}f_{i+1} \tag{41} $$

Where the following notation was used:

- $f_i$ is the force exerted by *link i-1* on *link i*
- $n_i$ is the torque exerted by *link i-1* on *link i*

Equation 41 can be rearranged with the aid of rotational matrices and the results from Eq. 40:

$$ {}^{i}N_i = {}^{i}n_i - {}^{i}_{i+1}R^{i+1}n_{i+1} - {}^{i}P_{C_i} \times {}^{i}F_i - {}^{i}P_{i+1} \times {}^{i}_{i+1}R^{i+1}f_{i+1} \tag{42} $$

Reordering Eq. 40 and Eq. 42, we can finally obtain the iterative expressions needed. This time the index count for the solution will start from N and decreases to 1.

$$\begin{cases} {}^if_i &= {}^iF_i + {}^i_{i+1}R^{i+1}f_{i+1} \\ {}^in_i &= {}^iN_i + {}^i_{i+1}R^{i+1}n_{i+1} + {}^iP_{C_i} \times {}^iF_i + {}^iP_{i+1} \times {}^i_{i+1}R^{i+1}f_{i+1} \\ \tau_i &= {}^in_i^T \hat{Z}_i \end{cases} \tag{43}$$

Since the calculations start from the end effector and end at the first link, this second part of the routine is called *inward*.

*Initial conditions*

For both inward and outward iterations, we need some starting conditions. Referring to equations block 39, the computation process starts for *i=0*. This means that some of the parameters need to be known: $\omega_0$, $\dot{\omega}_0$, $\dot{\mathbf{v}}_0$. These have to be set in this fashion:

$$\omega_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad \dot{\omega}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad \dot{\mathbf{v}}_0 = k \cdot \begin{bmatrix} 0 \\ 0 \\ \mathbf{g} \end{bmatrix} \tag{44}$$

Where $k$ is zero if gravity is not considered, 1 if it is considered. Obviously, if the base is connected to ground, $\omega_0$ and $\dot{\omega}_0$ will be zero.
The initial conditions concerning equations block 43 are related to the dynamic effects present at the end effector; these effects can be due to impacts/contacts or to the presence of a tool or a load (i.e. industrial manipulators). In this case, we suppose these components to be zero, that is, we suppose an unloaded robot, subjected only to its dynamics with no external contributions (apart from gravity). Thus, the initial conditions are:

$${}^7f_7 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad {}^7n_7 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{45}$$

This algorithm is at the core of the robotic manipulator control and has been thoroughly described and numerically verified in a previous work by this author [10]. For details on the simulations please refer to [10, 11, 14].

3

# PRELIMINARY DESIGN

## 3.1 MASTER THESIS RESULTS SUMMARY

A preliminary analysis of the design of the platform was performed by the author in the Master Thesis "Design of a Robotic Arm for Laboratory Simulations of Spacecraft Proximity Navigation and Docking" [Antonello, 2013].

In this work, an iterative approach was presented to solve the sizing issue. It was found that several combinations of the parameters satisfy the requirements.

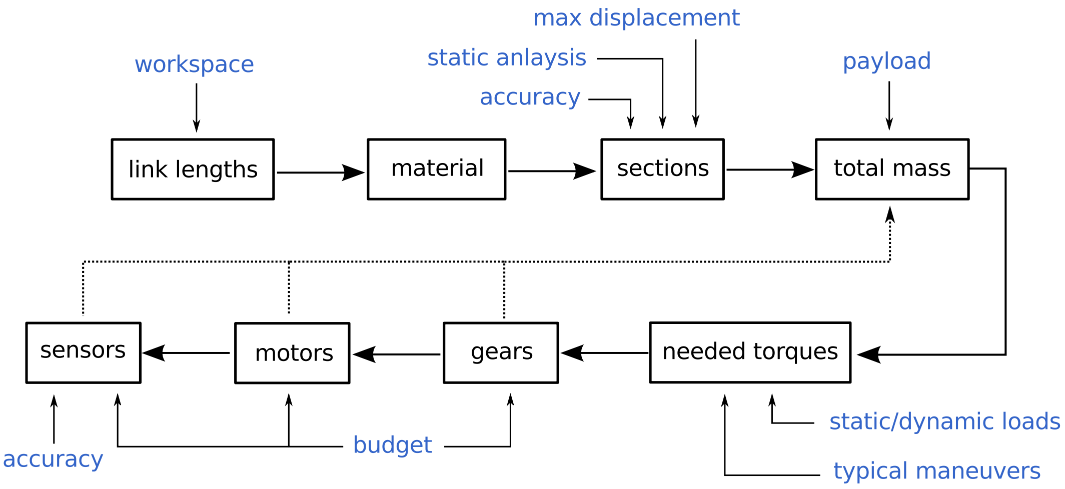


Figure 11: Block diagram representing the iterative sizing process.

It is possible to draw a generic block diagram, presented in Fig. 11, that represents the logic process on which the sizing procedure is based. In the picture, the blue writings represents the requirements, and are needed for the calculation of the different steps, whereas the dotted lines are the parameters that need to be assumed and then back-checked in a *trial-error* type of procedure.

### 3.1.1 Original requirements and constraints

The preliminary choices for the robot sizing were based on the required workspace of at least 0.5 × 0.5 × 0.5 m. The following lengths for the first three links were chosen:

$$l_1 = 0.7 \ m \qquad l_2 = 0.7 \ m \qquad l_3 = 0.6 \ m \tag{46}$$

The material choice is fairly straightforward: what is needed is a material with a high resistance-to-weight (RtoW) ratio. Weight saving is a must in order to limit the size and cost of the motors. Extruded aluminium profiles proved to be the most favorable solution [10]: they presents a good RtoW ratio, they are very easy to machine and there are a lot of section choices which are relatively cheap due to the simple process (extrusion) used in their production.

In the second iteration of this design process, a new set of leghts for the links was defined:

$$l_1 = 1 \ m \qquad l_2 = 0.7 \ m \qquad l_3 = 0.7 \ m \tag{47}$$

This slight modification is due to a backiteration rising from the new requirements on the end-effector payload, which has now been set at 2.5 kg. With this data, the chosen motors allowed for a slight increase in the lengths of the links. In particular, $link1$ was increased of 40% due to the foreseen docking and OOS applications; this, in turn, resulted in a newly designed base joint for the satisfaction of the more demanding force and torque constraints.

The availability of a rough general design from [10] allowed to perform fine tuning iterations in the choice of the link extruded sections and the mechanical components. The motors and gearings previously selected have hence been accurately verified under the light of this fine tuning.

## 3.2 MOTOR SELECTION

The key parameters to seek for when choosing a motor are:

- high *power-to-weight* ratio
- low mass and inertia
- high rotational speed
- low backlash (e.g. high precision)
- low torque ripple
- *(if available)* accurate built-in sensors

The most common DC electric actuators can be further divided into two classes: permanent-magnet DC servomotors and brushless DC servomotors [18].

In the brushless type, the rotor (made of ferromagnetic material) generates the magnetic flux, whereas the fixed external armature (stator) has the windings. The commutation is provided by a position sensor placed on the shaft, which generates the feed sequence for the windings.

It is clear that in the latter case, because of the absence of physical contact between the rotor and the stator, the performances are definitely superior. First of all, with no contact, the mechanical losses due to friction are minimized. The elimination of brushes eliminates also the electric loss due to voltage drops at the contact of brushes and plates. Moreover, with no contact there is also less material wear, and the motor life is increased.

### 3.2.1 Motors: torque requirements

The motor choice main driver is the torque required in the **worst operational case**. Three contributions need to be taken into account:

1. the torques needed for supporting the weight of the structure
2. the torques needed for withstanding the dynamic effects
3. the overall efficiency

For this application, the first contribution accounts for most of the torque that needs to be provided. Due to the low velocity of this application, the added dynamic effects due to the trajectory tracking will be taken into account in the form of a correction factor chosen accordingly to the typical torques calculated in the dynamics simulations. Since this parameter is evidently dependent from the simulated trajectory under analysis, it is not possible to design an architecture that will satisfy all the conditions beforehand. On the contrary, the correction factor will define an operative range in terms of payload loading and acceleration of the end effector, which will have to be respected every time a simulation scenario is designed by the control block.

We start by analyzing the **static** situation: the motor torques depend on the configuration, that is, on the values of the generalized coordinates $\bar{\mathbf{q}} = [q_1 \dots q_n]$. However, it is immediate to note that $\theta_1$ does not have any relevance in changing the configurations loads. For the same reason, $\theta_6$ has no influence neither.

The remaining coordinates $\theta_2$, $\theta_3$, $\theta_4$ and $\theta_5$ affect the torques needed for static equilibrium. The situation can be further simplified by noting that the end effector can be considered as a single body, thus arriving at the conclusion that only $\theta_2$ and $\theta_3$ are playing a significant role in the variation of the static torque. (With this simplification, the static torque analysis for *joint 1* and *6* is momentarily ignored).

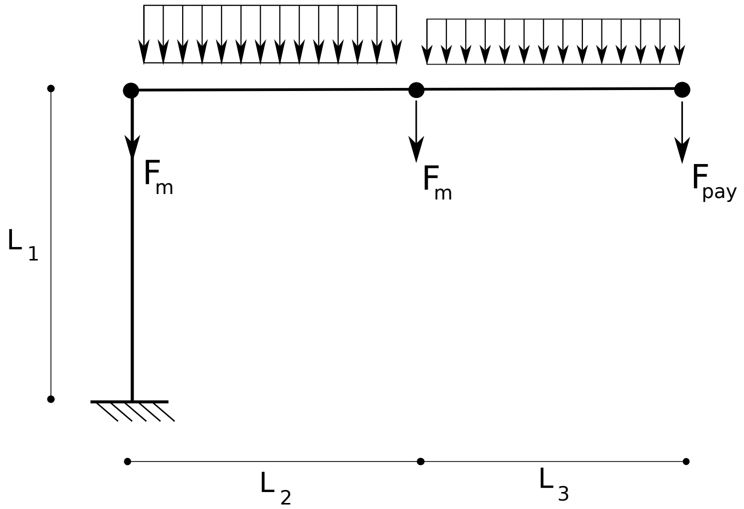


Figure 12: Simplified model of the robot structure.

### 3.2.2 Load analysis

When working with slender bodies, it is key to worry about their rigidity in order to avoid bending and buckling [19]. In this case, due to the way loads are applied, it is unlikely for buckling to occur for *link 2* and *link 3*, whereas bending could be a serious issue. Moreover, a flimsy structure might cause vibrations and disturbances. *Link 1*, on the other hand, could be subjected to both buckling and bending.

To avoid these phenomena, proper sections need to be chosen. We first analyze a simplified 2D structure with the arm fully extended. The diagram in Fig. 12 schematizes the problem.

In the figure, $F_M$ is the weight of the motor and $F_{pay}$ is the weight of the payload. The latter accounts not only for the object attached at the tip of the end effector, but also for the weight of the last three links. The distributed loads represent the weight of the links.

This structure can be analyzed analytically in order to obtain the moment, shear and normal force distribution along the links. A qualitative

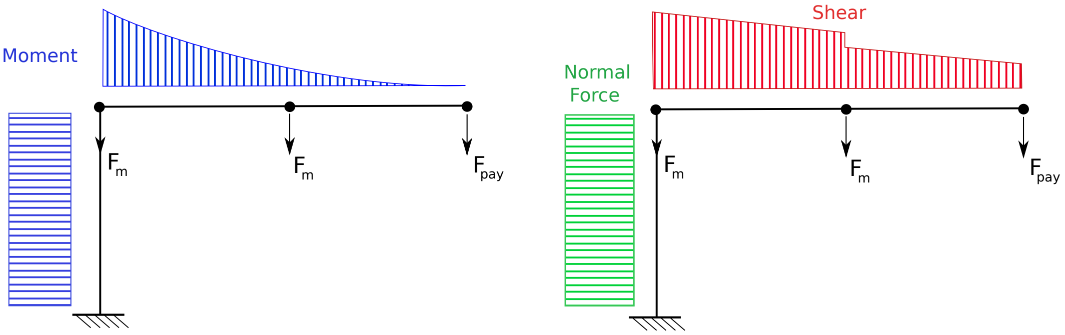


Figure 13: Moment, shear and normal force behavior of robot's simplified structure.

plot of the general behavior is presented in Fig. 13.

It is clear that *link 1* is subjected to the highest load, in terms of moment (which is constant along its span) and normal force. The normal force acting on this first link may create buckling: this situation has been already analyzed in [10] and the selected link proved to be safe from buckling phenomena.

Moving on towards the end effector, we can see that *link 2* is subjected to a bending moment that has the highest value at *joint 2* and decreases in a parabolic fashion until the end effector. The shear force acting on *link 2* is linear.

The presence of a concentrated force at *joint 3*, induced by the weight of the motor, changes the slope of the moment profile, which keeps decreasing till zero at the end effector. On the other hand, there is a discontinuity of the shear profile due to the concentrated load $F_m$ (note that this offset is equal to the value of the force), which decreases linearly from *joint 2* to the end effector, where its value is $F_{pay}$.

The sizing of the links, once the length is known, starts from the displacement analysis. It is possible to set a requirement on the maximum vertical displacement in the worst case configuration: this happens when the manipulator is fully stretched (Fig. 12).

With a simple analytical procedure, this approach led to the calculation of the minimum moment of inertia $I_x$ required for the links' sections in [10]. The commercial link choice is then quite straightforward. The iterative process for the selection of the links can be summarized as follows:

1. The motor and payload weights are accounted with a safety factor of 1.2
2. A maximum tip displacement of 4 mm is set
3. The displacement analysis is executed: this yields the product $E \cdot I$
4. Knowing the material properties (and $E$), the value of $I$ is obtained
5. A profile having this $I$ and the previously estimated linear weight is searched among the commercially available sections
6. If commercial profiles present higher weight for that $I$, the estimated weight has to be increased. Analysis is executed again with these modifications.
7. If commercial profiles present lower weight for that $I$, the estimated weight has to be decreased. Analysis is executed again with these modifications.
8. If there exists a commercial profile with the parameters used, then the problem is solved and the procedure ends.

After the definition of the first-try parameters, we need to calculate the displacement. From beam theory [20], the formula relating the moment and the curvature induced on a beam is:

$$\begin{cases} M_x = -EI_{xy}u'' - EI_{xx}v'' \\ M_y = -EI_{yy}u'' - EI_{xy}v'' \end{cases} \tag{48}$$

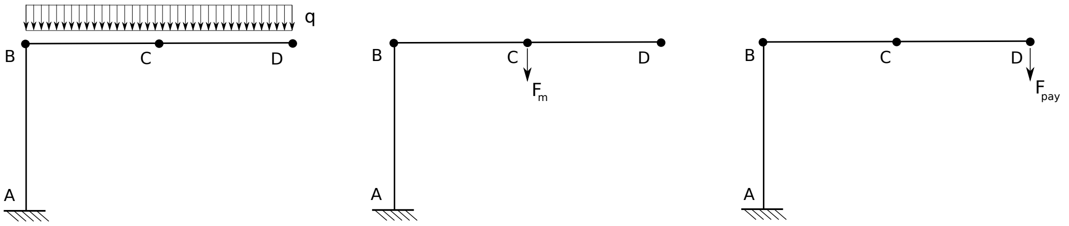


Figure 14: Load decomposition for structural analysis.

Since we assume to work with symmetric profiles, $I_{xy} = 0$. Moreover, in the 2D model, $M_y = 0$. We are left with the formula for the vertical displacement:

$$v''(x) = -\frac{M_x(x)}{EI_{xx}} \tag{49}$$

Integrating twice this formula, it is possible to obtain the displacement of the beam as a function of $x$. The main problem is to obtain the $M_x(x)$ function. This can be easily accomplished remembering that we are analyzing a linear elastic problem, and the superposition of effects is valid.

Therefore, the problem in Fig. 12 can be decomposed in three parts (we initially assume the distributed weight to be constant, that is, *link 2* and *link 3* to have the same profile): we obtain the three cases presented in Fig. 14. The computation of the moments derives from static equilibrium, and yields, for the three cases (referring to Fig. 12, we set the $x$ axis as starting from *joint 2* and going right-wise, and $L = L_2 + L_3$):

$$\begin{cases} M_1(x) = \frac{q}{2}(L-x)^2 & \text{if } 0 \leq x \leq L, \\ M_2(x) = F_m(L-x) & \text{if } 0 \leq x \leq L_2, \\ M_3(x) = F_{pay}(L-x) & \text{if } 0 \leq x \leq L \end{cases} \tag{50}$$

The moment diagrams are presented in Fig. 15. From these expressions, we can obtain the corresponding displacements [20]:

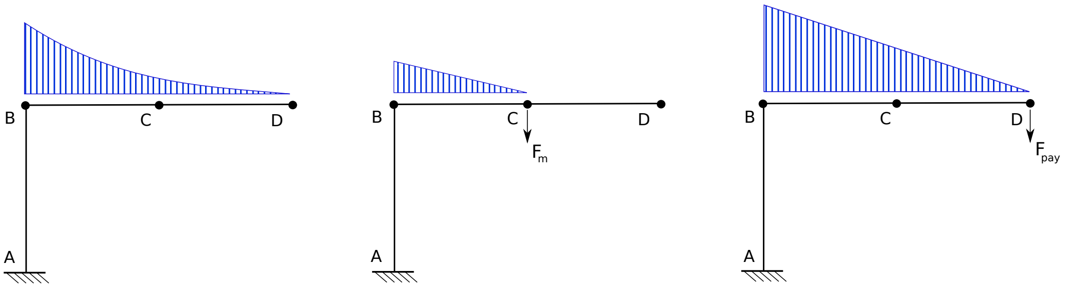


Figure 15: Moment diagrams for the three decomposed cases.

**Case 1**

$$EIv_1'' = -M_1(x) \tag{51a}$$

$$EIv_1'' = -\frac{q}{2}(L-x)^2 \tag{51b}$$

$$EIv_1' = -\frac{q}{2}\left[L^2x + \frac{x^3}{3} - Lx^2\right] + C_1 \tag{51c}$$

$$EIv_1 = -\frac{q}{2}\left[L^2\frac{x^2}{2} + \frac{x^4}{12} - L\frac{x^3}{3}\right] + C_1x + C_2 \tag{51d}$$

From which, using the assumption of a fixed constraint at *joint 2*, $v'(x=0)=0$ and $v'(x=0)=0$. Thus, $C_1 = C_2 = 0$. The final expression for $v_1(x)$ is:

$$v_1(x) = -\frac{q}{24\,EI}\left[6L^2x^2 + x^4 - 4Lx^3\right] \tag{52}$$

**Case 2**

$$EIv_2'' = -M_2(x) \tag{53a}$$

$$EIv_2'' = -F_m(L_2 - x) \tag{53b}$$

$$EIv_2' = -F_m\left(L_2x - \frac{x^2}{2}\right) + C_1 \tag{53c}$$

$$EIv_2 = -F_m\left(L_2\frac{x^2}{2} - \frac{x^3}{6}\right) + C_1x + C_2 \tag{53d}$$

From which, using the assumption of a fixed constraint at joint 2, $v'(x=0)=0$ and $v'(x=0)=0$. Thus, $C_1 = C_2 = 0$. The final expression for $v_2(x)$ is:

$$v_2(0 \leq x \leq L_2) = -\frac{F_m}{6\,EI}\left[3L_2x^2 - z^3\right] \tag{54}$$

Since the load is effective till $x = L_2$, the slope of the curvature past $L_2$ will remain constant, and the deformed curve will be a segment. Since we can calculate the value of the displacement and its derivative in $L_2$, the line equation[1] yields the formula for $v_2(x)$ when $L_2 \leq x \leq L$:

$$v_2(L_2 \leq x \leq L) = -\frac{F_m}{6\,EI}\left[3L_2^2 x - L_1^3\right] \tag{55}$$

**Case 3**

$$EIv_3'' = -M_3(x) \tag{56a}$$

$$EIv_3'' = -F_{pay}(L - x) \tag{56b}$$

$$EIv_3' = -F_{pay}\left(Lx - \frac{x^2}{2}\right) + C_1 \tag{56c}$$

$$EIv_3 = -F_{pay}\left(L\frac{x^2}{2} - \frac{x^3}{6}\right) + C_1 x + C_2 \tag{56d}$$

From which, using the assumption of a fixed constraint at *joint 2*, $v'(x = 0) = 0$ and $v'(x = 0) = 0$. Thus, $C_1 = C_2 = 0$. The final expression for $v_2(x)$ is:

$$v_3(x) = -\frac{F_{pay}}{6\,EI}\left[3Lx^2 - z^3\right] \tag{57}$$

Putting together the three case, we can obtain the equation describing the total displacement as the sum of $v_1$, $v_2$, $v_3$:

$$v(x) = \begin{cases} \frac{1}{6\,EI}\{q[6L^2x^2 + x^4 - 4Lx^3] + \\ -4[F_m(3L_2x^2 - z^3) + F_{pay}(3Lx^2 - z^3)]\} & \text{if } 0 \leq x \leq L_2 \\ \\ \frac{1}{6\,EI}\{q[6L^2x^2 + x^4 - 4Lx^3] + \\ -4[F_m(3L_2^2x - L_1^3) + F_{pay}(3Lx^2 - z^3)]\} & \text{if } L_2 \leq x \leq L \end{cases} \tag{58}$$

[1] We can write the line equation as $v_2(x) = v_2'(L_2)(x - L_2) + v_2(L_2)$, where $v_2(L_2) = -\frac{F_m}{3EI}L_1^3$ and $v_2'(L_2) = -\frac{F_m}{2EI}L_1^2$.

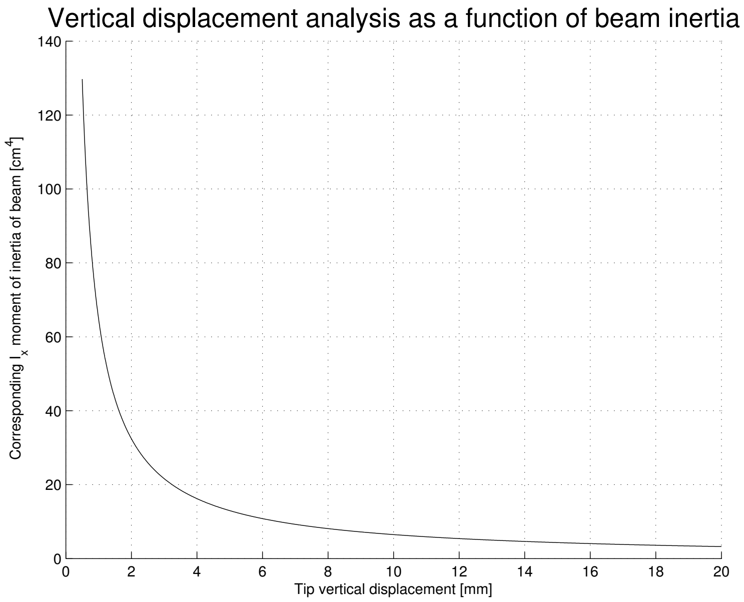


Figure 16: Vertical displacement versus $I_x$ value.

Finally, the tip displacement can be obtained by evaluating Eq. 58 for $x = L$. This yields:

$$v_{tip} = -\frac{qL^4}{8EI} - \frac{F_m L_1^2(3L - L_1) + 3F_{pay}L^3}{6EI} \tag{59}$$

The requirement set for the displacement is $v_{tip}$. Since we are looking for the value of $I$, Eq. 59 can be rewritten extracting $EI$:

$$EI = -\frac{1}{v_{tip}}[3qL^4 + 4F_m L_1^2(3L - L_1) + 12F_{pay}L^3] \tag{60}$$

The parameter $E$, Young's modulus, is material dependent and is known: for aluminium, $E = 70$ GPa. We are left with $I$, which can be easily computed. The relationship between the required $v_{tip}$ and the minimum $I$ needed can be expressed with a plot (Fig. 16). It can be seen that the smaller the requirement for the displacement becomes, the steeper the rise in $I$ is.

The procedure now is straightforward: with the requirements and the given values for the loads, $I$ is computed. From commercial available profiles, the sections with a similar $I$ are pinpointed, and the specific mass for unit length is compared to the one imposed at the beginning ($q$).

If these values are close to each other, in a $\pm 10\%$ range, the assumptions were good and the sections are readily available. If, on the other hand, this is not happening, we need to re-iterate the process. Two parameters can be changed: the requirement on the displacement and/or the weight for unit length $q$. If, for example, the computed $I$ is typical of profiles with higher mass per unit length, the simulation will be repeated increasing the presumed $q$. This tuning will finally provide a compatible solution. In our case, the sizing parameters were chosen as follows:

$$F_m = 30 \tag{61a}$$

$$F_{pay} = 50\,kg \tag{61b}$$

$$E = 70\,GPa \tag{61c}$$

$$q = 2\ kg/m \tag{61d}$$

$$v_{tip} = 4\,mm \tag{61e}$$

With these data, Eq. 60 yields the value for the $x$ moment of inertia:

$$I_x = 11.4\ cm^4$$

From commercial catalogues, we can find some typical extruded aluminium profiles: for profiles with a linear weight similar to the one we chose, the moment of inertia abundantly satisfies the computed value. In this case, selection started by satisfying the minimal requirements presented in the previous page. This resulted in the selection of the 30$x$60 Bosch Rexroth profile, which has the following specifications:

$$I_x = 19.6\ cm^4 \tag{62a}$$

$$I_y = 5.1\ cm^4 \tag{62b}$$

$$q = 1.5\ kg/m \tag{62c}$$

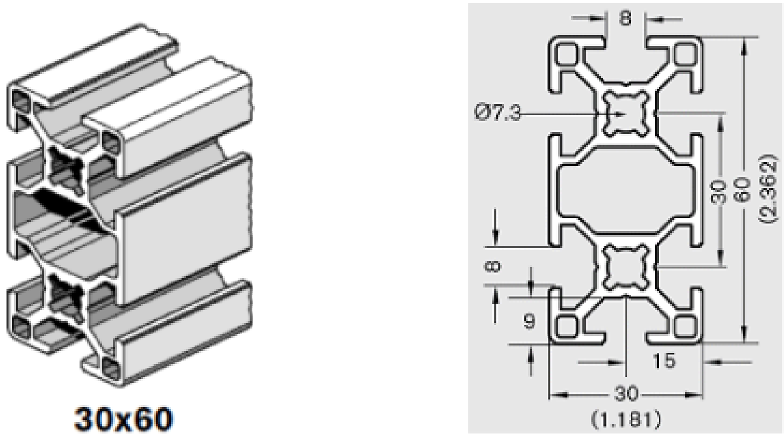


Figure 17: Bosch rexroth 30x60 extruded profile

Once this section was selected, we had to perform an additional iteration in order to update the parameters of the simulations: weight requirement will decrease from 2 *kg/m* to ∼1.5 *kg/m*. Since we have the moment of inertia of the section, we solve equation Eq. 59 for the maximum displacement. This yields:

$$v_{tip}^{max} = 3.63\, mm \tag{63}$$

Which satisfies the requirement set at the beginning. With this initial draft of the robot structure, we proceed to the following stage of the design, introduced in the following pages, in which we add detail to the mechanical model and we perform a differentiated selection of the links for each of the arm sections.

## 3.3 stiffness-based positioning precision map

The static analysis presented in the previous section provided an engineering model to define the section properties for the links. The problem can be restated if we assume that the geometrical properties are known: it is interesting to note, in fact, that once the design is completed, it is possible

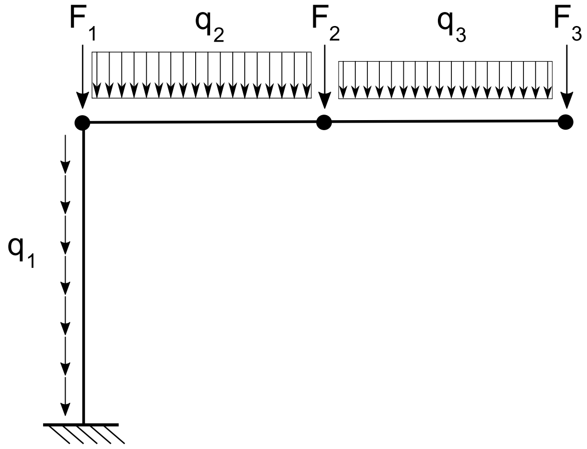


Figure 18: Complete diagram of efforts acting on the manipulator.

to create a 3D map of the effects of the assembly's flexibility in terms of end-effector displacement.

In applications in which high positioning precision is of outmost importance, a model that takes into account the intrinsic mechanical limitation due to the non-infinite rigidity could be applied to the control system to adaptively correct the commanding angles of the joints. In order to design such a model, we generalize the problem by parametrizing the characteristics of the system.

In addition, we now take into account the first link flexibility and the horizontal displacement it generates, which was ignored in the first level computations of the previous section.

Supposing that the last three links can be accounted for as a concentrated force, we obtain the diagram presented in Fig. 18.

Once the moment diagram is computed in the general case, we can use beam deflection theory to obtain the displacements. Since we suppose to have symmetric sections:

$$v''(x) = -\frac{M_z(x)}{EI_{xx}} \tag{64}$$

In order to find the displacement at the end effector, we need to extract the

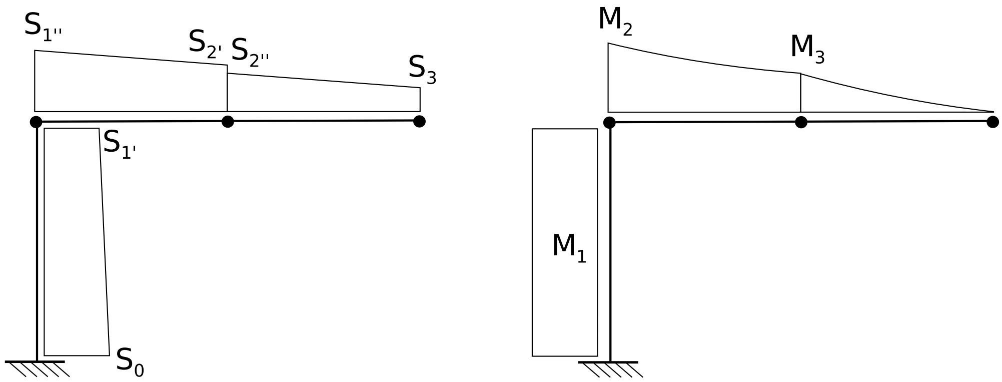


Figure 19: Total diagrams for a) normal force, shear and b) bending moment.

position and attitude in an sequential fashion, and use the superposition of effects to compute the end result. The iterative algorithm can be described as follows (with $x$ being the coordinate along the direction of the beam):

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_{i+1} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_{i} + R \cdot \begin{bmatrix} l_{i+1} \\ \iint \frac{M_z(x)}{EI_{xx}} \\ \int \frac{M_z(x)}{EI_{xx}} \end{bmatrix}_{i+1} \tag{65}$$

Where the rotation matrix R ensures the correctness of the sum in the case of non parallel links.

In this case, the moments to which the links are subject, expressed in the local reference frame of the $i - th$ link, are:

$$M_1(x) = M_1 = F_2\, l_2 + F_3\,(l_2 + l_3) + q_2 \cdot \frac{l_2^2}{2} + q_3\, l_3 \cdot \left(l_2 + \frac{l_3}{2}\right) \tag{66}$$

$$\begin{aligned} M_2(x) = q_2 \frac{l_2^2}{2} - q_2\, l_2\, x + q_2\, \frac{x^2}{2} + q_3\, l_2\, l_3 + q_3 \frac{l_3^2}{2} - q_3\, l_3\, x \\ + (F_2 + F_3)\, l_2 + F_3\, l_3 - (F_2 + F_3) x \end{aligned} \tag{67}$$

$$M_3(x) = q_3 \frac{(l_2 + l_3)^2}{2} + q_3 \frac{x_2^2}{2} - q_3(l_2 + l_3)x + F_3(l_2 + l_3) - F_3 x \tag{68}$$

The angular deflection and $y$-displacement caused by these bending moments are:

$$v_1'(x_1) = \frac{1}{EI_1} \cdot M_1 x_1 \tag{69}$$

$$\begin{aligned} v_2'(x_2) = & \frac{1}{EI_2}\left(q_2 \frac{l_2^2}{2} - q_2\, l_2\, \frac{x^2}{2} + q_2\, \frac{x^3}{6} + q_3\, l_2\, l_3 + q_3 \frac{l_3^2}{2} - q_3\, l_3\, \frac{x^2}{2}\right. \\ & \left. + (F_2 + F_3)\, l_2 + F_3\, l_3 - (F_2 + F_3) \frac{x^2}{2}\right) \end{aligned} \tag{70}$$

$$v_3'(x_3) = \frac{1}{EI_2}\left(q_2 \frac{l_3^2}{2} x + \frac{x^3}{3} - 2\frac{x^2 l_3}{2} + F_3\, l_3 x - F_3 \frac{x^2}{2}\right) \tag{71}$$

$$v_1(x_1) = \frac{1}{2\, EI_1} \cdot M_1 x_1^2 \tag{72}$$

$$\begin{aligned} v_2(x_2) = & \frac{1}{EI_2}\left(q_2 \frac{l_2^2}{2} - q_2\, l_2\, \frac{x^3}{6} + q_2\, \frac{x^4}{12} + q_3\, l_2\, l_3 + q_3 \frac{l_3^2}{2} - q_3\, l_3\, \frac{x^3}{6}\right. \\ & \left. + (F_2 + F_3)\, l_2 + F_3\, l_3 - (F_2 + F_3) \frac{x^2}{2}\right) \end{aligned} \tag{73}$$

$$v_3(x_3) = \frac{1}{EI_2}\left(q_2 \frac{l_3^2}{4} x^2 + \frac{x^4}{12} - \frac{x^3 l_3}{3} + F_3\, l_3 \frac{x^2}{2} - F_3 \frac{x^3}{6}\right) \tag{74}$$

In terms of tip deflection, if we assume the base of the first link to be the origin of a 2D cartesian reference frame, the composite equation to compute it is:

$$v_{ee} = v_1 + v_2 + v_3 \tag{75}$$

$$v_{ee} = \begin{bmatrix} v_1 \\ l_1 \end{bmatrix} + \begin{bmatrix} \cos(v_1') & \sin(v_1') \\ -\sin(v_1') & \cos(v_1') \end{bmatrix} \begin{bmatrix} l_2 \\ v_2 \end{bmatrix} + \begin{bmatrix} \cos(v_2') & \sin(v_2') \\ -\sin(v_2') & \cos(v_2') \end{bmatrix} \begin{bmatrix} l_3 \\ v_3 \end{bmatrix} \tag{76}$$

This approach can be extended to the computation of the end effector displacement for each joint configuration.

The equations need to take into account the general coordinates $\theta_2$ and $\theta_3$ and are independent from $\theta_1$. A way to approach the problem would be to scale the distributed loads and the concentrated forces according to the configuration under analysis, that is:

$$q_2^\dagger = q_2 \cos(\theta_2) \qquad q_3^\dagger = q_3 \cos(\theta_3) \tag{77a}$$

$$F_2^\dagger = F_2 \cos(\theta_2) \qquad F_3^\dagger = F_3 \cos(\theta_3) \tag{77b}$$

Naturally, the rotation matrices will contain also the rotation due to the configuration. This, ultimately, allows to obtain a map of the entire manipulator's workspace which describes the positioning error due to the link flexibility and payload loading.

The complete expressions in this case are the same of Eq. 71, 73, 74, with the only caveat of the modified loading. The visualization of this performance parameter is shown for different joint configurations in Fig. 20, where the norm of the displacement error is represented for the angular set $[\theta_2; \theta_3]$ spanning from $-\pi$ to $+\pi$.

In more detail, the same plot is separated in Fig. 21-25 into the $x, y$ components of the resulting end effector displacement. In these plots, the parameters being used are:

$$E = 70 \cdot 10^9 \, Pa \tag{78a}$$

$$I = 19.60 \cdot 10^{-8} \, m^4 \tag{78b}$$

$$l_1 = 1 \, m \tag{78c}$$

$$l_2 = 0.7 \, m \tag{78d}$$

$$l_3 = 0.7 \, m \tag{78e}$$

With the loads being:

$$F_1 = 100 \, N \tag{79a}$$

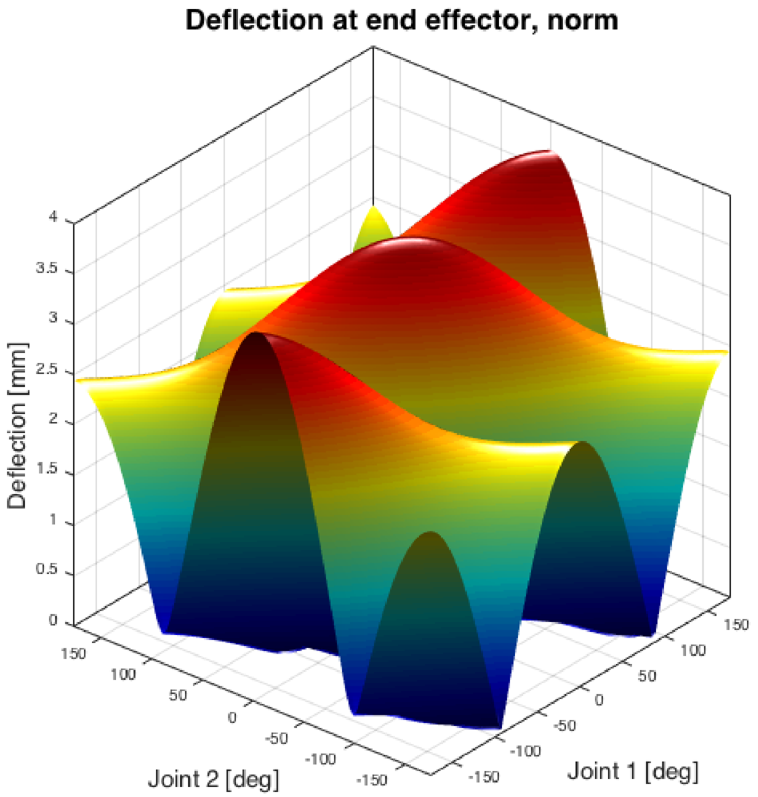


Figure 20: End effector deflection, norm

$$F_2 = 50\,N \tag{79b}$$

$$F_3 = 30\,N \tag{79c}$$

$$q = 1.50\,kg/m \tag{79d}$$

Which are derived from the previous section. From the plot, it is possible to start a refinement process of the current architecture: first of all, it can be seen by comparing the $x$ and $y$ displacement, that most of the displacement happens in the $x$-axis; this can be greatly improved by increasing the stiffness of the first link. Since the weight of *link* 1 will be unloaded by a proper bearing structure, the added weight will not influence the sizing of the motors (apart from the added rotational inertia). To

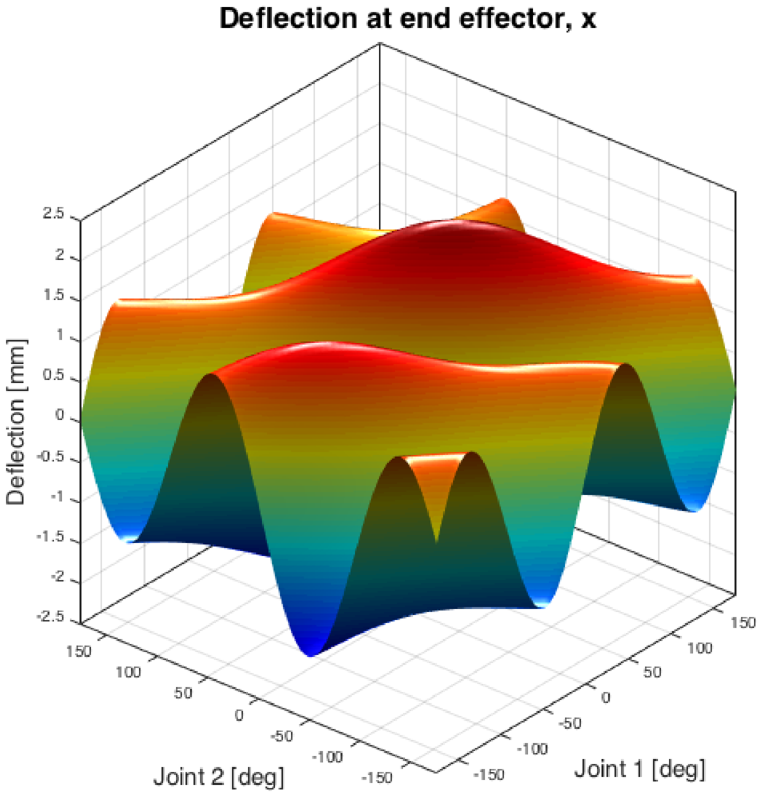


Figure 21: End effector deflection, $x$-component

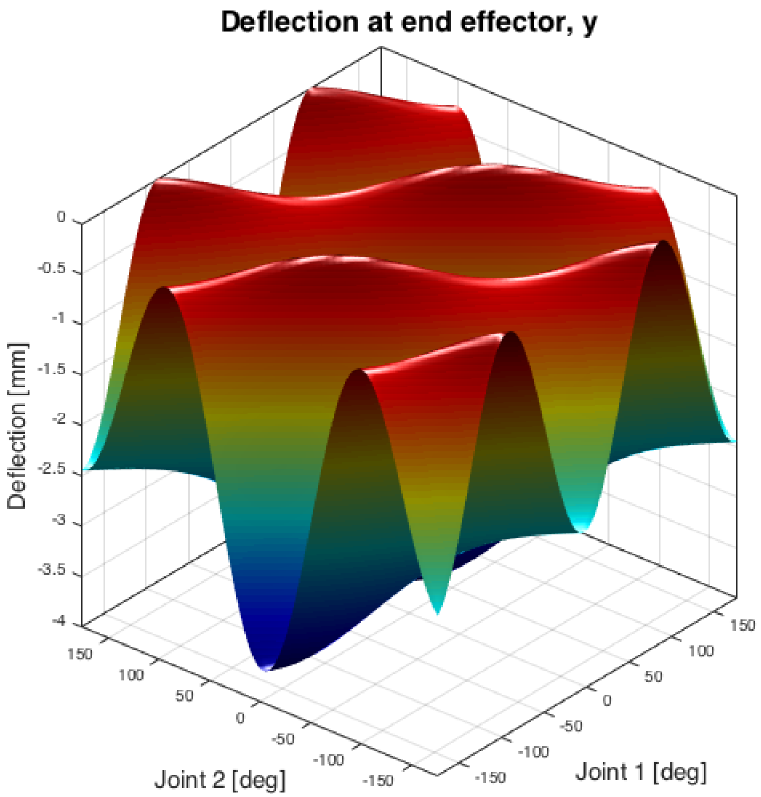


Figure 22: End effector deflection, *y*-component

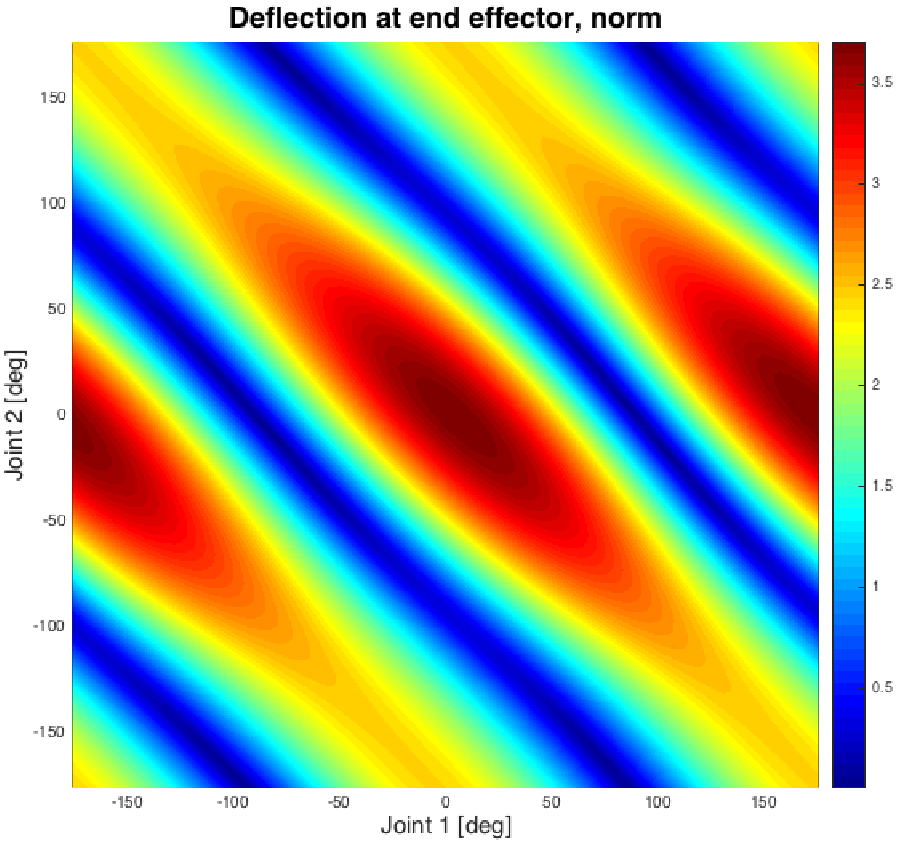


Figure 23: End effector deflection, norm. Contour plot

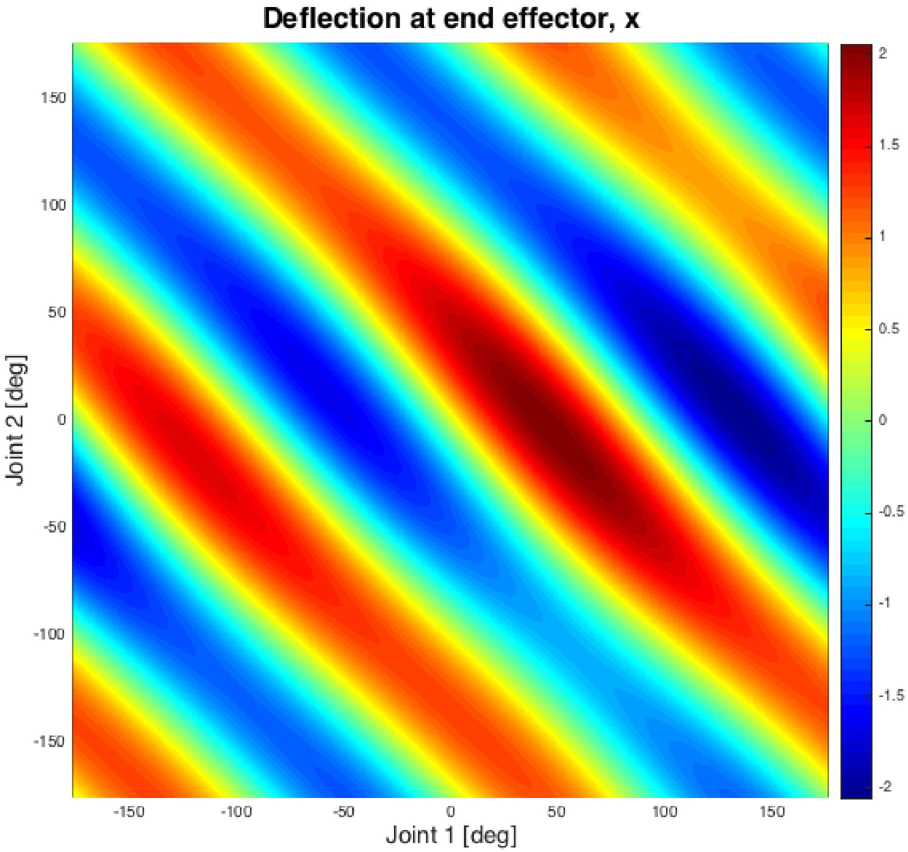


Figure 24: End effector deflection, $x$-component. Contour plot

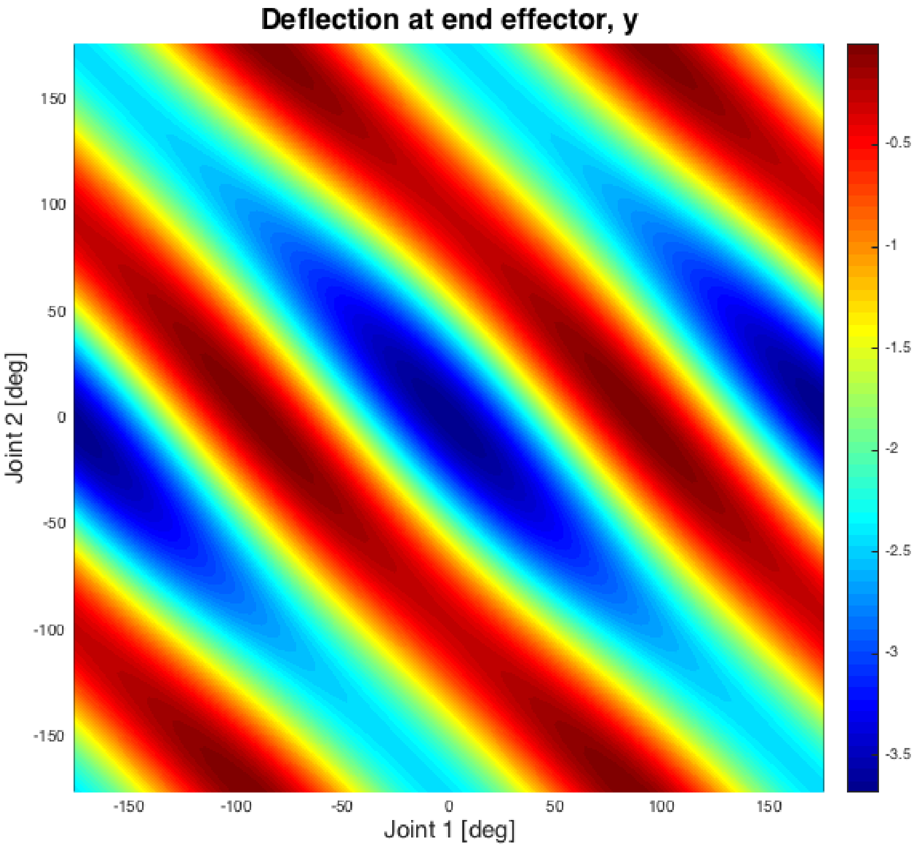


Figure 25: End effector deflection, *y*-component. Contour plot

this extent, we modify the choice of *link* 1 with a more suitable profile: in this case, the Bosch 90x90L was selected:

$$I_x = 211.1\ cm^4 \qquad (was\ \ 19.6\ cm^4) \tag{80a}$$

$$I_y = 211.1\ cm^4 \qquad (was\ \ 5.1\ cm^4) \tag{80b}$$

$$q = 6.5\ kg/m \qquad (was\ \ 1.5\ kg/m) \tag{80c}$$

Then, for the selection of the next two links, we started from the baseline Bosch 30x60 configuration selected in the previous chapter and applied a tapered approach: we increased progressively the second link section until we reached a reasonable tradeoff between the increased stiffness and the increased weight (which will naturally influence the selection of motor 2).

Finally, we selected the Bosch 45x90SL profile, where SL stands for $super - light$. These are the specifications:

$$I_x = 73.40\ cm^4 \qquad (was\ \ 19.6\ cm^4) \tag{81a}$$

$$I_y = 9.1\ cm^4 \qquad (was\ \ 5.1\ cm^4) \tag{81b}$$

$$q = 2.4\ kg/m \qquad (was\ \ 1.5\ kg/m) \tag{81c}$$

The last link is kept unchanged and is a Bosch 30x60 profile. Before proceeding with FEM analysis, the new design was tested for tip displacement. The results are shown in Fig. 27-32: as expected, this new configuration drastically reduced the norm of the displacement, with a mean reduction of 54%. The highest displacement, happening when the arm is fully extended, is reduced to:

$$v_{tip}^{max} = 1.41\, mm \tag{82}$$

The data used in the plots is summarized in the following:

$$E = 70 \cdot 10^9\ Pa \tag{83a}$$

$$I_1 = 211.1 \cdot 10^{-8}\, m^4 \tag{83b}$$

$$I_2 = 73.40 \cdot 10^{-8}\, m^4 \tag{83c}$$

$$I_3 = 19.60 \cdot 10^{-8}\, m^4 \tag{83d}$$

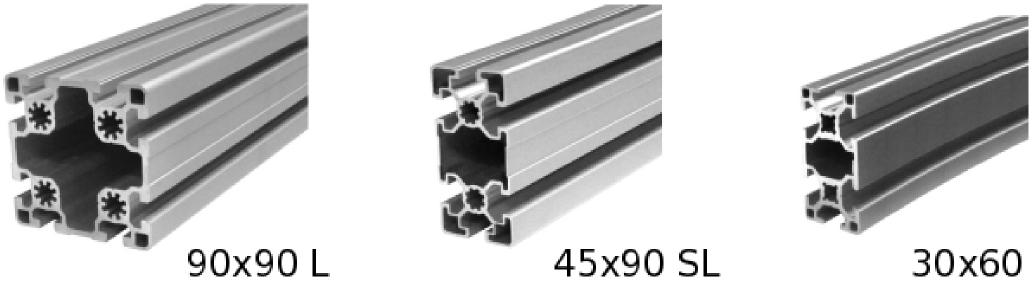


Figure 26: Bosch profiles chosen for link 1, 2 and 3 respectively.

$$l_1 = 1\,m \tag{83e}$$

$$l_2 = 0.7\,m \tag{83f}$$

$$l_3 = 0.7\,m \tag{83g}$$

With the updated loads being:

$$F_1 = 100\,N \tag{84a}$$

$$F_2 = 50\,N \tag{84b}$$

$$F_3 = 30\,N \tag{84c}$$

$$q_1 = 6.5\,kg/m \tag{84d}$$

$$q_2 = 2.4\,kg/m \tag{84e}$$

$$q_3 = 1.50\,kg/m \tag{84f}$$

Reassuming the results obtained in this section, the links' sizing parameters are:

| | length $[m]$ | area $[cm^2]$ | $I_x[cm^4]$ | $I_y[cm^4]$ | q $[kg/m]$ | mass $[kg]$ |
|---|---|---|---|---|---|---|
| *link 1* | 1 | 24.1 | 211.1 | 211.1 | 6.5 | 6.5 |
| *link 2* | 0.7 | 9.04 | 73.4 | 18.1 | 2.44 | 1.71 |
| *link 3* | 0.7 | 5.5 | 19.6 | 5.1 | 1.5 | 1.05 |

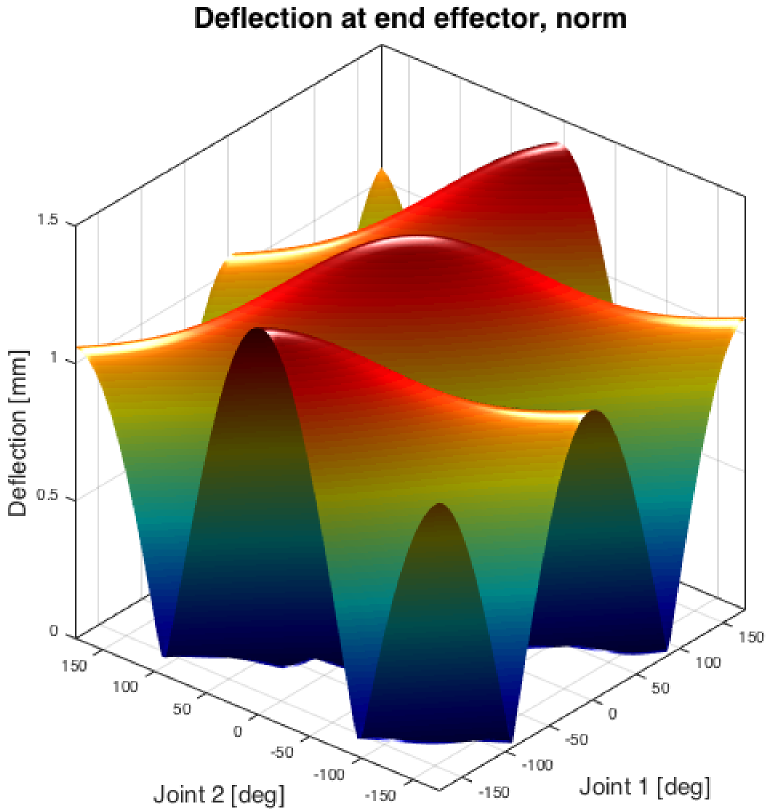


Figure 27: End effector deflection, modified designed, norm.

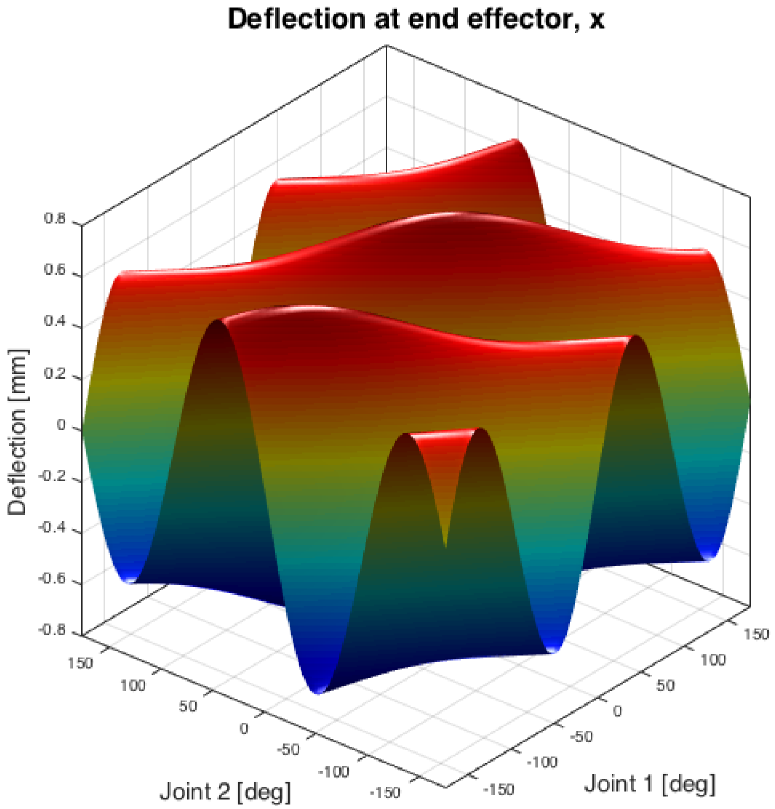


Figure 28: End effector deflection, modified designed, $x$-component.

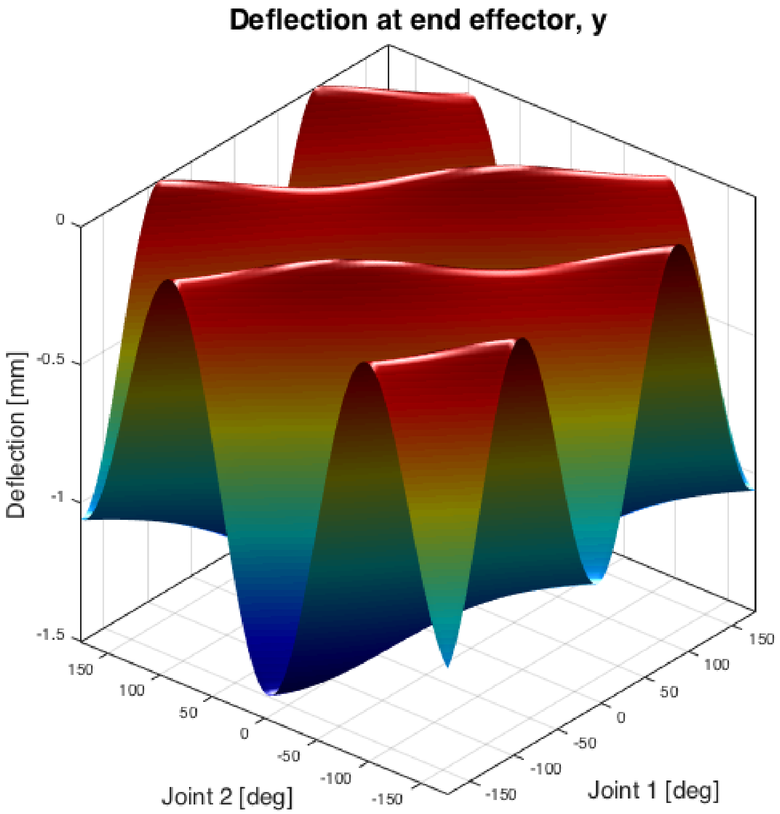


Figure 29: End effector deflection, modified designed, $y$-component.

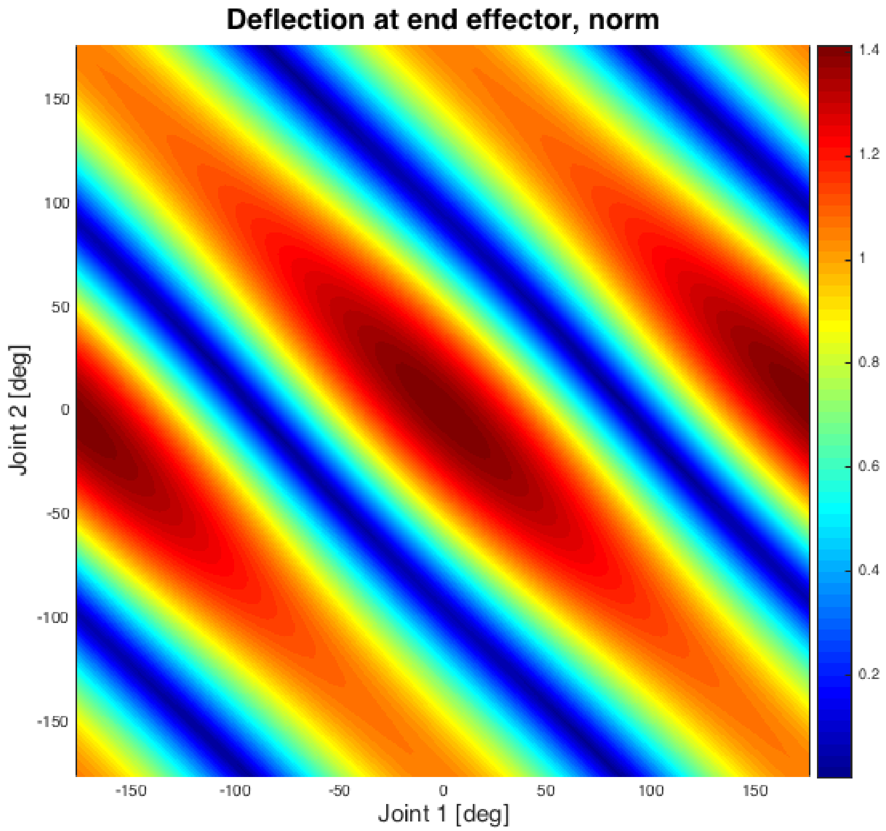


Figure 30: End effector deflection, modified designed, norm. Contour plot

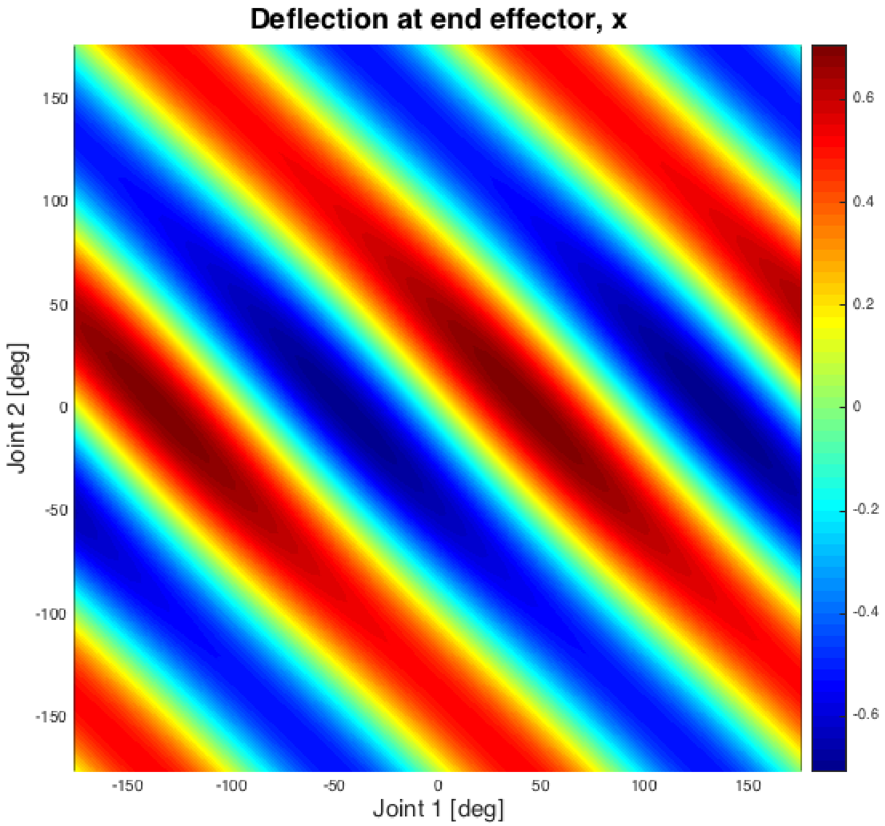


Figure 31: End effector deflection, modified designed, $x$-component. Contour plot

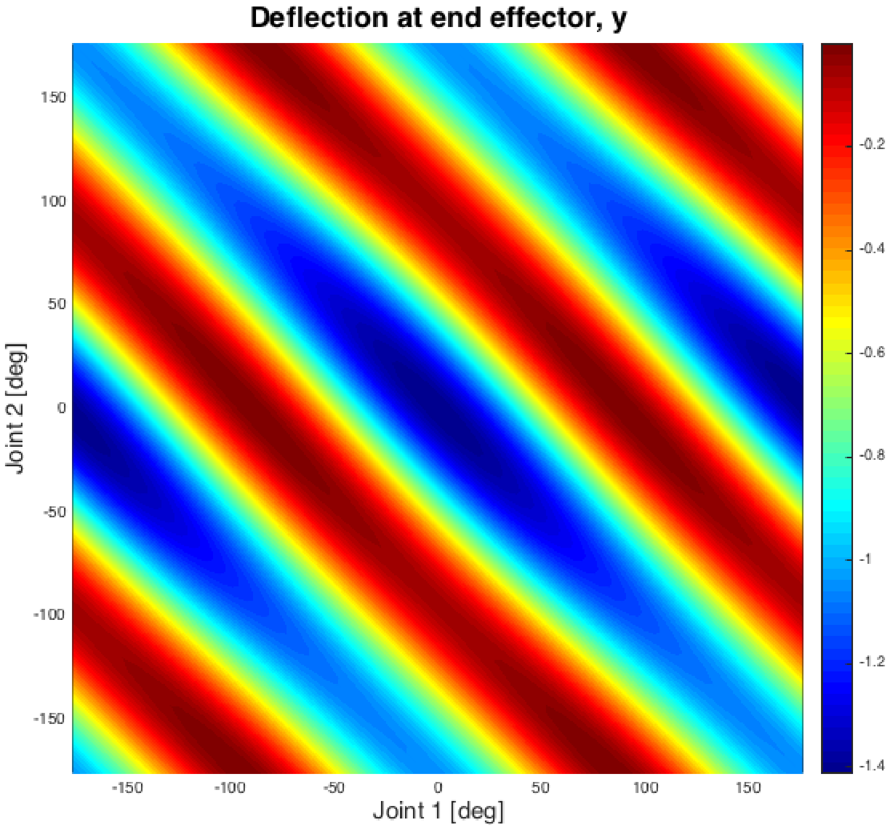


Figure 32: End effector deflection, modified designed, $y$-component. Contour plot

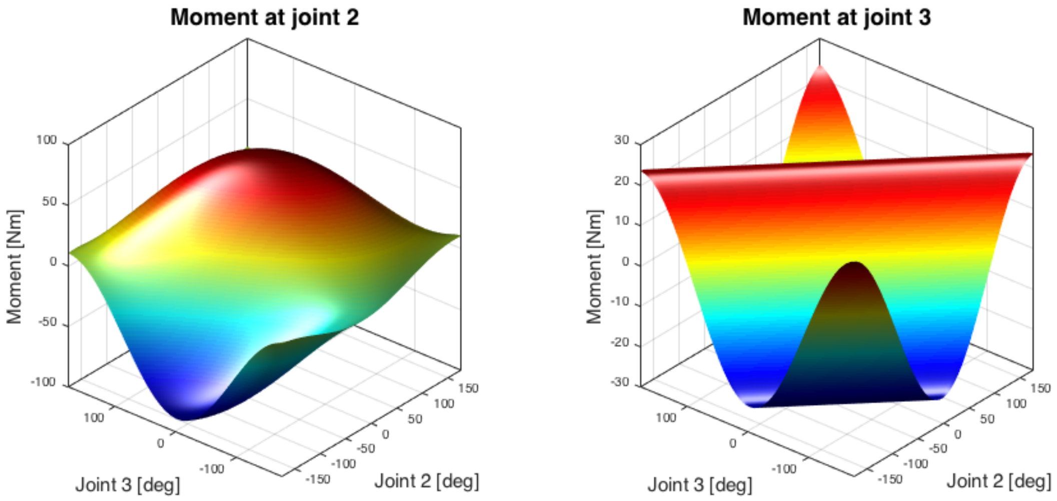


Figure 33: Torque profiles for different arm configurations, 3D plot.

## 3.4 MOTORS AND GEARS SELECTION

Using the results obtained in the previous paragraph, it is now possible to plot the torque required at *joints* 2 and 3 for the chosen links for each of the arm configurations in the workspace. Fig. 33 and 34 allow the determination of the required motor torques, by taking into account the efficiency and the correction factor to account for the dynamic loading during typical trajectory simulations.

When the arm is fully stretched, there is a maximum of the torque to be provided at *joint* 2. The torques required in the dynamic case [10] are in the +10÷12% range of the static torque. A good design strategy shall be to multiply the maximum values obtained from Fig. 33, 34 by a dynamic correction coefficient of 1.2. Moreover, in addition to the dynamic effects, we need to take into account also friction and all the additional hardware and harness weight needed in the installation.

An additional safety factor (SF) of 1.1 can be reasonably introduced.

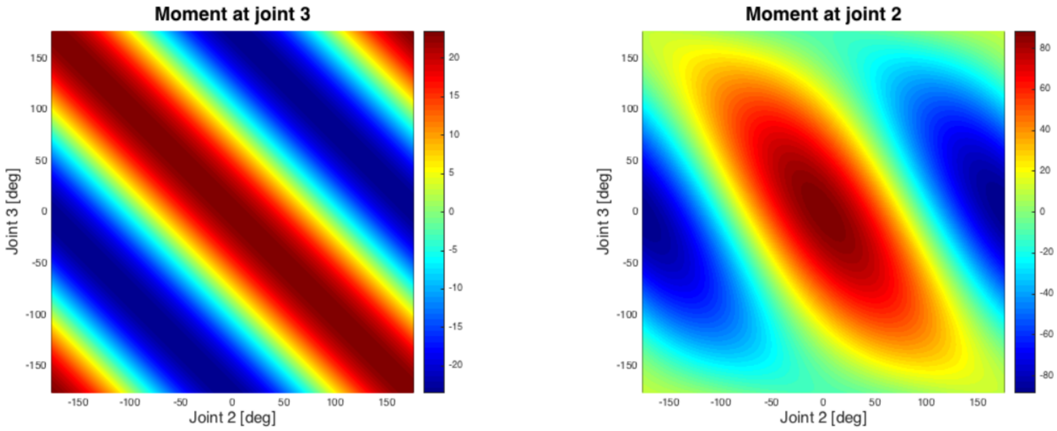


Figure 34: Torque profiles for different arm configurations, 2D plot.

Besides, this allows to still have some authority margin when the critical conditions are reached.

As far as *joint 1* is concerned, it can be seen that, ideally, there are no torque requirements for static equilibrium: since the axis of rotation is parallel to the gravity vector, the vectorial moment acting on *joint 1* due to gravity can not be handled by *joint 1*. Consequently, an appropriate bearing attached to *joint 1* will be needed in order to handle this load.

In this case, the only torque contribution for the sizing of motor 1 is given by inertial and dynamic loads, which can be inferred from [10]. A sizing torque of 10 Nm seems to be a reasonable choice. We again multiply this value by the usual safety factor SF.

The last three joints are clearly not subjected to high torque values. *Joint 4* experiences the maximum static torque in the case in which *link 3* is parallel to the $\mathbf{z_0}$ direction and *link 4* is perpendicular to *link 3*. Maximum torque for *joint 5* happens when *link 5* and its axis of revolution lie on the ground plane.

*Joint 6* maximum torque takes into account only dynamic related torques: as long as *link 6* and the objects connected to it are axial-symmetric, there are no acting torques for static equilibrium.

From [10] we can see that for the fastest trajectory simulated, a torque value of $7 \cdot 10^{-6}$ Nm is needed. However, we are in the approximation of an axialsymmetric body connected to the shaft whose axis of giration is coincident with the shafts axis. Since other bodies, non necessarily axialsymmetric, might be attached to it for testing, and due to possible misalignments between the axis, the torque required could be bigger. In order to stay away from saturation, we can think of increasing the requirements: a commercial motor in the 0.1÷0.2 Nm range appears to be more than sufficient to withstand misalignments and (limited) extra weight.

The following table summarizes the maximum torque required by each motor (all values are in $Nm$):

| | *Joint 1* | *Joint 2* | *Joint 3* | *Joint 4* | *Joint 5* | *Joint 6* |
|---|---|---|---|---|---|---|
| Static torque | / | 85.9 | 20.3 | 0.8 | 0.7 | / |
| Dynamic correction | 10 | 103.1 | 24.4 | 1.02 | 0.84 | 0.2 |
| SF correction | 12 | 113.9 | 26.8 | 1.12 | 0.92 | 0.22 |

There are no problems in finding a motor that satisfies the torque requirement for *joint 6*. Problems arise when looking at the other joints. The maximum output torque for an average commercial motor is usually on the order of ∼1 Nm: a transmission gear is obviously needed.

This kind of device allows to up-size/down-size a motor in terms of torque; the consequence is a proportional change in the revolution speed, according to the following laws:

$$\dot{\theta}_{out} = \frac{1}{\eta} \cdot \dot{\theta}_{in} \tag{85}$$

$$\tau_{out} = \eta \cdot \tau_{in} \tag{86}$$

From which:

$$\dot{\theta}_{in} \cdot \tau_{out} = \dot{\theta}_{out} \cdot \tau_{in} \tag{87}$$

Where $\eta$ is the gear ratio. However, this is an ideal law: in the real case mechanical losses are present. Clearly, a good performing transmission, in terms of efficiency, is what we are looking for. The reduction ratio should be as high as possible, as long as the minimum rotational speed requirement is satisfied.

Moreover, a high reduction rate means usually a longer series of reduction stages, which decrease the efficiency and increase the backlash. Among the cohort of available gear drives, planetary gears represent a good solution for this application. Although their performances dwarf if compared to harmonic drives, their quality-to-price ratio is extremely high, and the backlash is still very limited.

For this application, the motor choice was MAXON®, since a factory integrated gearing can be requested. In order to satisfy the requirements, the following pages describe the parts that were selected for the simulator. As far as the motors are concerned:

| *Joint* | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| *Motor model* | **EC 45** | **EC 60** | **EC 90** | **EC 45** | **EC 45** | **EC 32** |
| Nominal voltage [V] | 24 | 48 | 48 | 12 | 12 | 12 |
| Nominal current [A] | 3.21 | 5.94 | 2.27 | 2.02 | 2.02 | 1 |
| Nominal speed [rpm] | 4860 | 2670 | 1610 | 2940 | 2940 | 2790 |
| Nominal torque [mNm] | 128 | 859 | 533 | 55 | 55 | 25.1 |
| Power [W] | 70 | 400 | 90 | 30 | 30 | 15 |
| Max efficiency [%] | 85 | 88 | 85 | 76 | 76 | 73 |
| Weight [g] | 141 | 2400 | 600 | 75 | 75 | 46 |
| Model number | 397172 | 167131 | 244879 | 200142 | 200142 | 339268 |

Table 2: Selected MAXON® motors for the manipulator.

As far as the gearings are concerned, the selected parts are listed in Tab. 3.

| *Joint* | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| *Gear model* | **GP 42C** | **GP 81A** | **GP 52C** | **GS 45A** | **GS 45A** | **GS 38A** |
| Reduction factor | 126:1 | 308:1 | 113:1 | 47:1 | 47:1 | 30:1 |
| Stages | 3 | 3 | 3 | 3 | 3 | 3 |
| Max efficiency [%] | 72 | 70 | 68 | 76 | 76 | 73 |
| Weight [g] | 460 | 3700 | 770 | 224 | 224 | 60 |
| Model number | 203127 | 110413 | 223095 | 301171 | 301171 | 110454 |

Table 3: Selected MAXON® gearings for the manipulator.

## 3.5 CONTROL ELECTRONICS AND SENSORS

The sensors for the motor control are constituted by in-built encoders, which have the following characteristics:

| *Joint* | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| *Encoder model* | **M 256** | **HEDL 9140** | **M 512** | **M 256** | **M 256** |
| Counts per turn | 2048 | 500 | 6400 | 2048 | 2048 |
| Frequency [kHz] | 500 | 100 | 500 | 500 | 500 |
| Channels | 2 | 3 | 2 | 2 | 2 |
| Model number | 462005 | 137959 | 411966 | 462005 | 462005 |

Table 4: Selected MAXON® encoders for the manipulator.

In order to control the motors, the electronics has been chosen from the MAXON® catalogue, and consists of the following boards, each dedicated to a single motor:

| *Joint* | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| *Board model* | **EP 24.5** | **EP 70.10** | **EP 70.10** | **EP 24.5** | **EP 24.5** | **EP 24.2** |

Table 5: Selected MAXON® controllers for the manipulator.

The final torques, obtained combining the motor and gear assemblies, and taking into account their respective efficiencies, are:

| *Joint* | 1 | 2 | 3 | 4 | 5 | 6 |
|---:|:-:|:-:|:-:|:-:|:-:|:-:|
| Gross torque [Nm] | 16.13 | 264.6 | 60.23 | 2.59 | 2.59 | 0.75 |
| Net torque [Nm] | 11.61 | 162.98 | 34.81 | 1.49 | 1.49 | 0.41 |
| Required torque [Nm] | 12 | 113.9 | 26.8 | 1.12 | 0.92 | 0.22 |
| Compound efficiency [%] | 61 | 62 | 58 | 58 | 58 | 53 |
| Compound weight [g] | 601 | 6100 | 1370 | 299 | 299 | 106 |

Table 6: Final motor performances.

## 3.6 INTERFACE DESIGN

In order to properly transfer the required torque, it is mandatory to design adequate interfaces between the motors and the links. The main problem arising when designing a rotative system is the innate presence of uncertainties in the manufacturing and assembling that might lead to unacceptable tolerances in the shaft connections. If these misalignments are not adequately accounted for, the shaft can experience excessive torques and forces that could eventually damage the system.

A wise design principle when transmitting rotative motion, especially with high torques, is to insert a decoupling element that takes care of the misalignments, being these linear or angular, while maintaining torsional rigidity. Also, the addition of this element allows to unload the shaft of the motor, which is not designed to carry continuos axial/radial loading.

Easy-to-implent solutions are belts and/or pinions design. The main problem that these components bear are the introduction of a) control non linearities and b) mechanical backlash. The addition of a flexible element in the form of a belt, in fact, modifies the dynamic response of the system thus affecting the performances of the control architecture: in addition, the effective contribution of these modifications is difficult to estimate properly.

The preferred solution is to add a flexible coupling: in this way, we can accommodate varying degrees of misalignment and some parallel misalignment. In this application, all the transmissions between motor and link are mediated with a flexible coupling with high torsional rigidity: the only exceptions, due to the low level of torque to which they are subject, are *joints 5* and *6*, which are connected directly. The shafts are then unloaded through the use of two bearings properly spaced: this permits to avoid radial shaft loading and to limit the moment induced by the links. The solutions for *links 2* and *3* are conceptually identical, whereas *links 1* and *4* differ due to the vertical shaft configuration: in these cases, the shaft is tapered in order to sit on the upper bearing and properly unload the axial force and bending moment; the second bearing unloads the radial force.

In Fig. 35-37 the solutions for the horizontal and vertical shafts are illustrated for *joints 1, 2* and *4*.

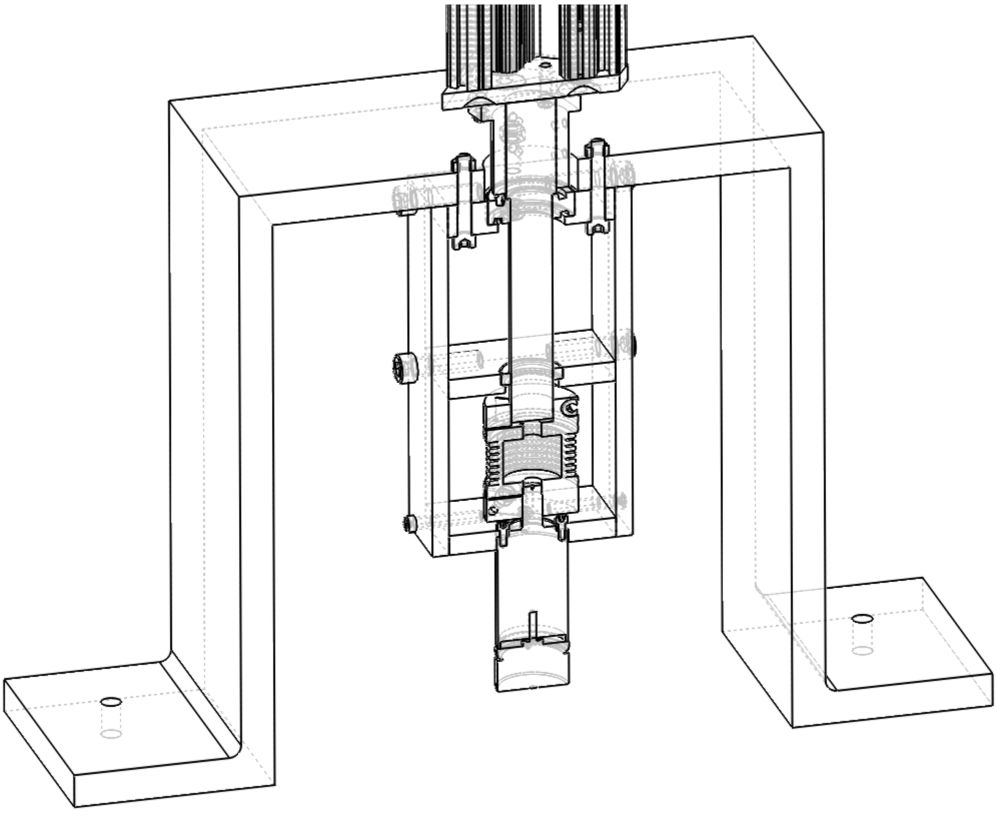


Figure 35: Section view of *joint 1*.

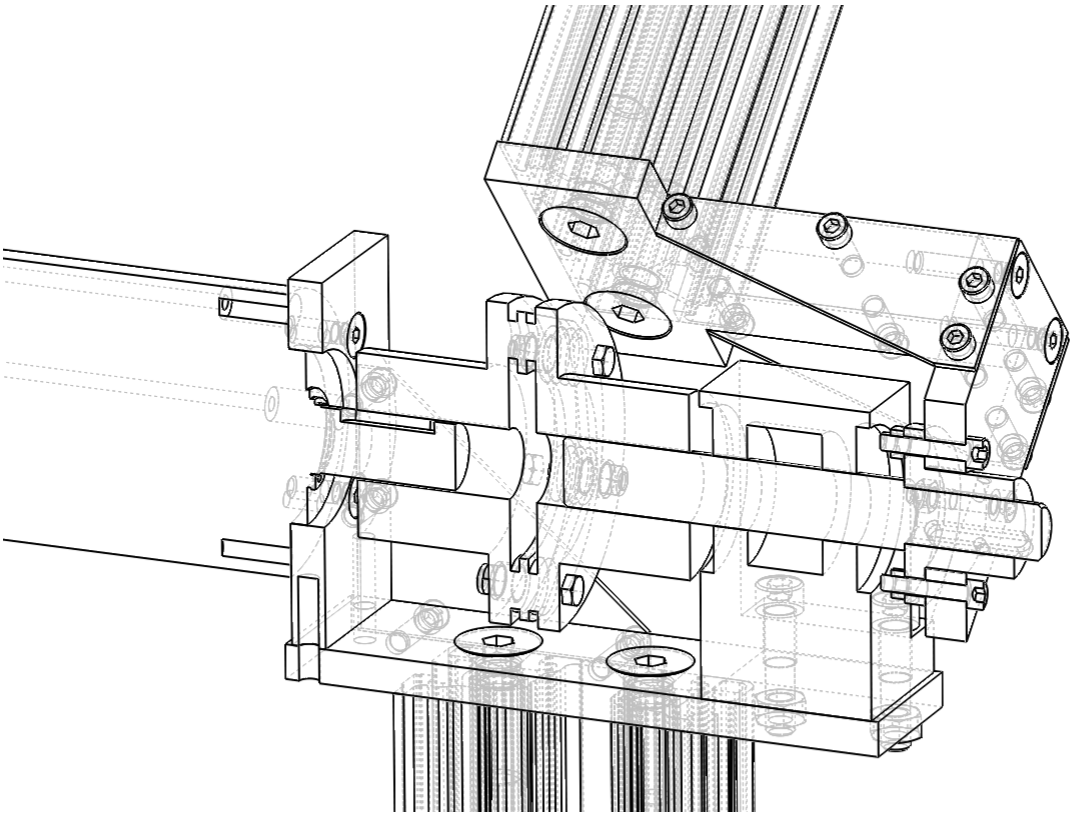


Figure 36: Section view of *joint 2*.

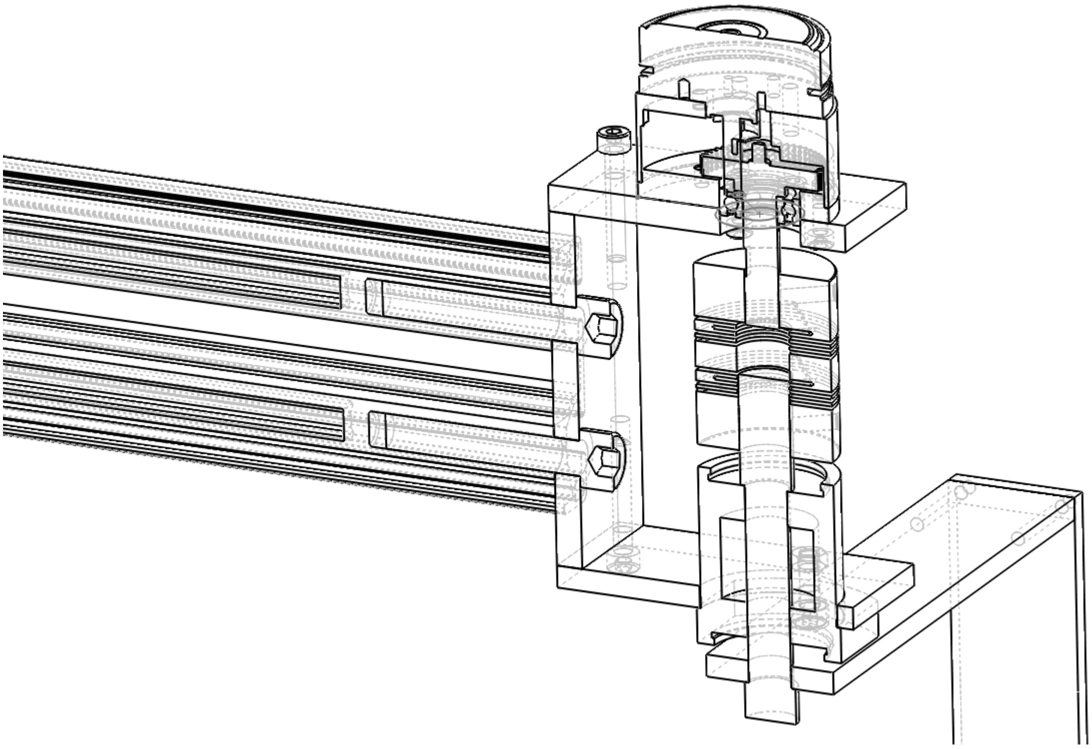


Figure 37: Section view of *joint 4*.

## 3.7 FINAL DESIGN

The following section summarizes the inertial characteristics of the final system. The inertia tensors, due to the relative complexity of the structures, are calculated with the aid of SolidWorks® "*Mass properties*" tool. The frames used in this calculation are centered in the center of rotations defined with Denavit-Hartenberg's notation (Fig. 38-39). A wireframe picture allows for easier identification of the assembly under analysis.

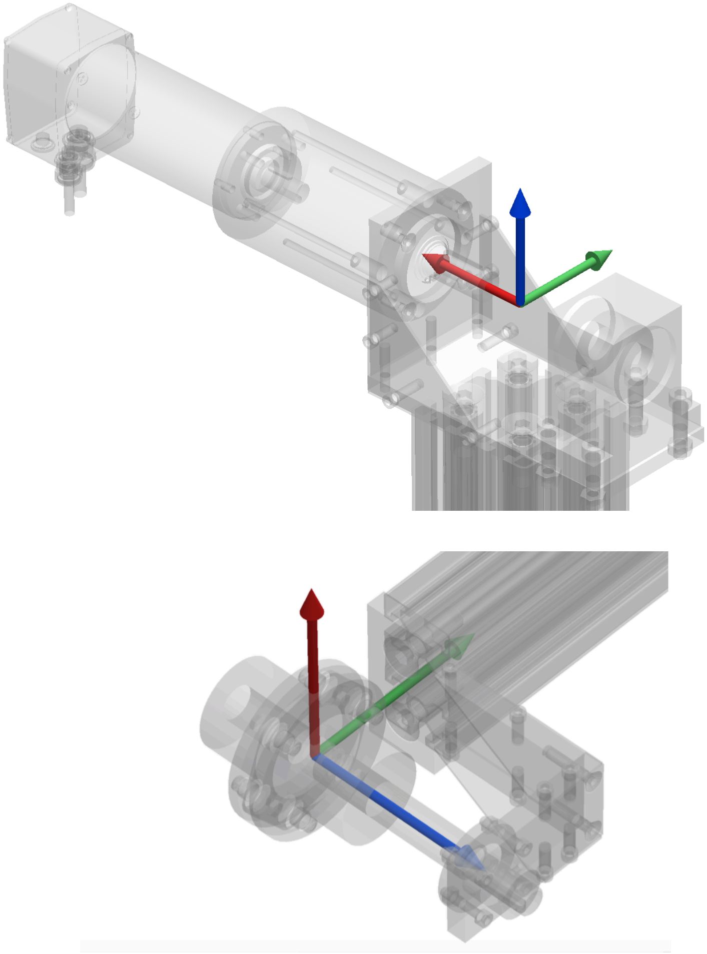


Figure 38: Frame 1 and 2 (*x=green, y=red, z=blue*).

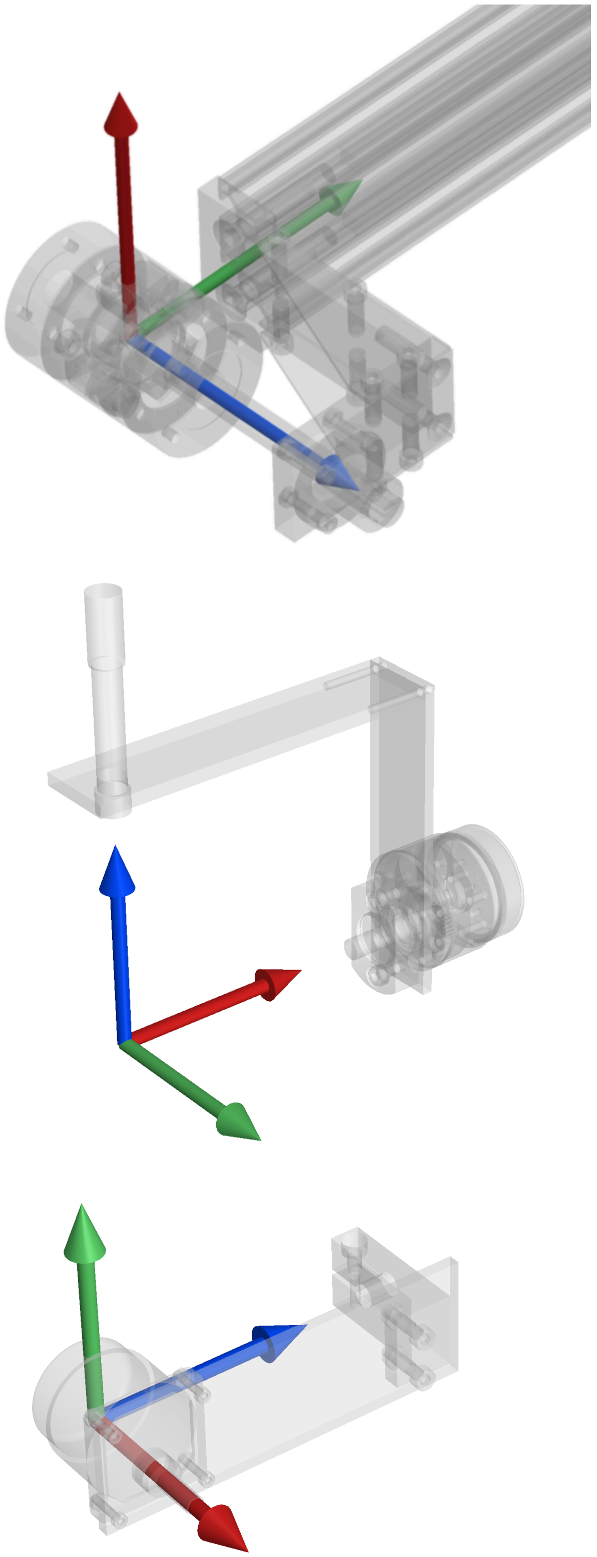


Figure 39: Frame 3, 4 and 5 (*x=green, y=red, z=blue*).

## 3.8 INERTIAL PROPERTIES

- Assembly 1: *link, motor, support structure, hardware fasteners*

$$m_1 = 15387.08\,g \tag{88}$$

$$\overrightarrow{Com_1} = \begin{bmatrix} -78.18 \\ -0.00 \\ -369.55 \end{bmatrix} mm \tag{89}$$

$$I_1 = \begin{bmatrix} 4.93 \cdot 10^9 & -5.37 \cdot 10^3 & 2.2199 \cdot 10^6 \\ -5.37 \cdot 10^3 & 5.232 \cdot 10^9 & 7.747 \cdot 10^4 \\ 2.2199 \cdot 10^6 & 7.747 \cdot 10^4 & 3.1078 \cdot 10^8 \end{bmatrix} g \cdot mm^2 \tag{90}$$

- Assembly 2: *link, motor, support structure, hardware fasteners*

$$m_2 = 4896.27\,g \tag{91}$$

$$\overrightarrow{Com_2} = \begin{bmatrix} 491.45 \\ 0.03 \\ -24.96 \end{bmatrix} mm \tag{92}$$

$$I_2 = \begin{bmatrix} 3.3917 \cdot 10^7 & 1.8840 \cdot 10^4 & -1.4974 \cdot 10^8 \\ 1.8840 \cdot 10^4 & 1.7614 \cdot 10^9 & -5.4241 \cdot 10^3 \\ -1.4974 \cdot 10^8 & -5.4241 \cdot 10^3 & 1.732 \cdot 10^9 \end{bmatrix} g \cdot mm^2 \tag{93}$$

- Assembly 3: *link, motor, support structure, hardware fasteners*

$$m_3 = 2299.15\,g \tag{94}$$

$$\overrightarrow{Com_3} = \begin{bmatrix} 409.64 \\ 7.25 \\ 13.04 \end{bmatrix} mm \tag{95}$$

$$I_3 = \begin{bmatrix} 4.9254 \cdot 10^6 & 1.3251 \cdot 10^7 & 8.4872 \cdot 10^5 \\ 1.3251 \cdot 10^7 & 6.1623 \cdot 10^8 & 6.2855 \cdot 10^3 \\ 8.4872 \cdot 10^5 & 6.2855 \cdot 10^3 & 6.1545 \cdot 10^8 \end{bmatrix} g \cdot mm^2 \tag{96}$$

- Assembly 4: *link, motor, support structure, hardware fasteners*

$$m_4 = 486.74\, g \tag{97}$$

$$\overrightarrow{Com_4} = \begin{bmatrix} 0.36 \\ 103.31 \\ 40.97 \end{bmatrix} mm \tag{98}$$

$$I_4 = \begin{bmatrix} 9.182 \cdot 10^6 & 1.335 \cdot 10^3 & 2.243 \cdot 10^3 \\ 1.335 \cdot 10^3 & 2.4517 \cdot 10^6 & 6.6737 \cdot 10^5 \\ 2.243 \cdot 10^3 & 6.6737 \cdot 10^5 & 6.8299 \cdot 10^6 \end{bmatrix} g \cdot mm^2 \tag{99}$$

- Assembly 5: *link, motor, support structure, hardware fasteners*

$$m_5 = 210.98\, g \tag{100}$$

$$\overrightarrow{Com_5} = \begin{bmatrix} 5.55 \\ 16.98 \\ 30.06 \end{bmatrix} mm \tag{101}$$

$$I_5 = \begin{bmatrix} 6.842 \cdot 10^5 & 1.9588 \cdot 10^4 & 1.4696 \cdot 10^4 \\ 1.9588 \cdot 10^4 & 6.162 \cdot 10^5 & 1.2017 \cdot 10^5 \\ 1.4696 \cdot 10^4 & 1.2017 \cdot 10^5 & 1.311 \cdot 10^5 \end{bmatrix} g \cdot mm^2 \tag{102}$$

Finally, the assembled structure has the following characteristics:

- Total weight: 34.405 kg (without harness)
- Maximum extension of the arm: 1680 m
- Maximum height of the arm: 3170 m (with basement)

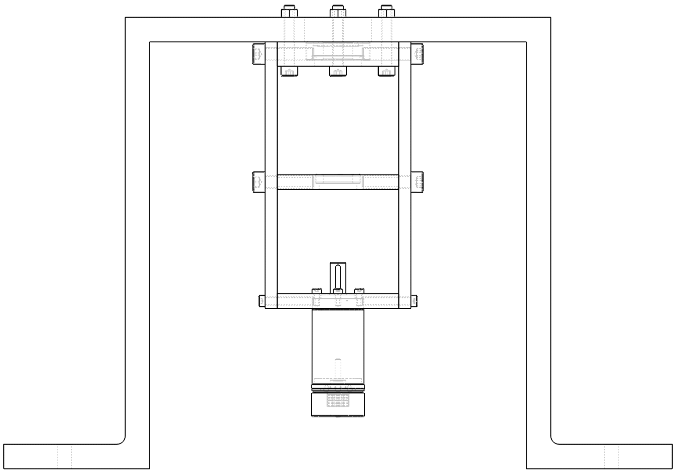


Figure 40: Base.

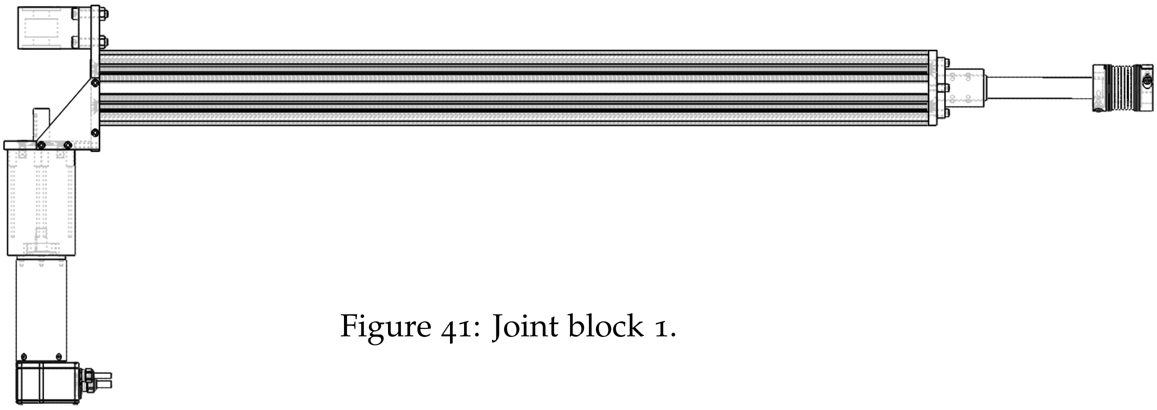


Figure 41: Joint block 1.

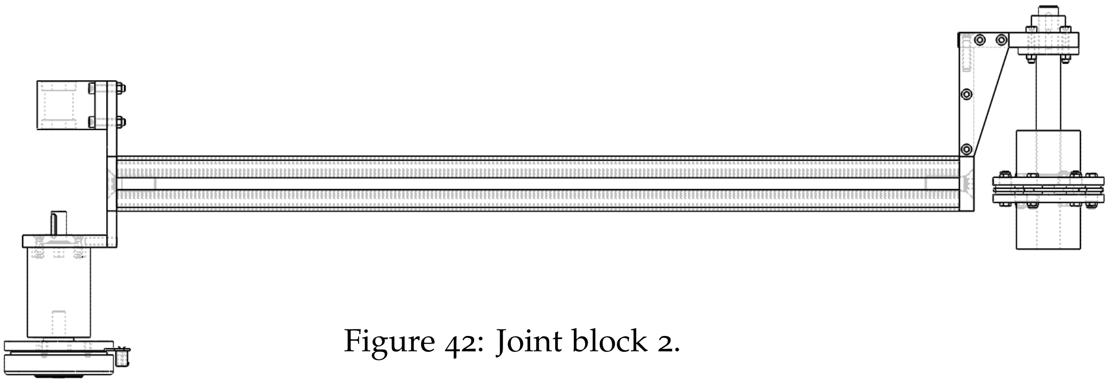


Figure 42: Joint block 2.

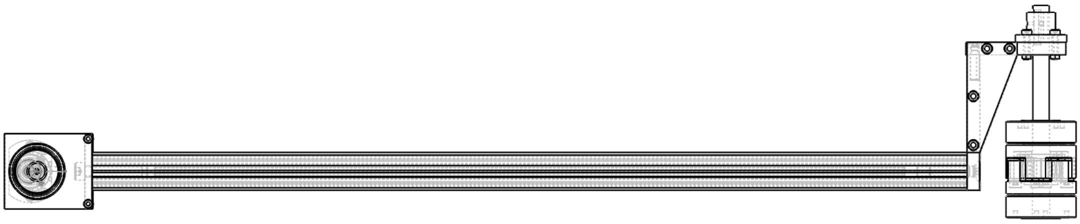


Figure 43: Joint block 3.

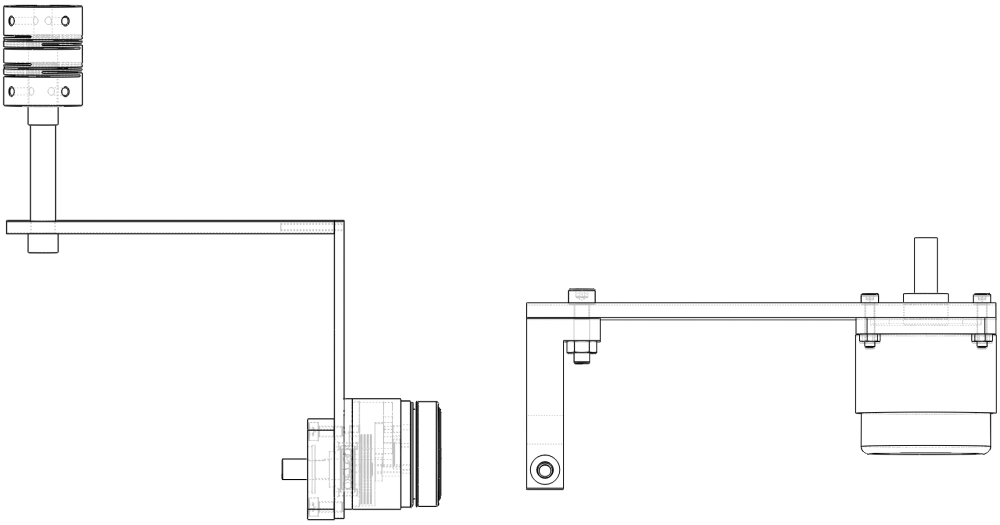


Figure 44: Joint block 4 and 5.

## 3.9 SOLIDWORKS RENDERS

In this section the graphical rendering of the manipulator is presented. First, each link assembly (link, motor, support structure, hardware fasteners) is displayed; subsequently, the whole structure is rendered in different positions and configurations.

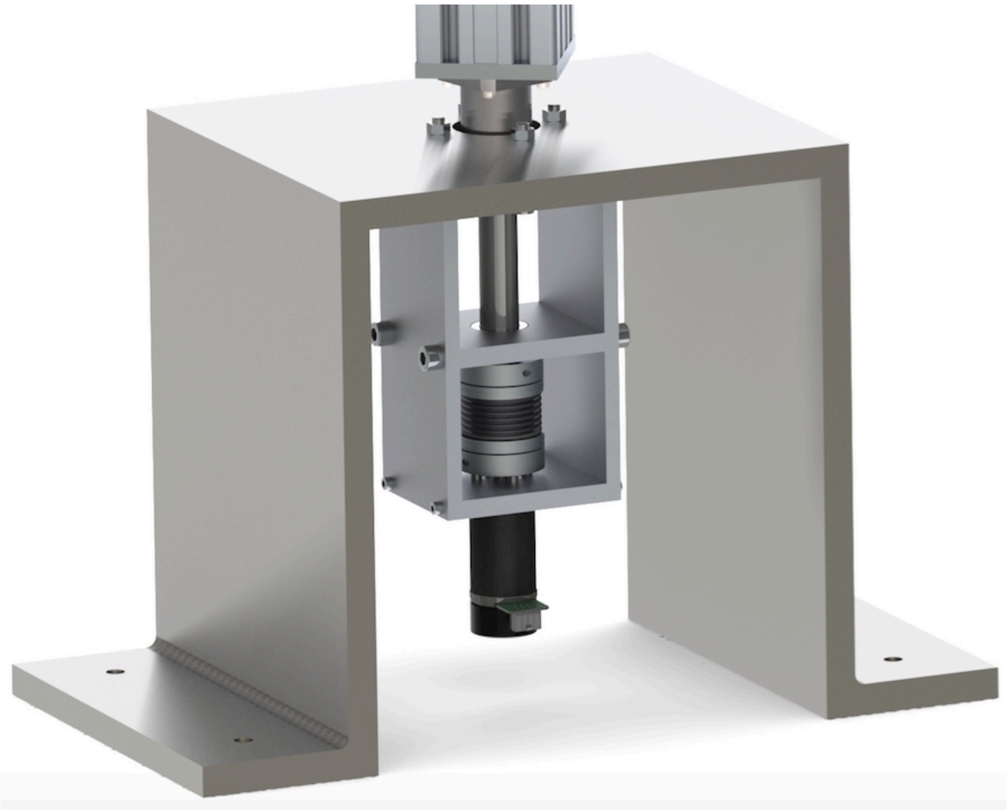


Figure 45: Render, base.

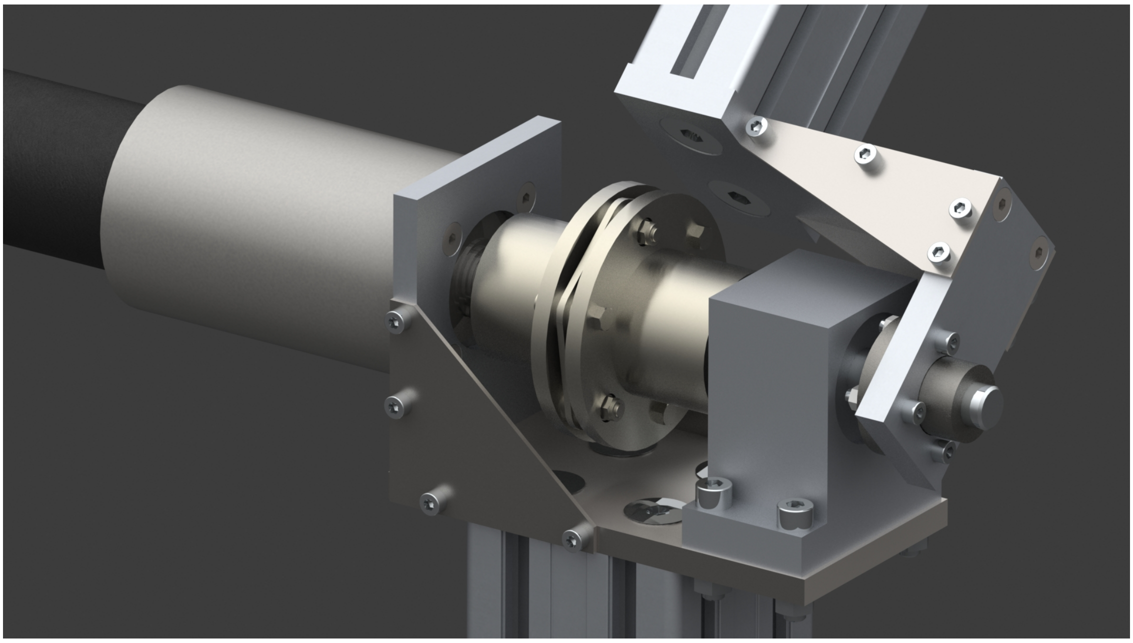


Figure 46: Render, joint 2.

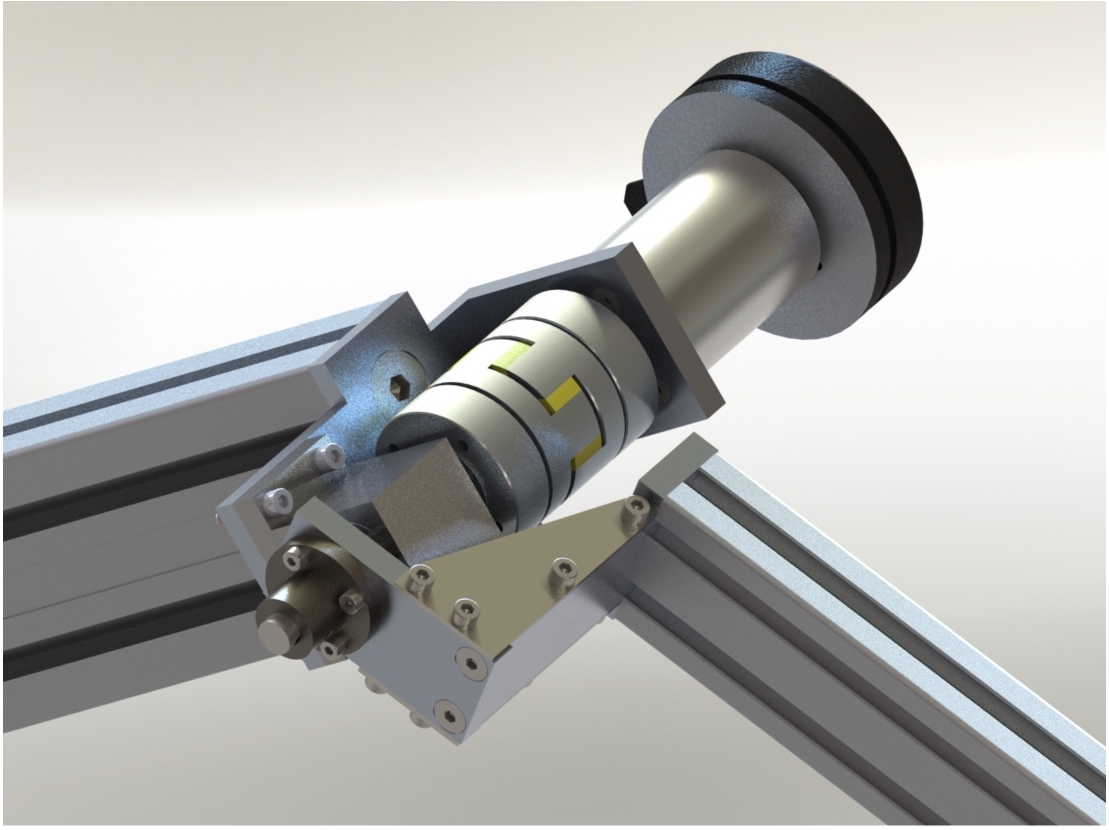


Figure 47: Render, joint 3.

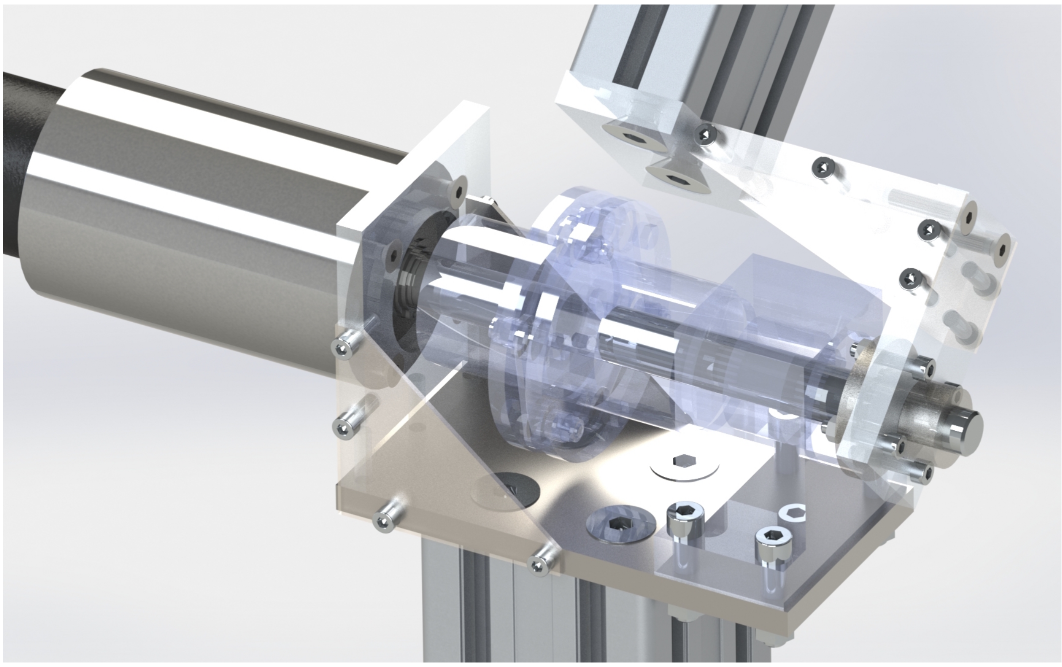


Figure 48: Render, joint 2. Detail.

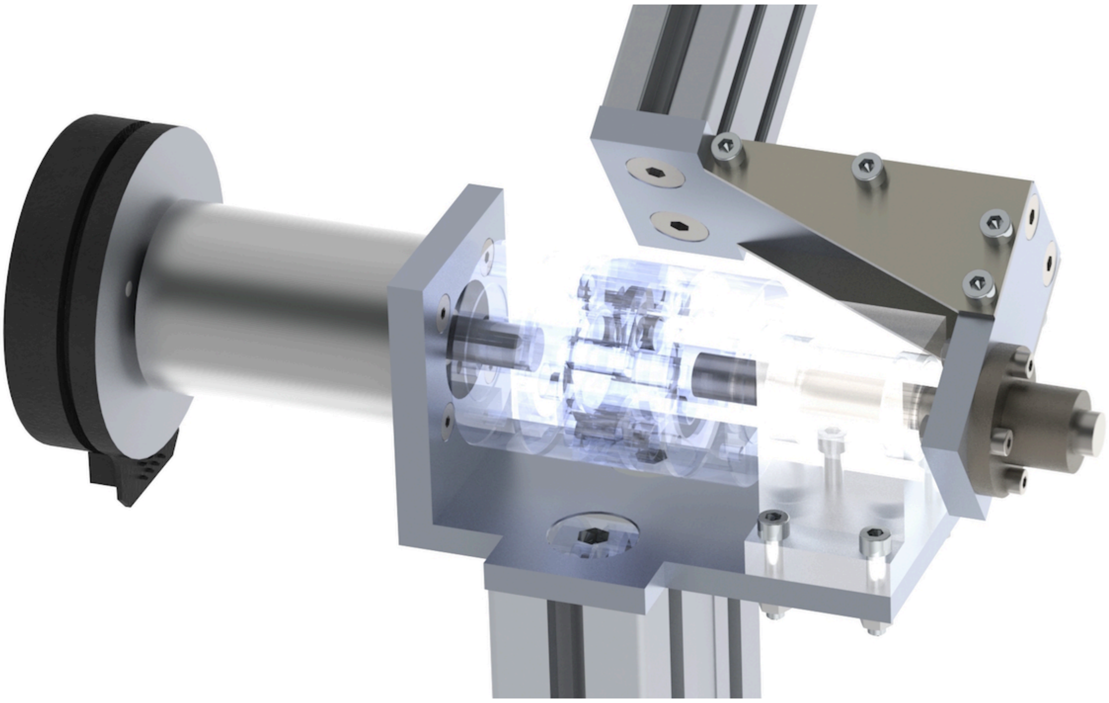


Figure 49: Render, joint 3. Detail.

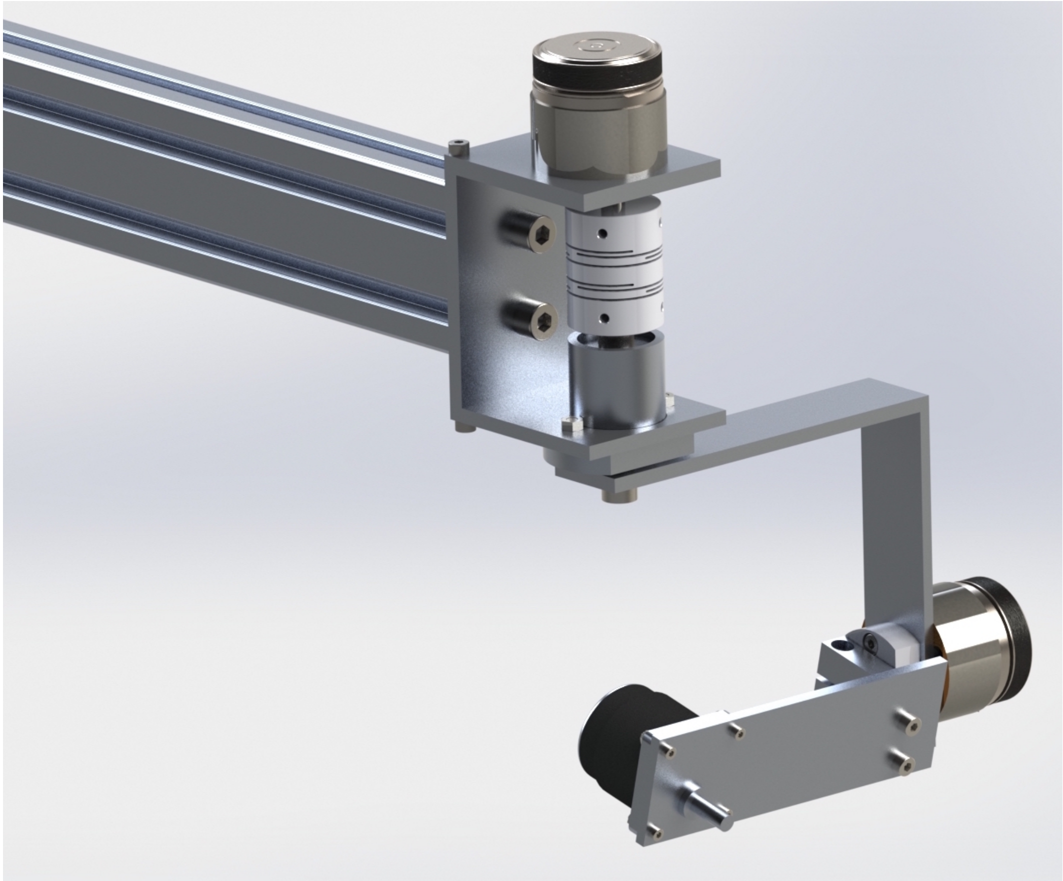


Figure 50: Render, end effector.

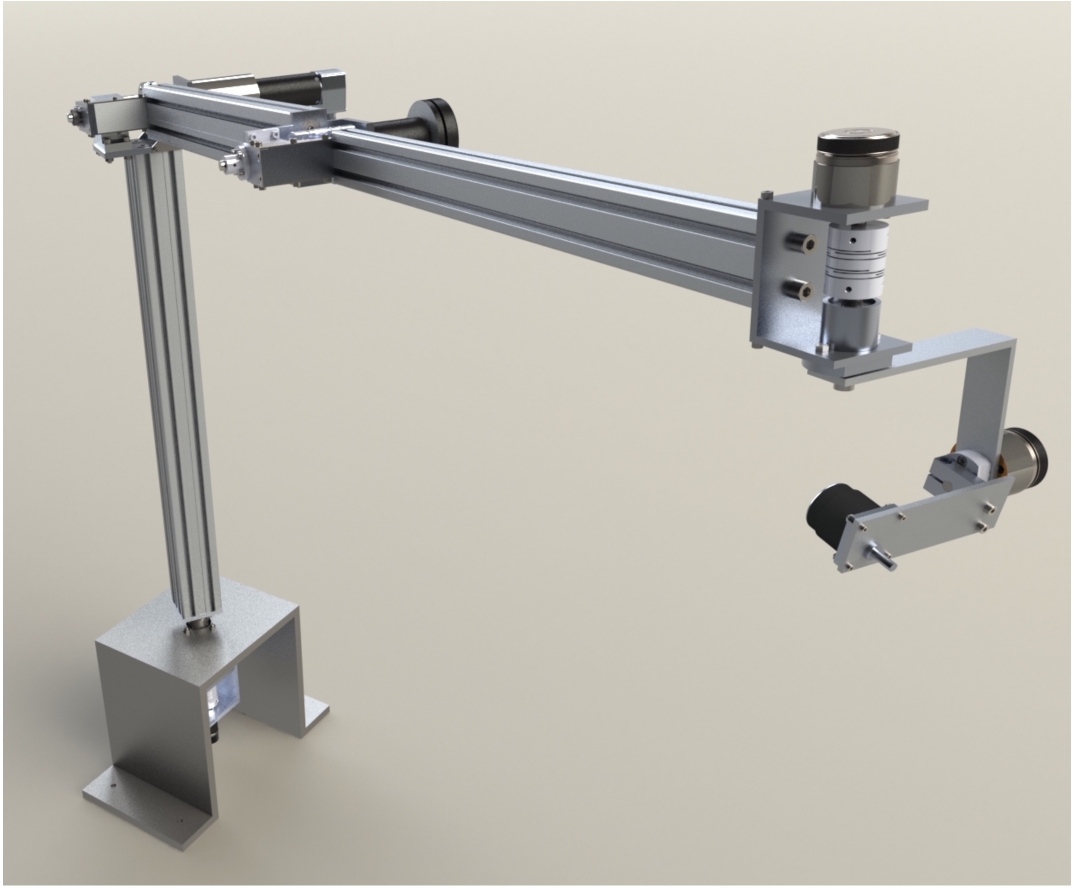


Figure 51: Render, complete arm.

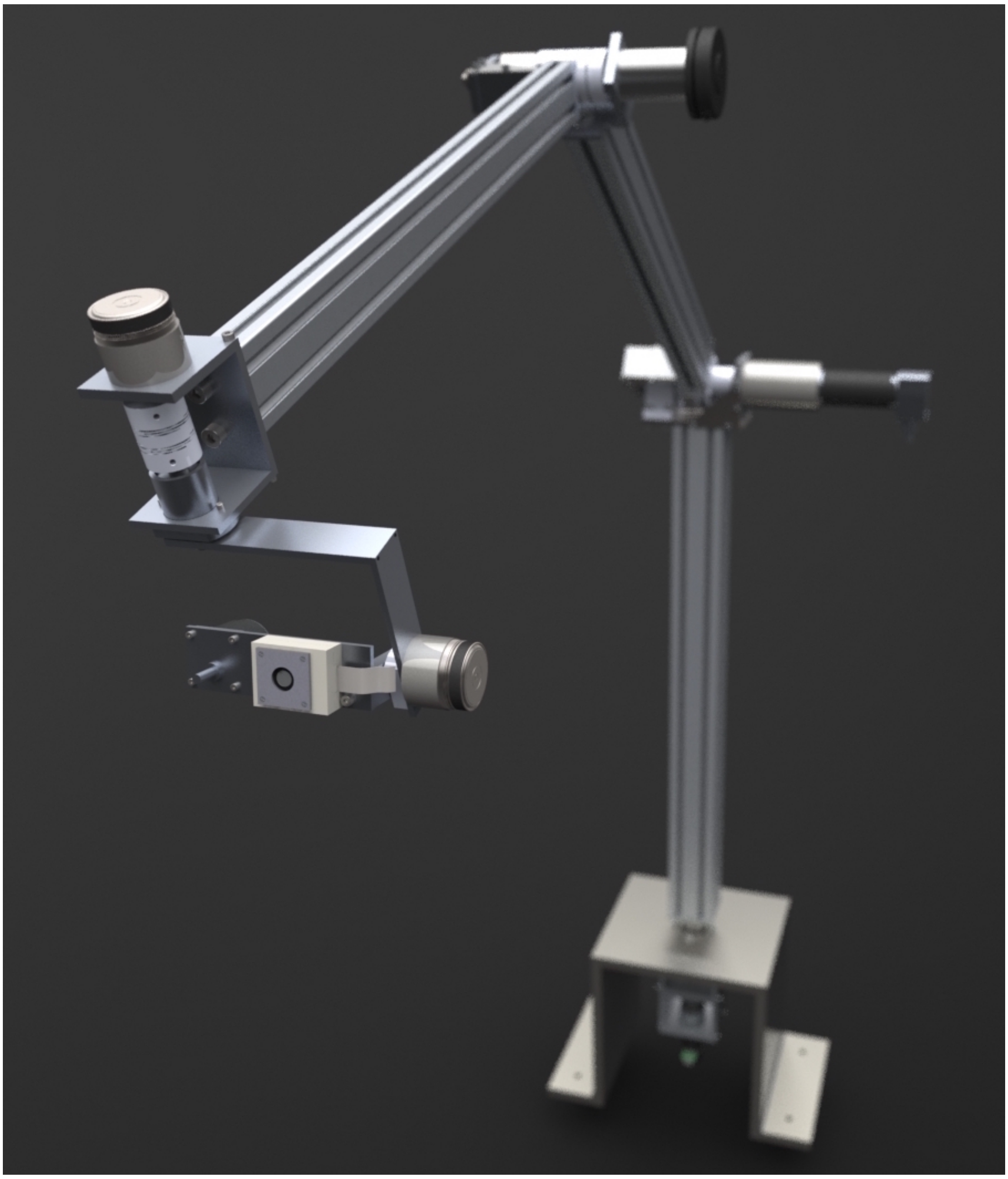


Figure 52: Render, complete arm with mounted Sun sensor.

## 3.10 MANUFACTURING, PROCUREMENT AND ASSEMBLING

In this section the hardware for the manipulator facility is presented. The motors have been procured from the company Maxon® Motors and the interfaces are custom manufactured. The pictures show the procured pieces and the assembled joints. Fasteners used for this application are metric standard. Additional details are available upon request.

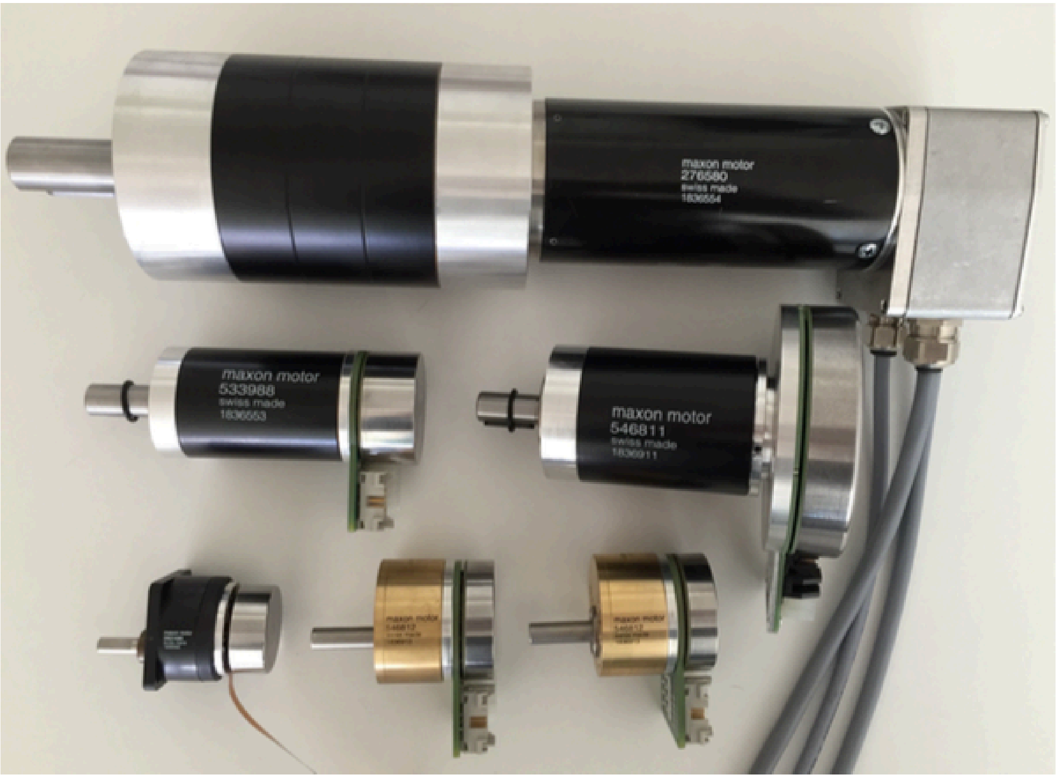


Figure 53: Motors with attached gearings, Maxon®.

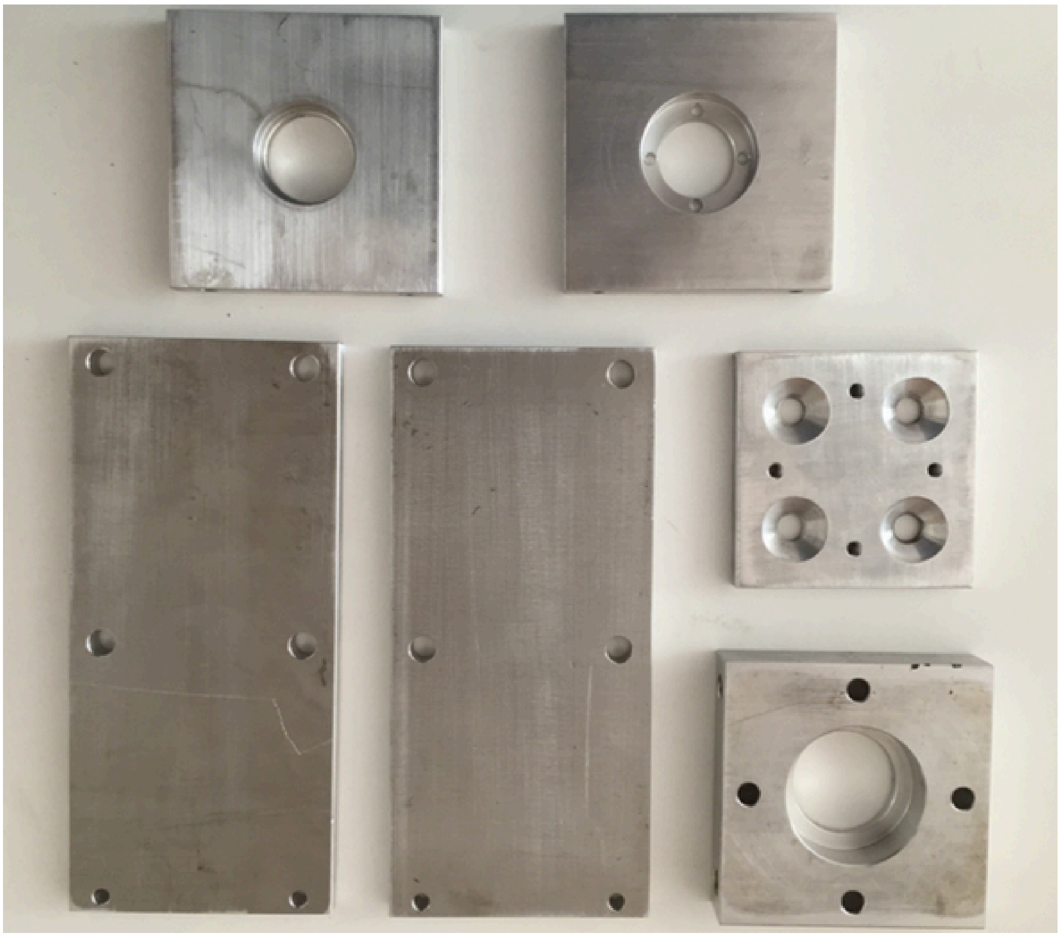


Figure 54: Mechanical parts for *joint 1*, Aluminum Alloy 6082.

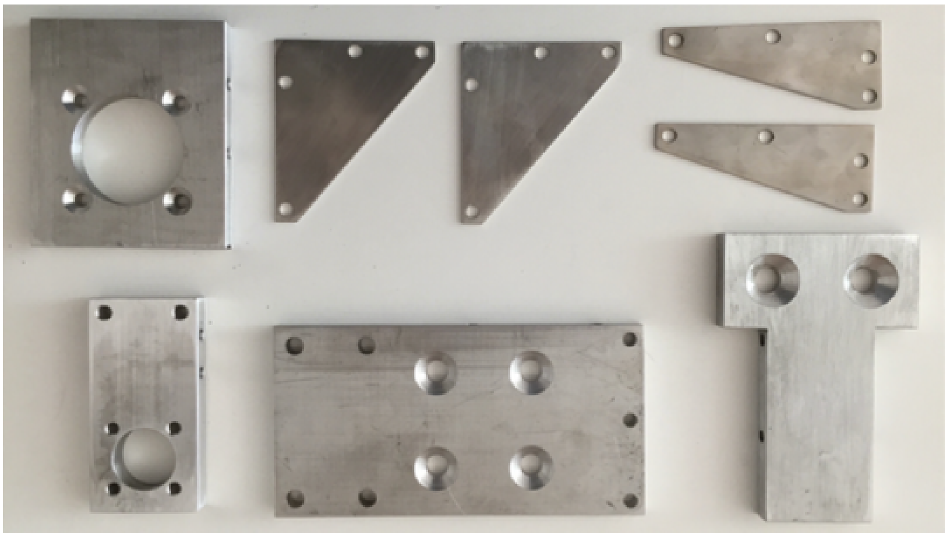


Figure 55: Mechanical parts for *joint 2*, A.A. 6082 and Stainless Steel 304.

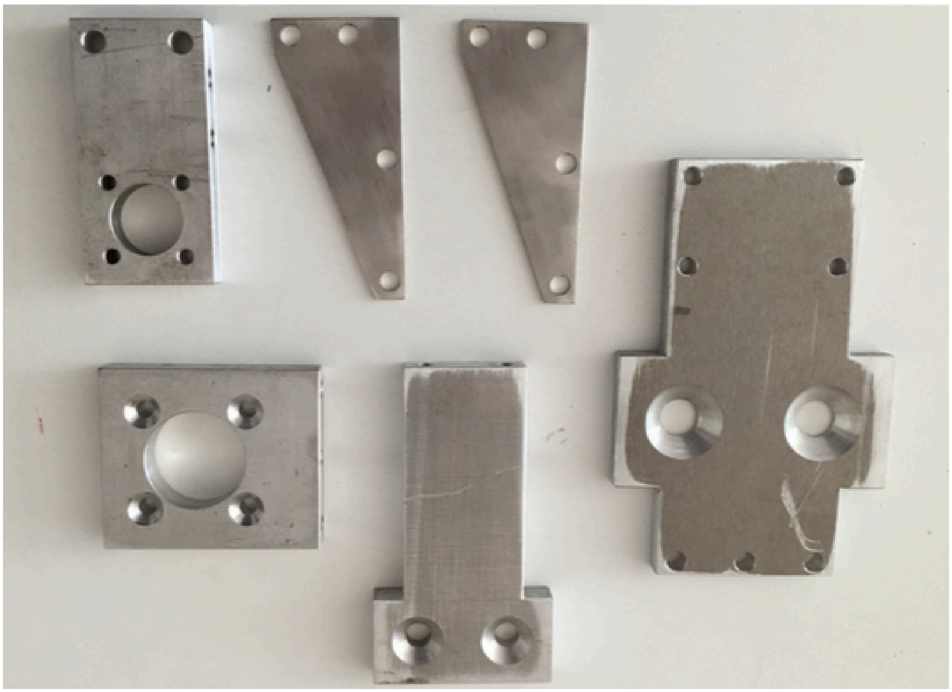


Figure 56: Mechanical parts for *joint 3*, A. A. 6082 and Stainless Steel 304.

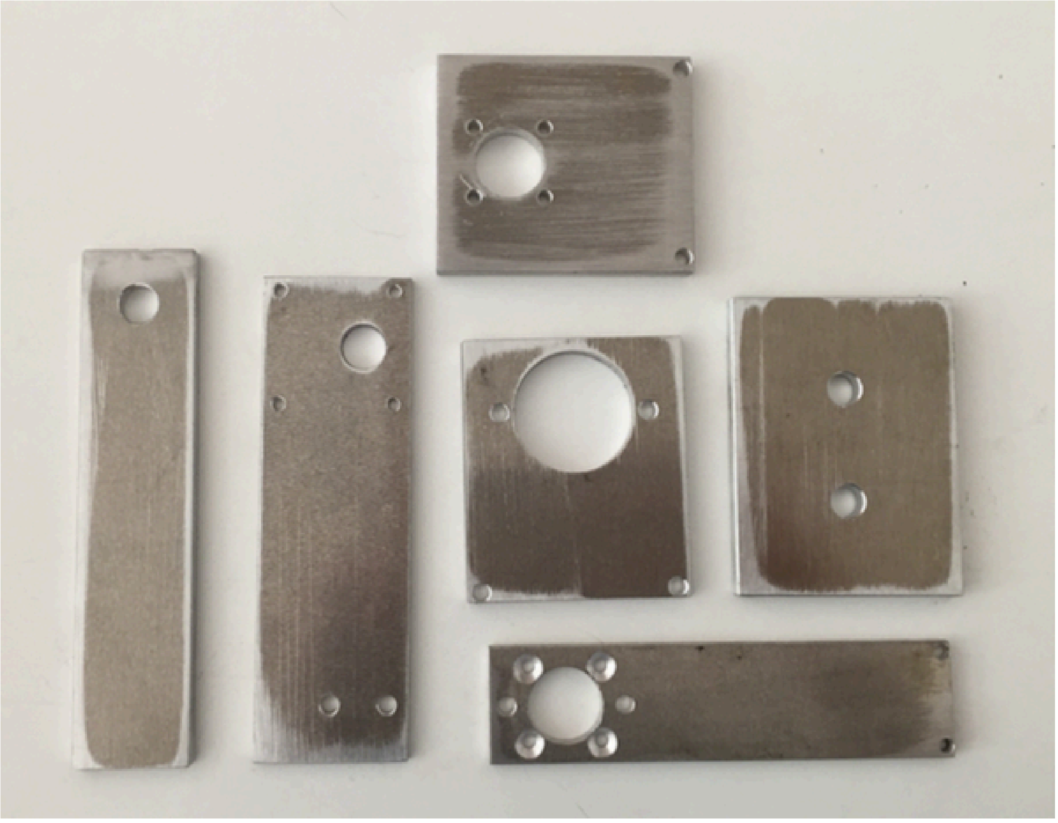


Figure 57: Mechanical parts for *joint 4* and *5*, Aluminum Alloy 6082.

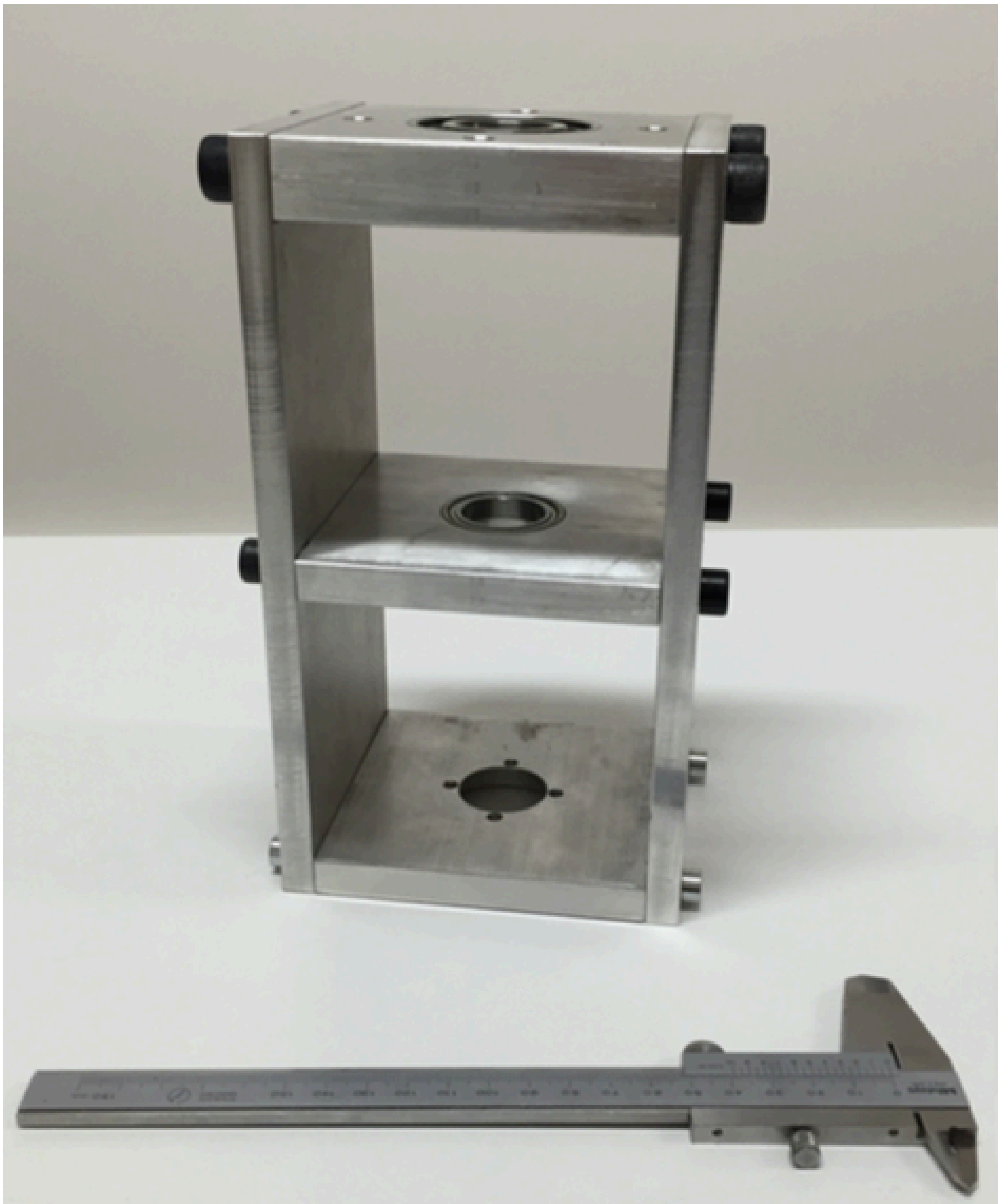


Figure 58: *Joint 1* assembly, no motor installed.

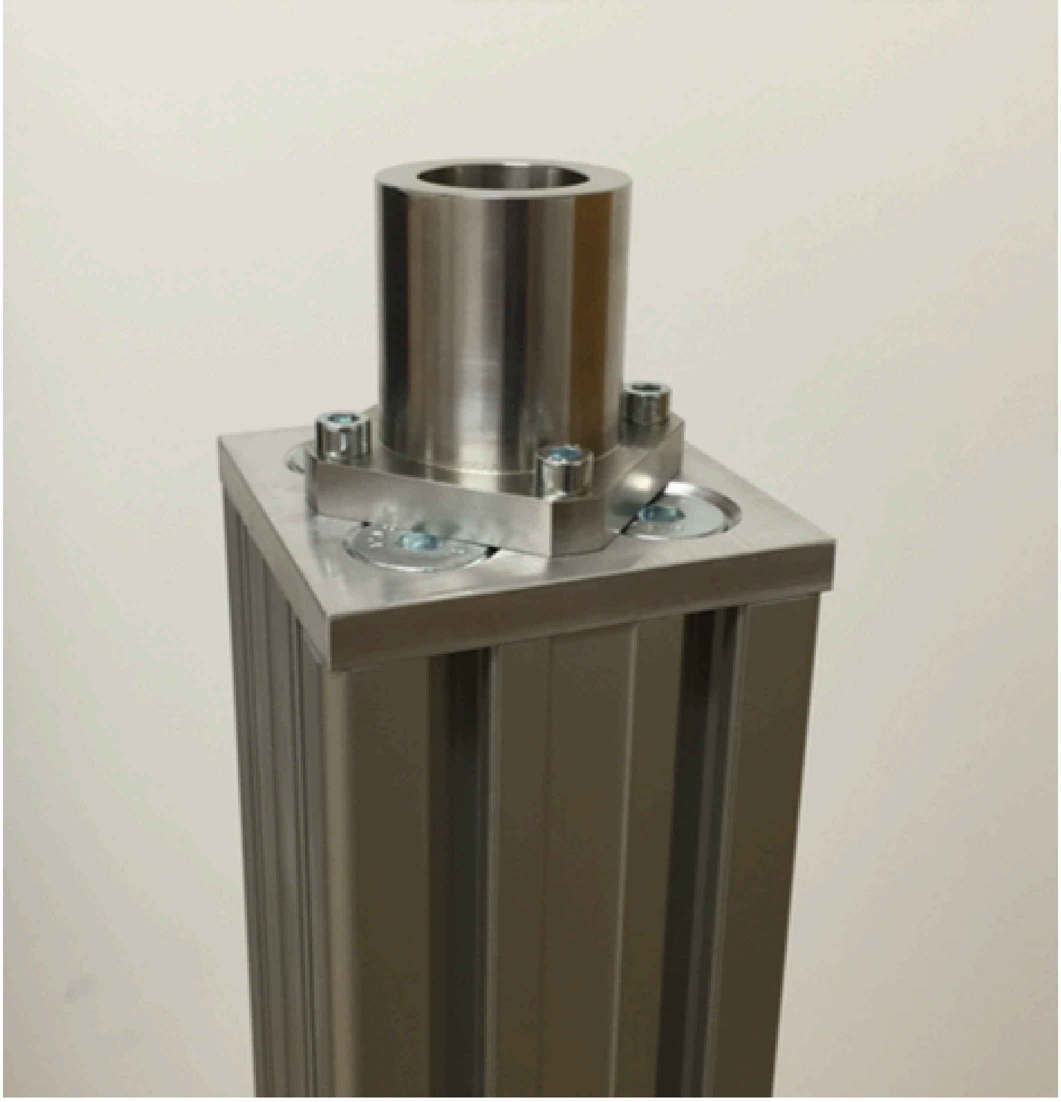


Figure 59: Detail of *link 1* with shaft holder. Bottom view.

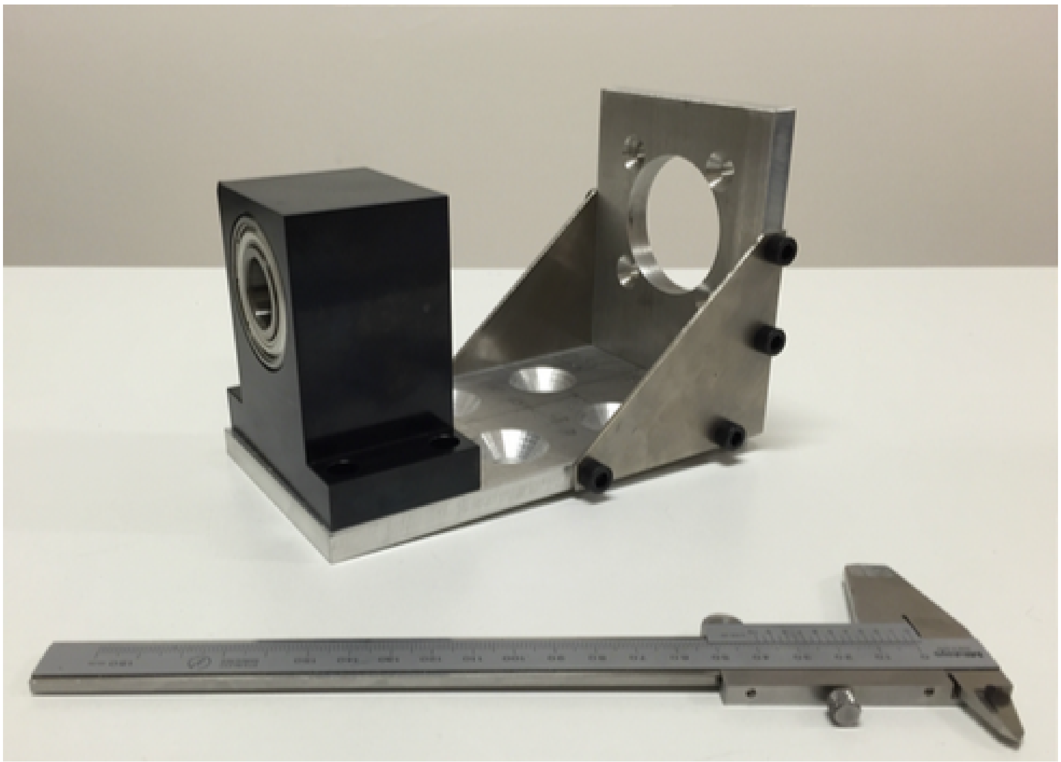


Figure 60: *Joint 2* assembly, no motor installed. The black component is the dual bearings shaft holder.

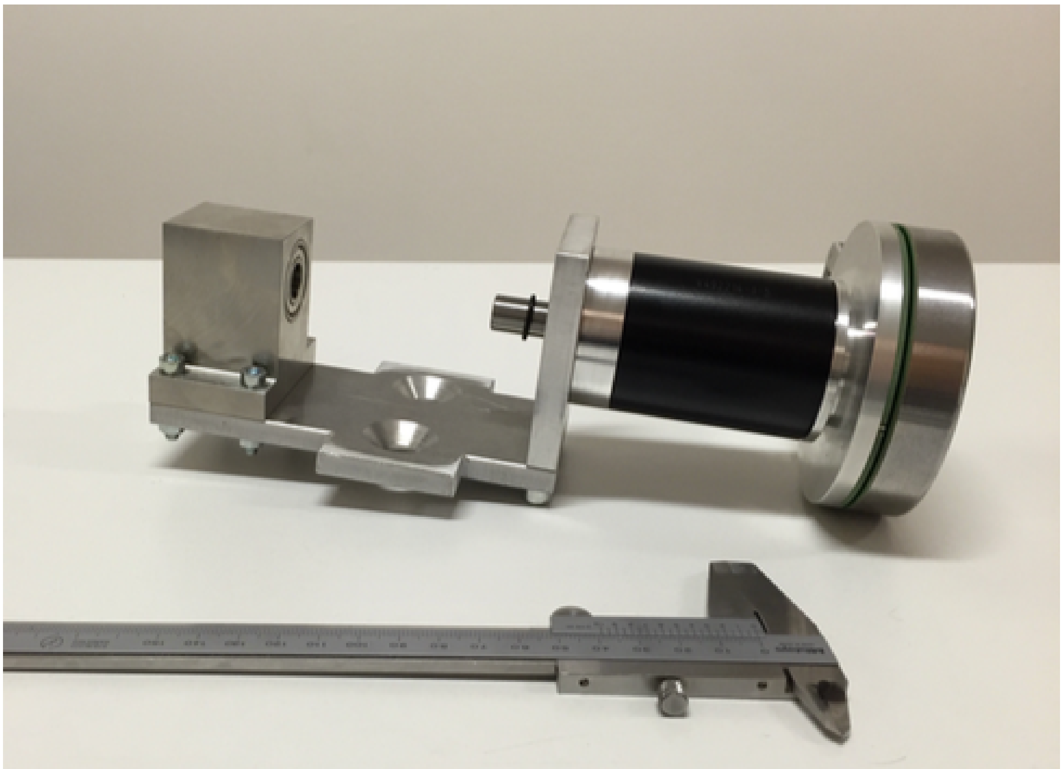


Figure 61: *Joint 3* assembly, motor installed. The silver component is the dual bearings shaft holder

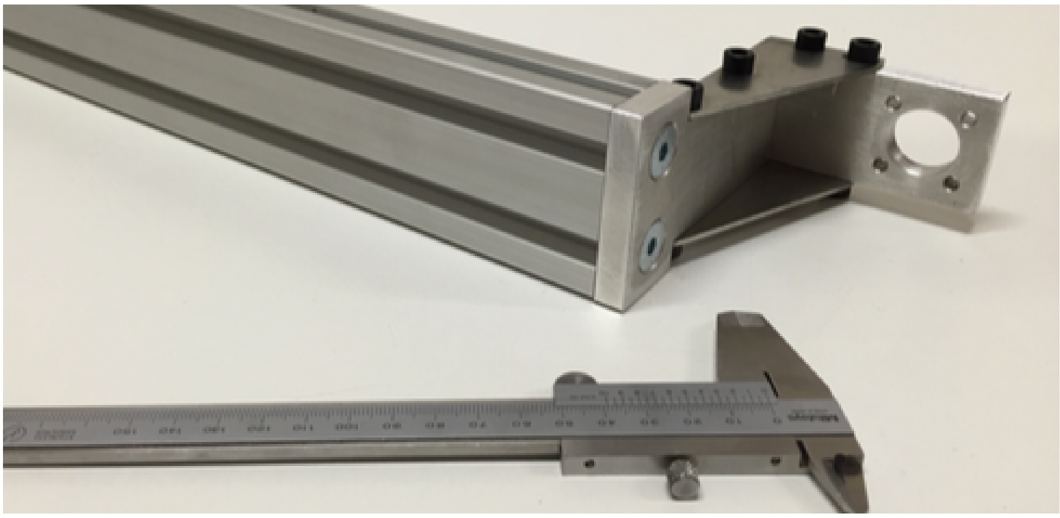


Figure 62: *Link 3* assembly, detail of the shaft-link interface. No shaft holder installed.

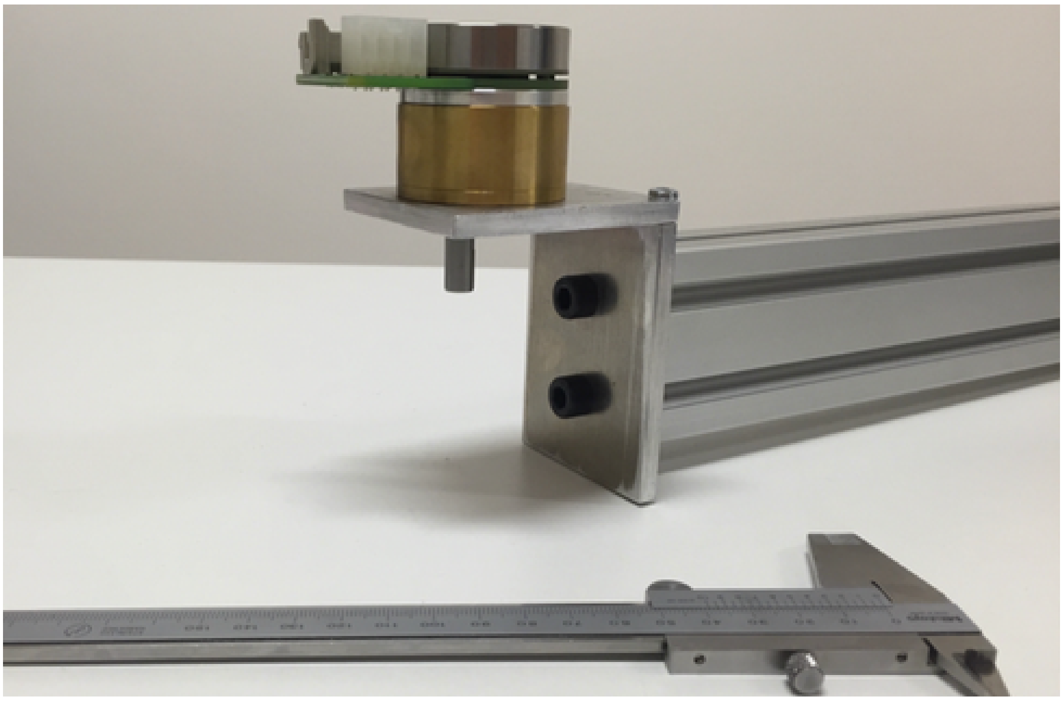


Figure 63: *Joint 4* assembly, mounted on *link 3*, with motor installed.

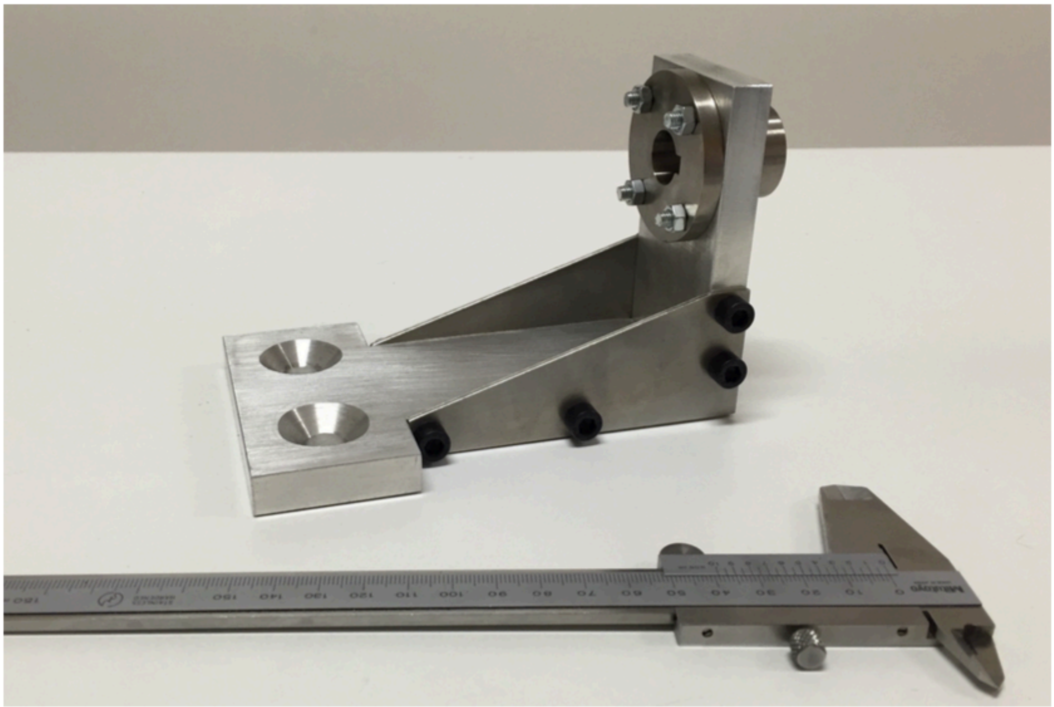


Figure 64: *Link 3* shaft-link interface assembly, shaft holder installed.

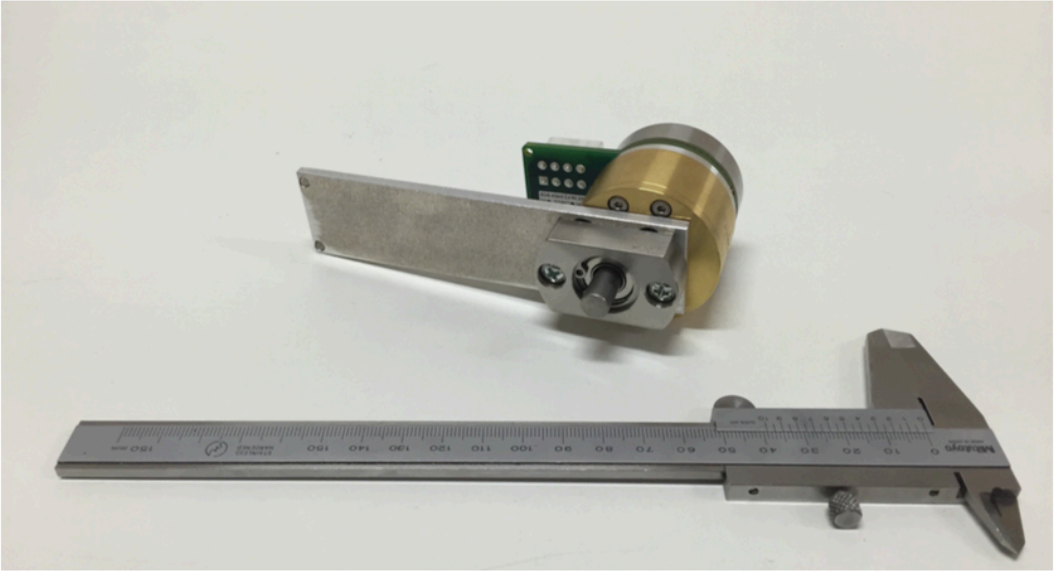


Figure 65: *Joint 5* assembly, mounted on *link 4*, with motor installed.

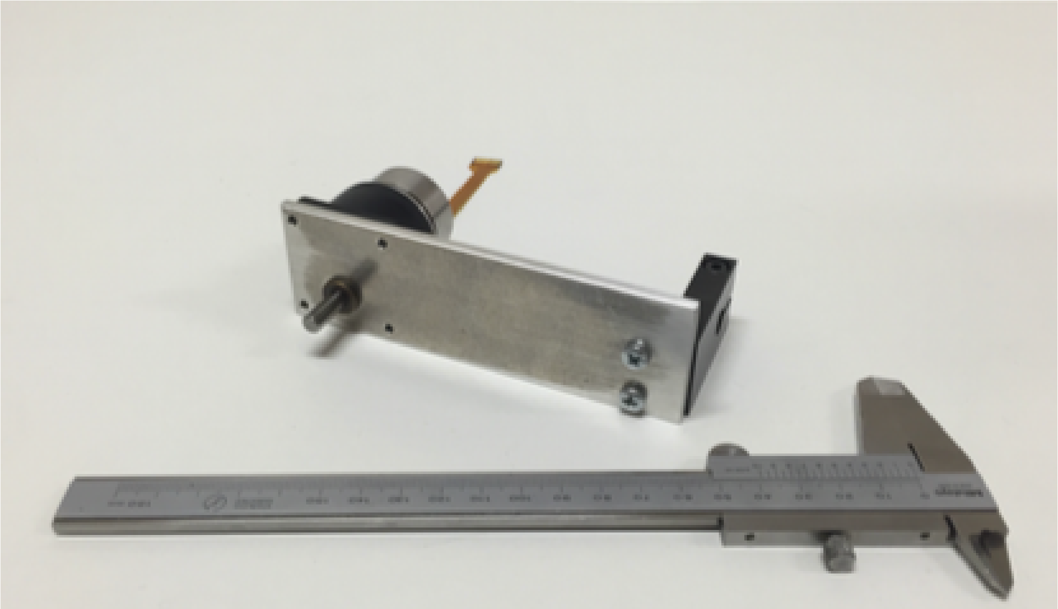


Figure 66: *Joint 6* assembly, mounted on *link 5*, with motor installed.

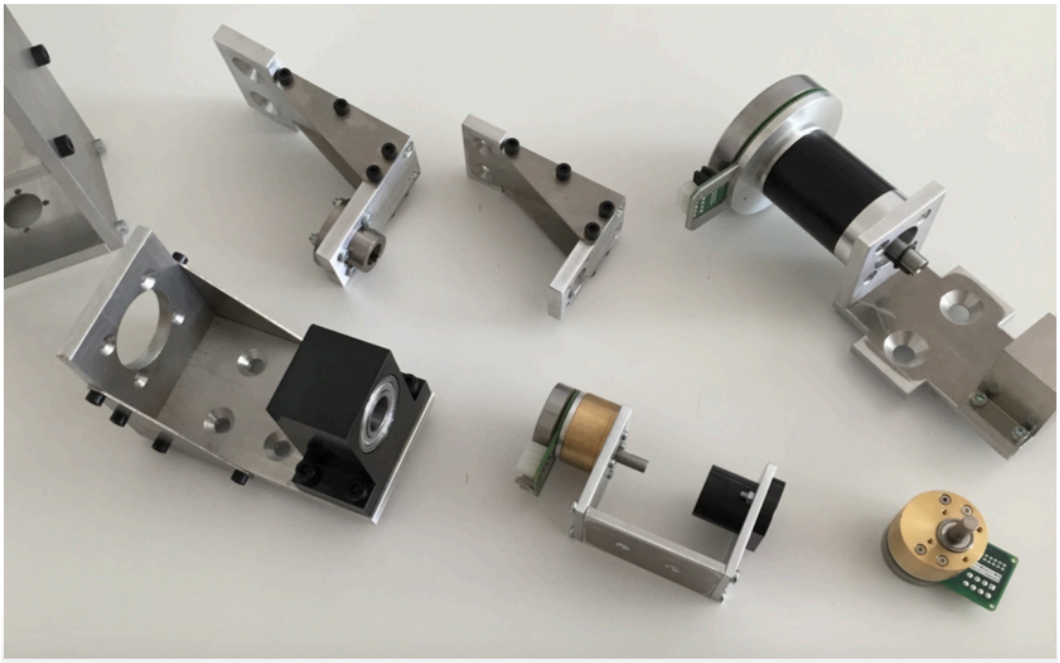


Figure 67: Global view of the assembled components as of November 2016.

### 3.10.1 Motor and rack wiring

In the following figures the rack assembly with the controllers and switching power supply unit, along with the wired motors are presented. The pictures were taken while performing the parameter tuning of the motors.

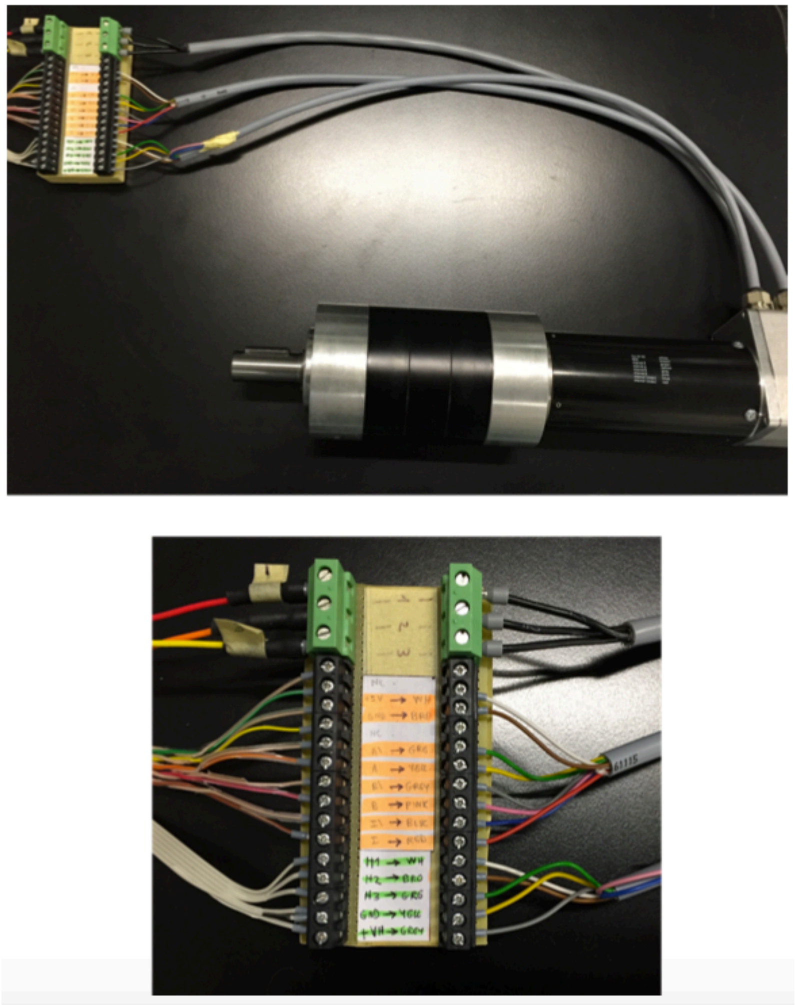


Figure 68: Motor 2 under parameter tuning and closeup of custom breakout board.

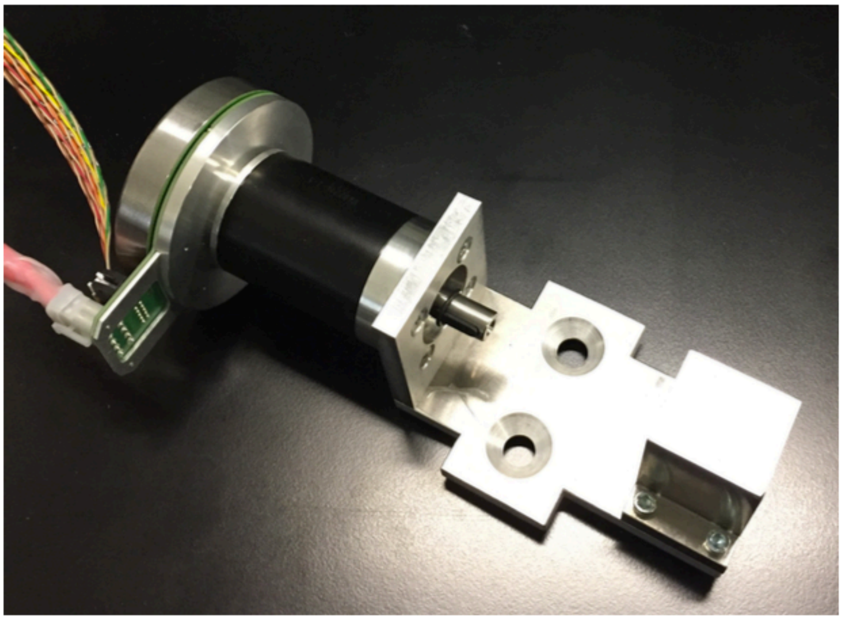


Figure 69: Motor 3 under parameter tuning.

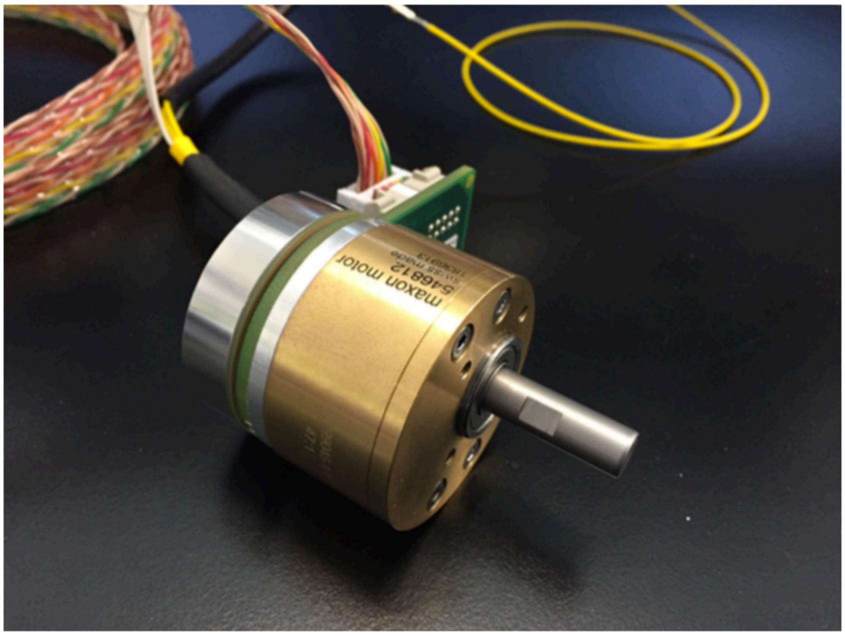


Figure 70: Motor 4 under parameter tuning.

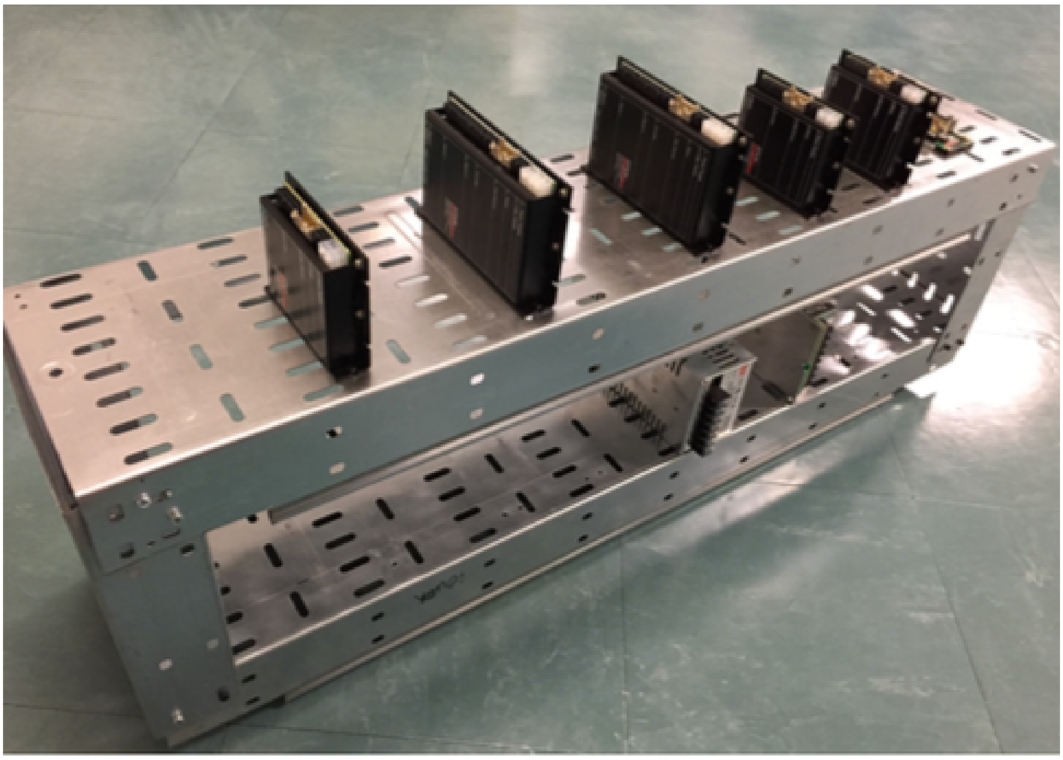


Figure 71: Rack assembly under construction. Controller are placed on the top shelf and power supply on the lower part.

# 4

# FINITE ELEMENTS ANALYSIS

## 4.1 PROBLEM DEFINITION

Once the CAD model of the facility has been finalized (refer to the previous chapter), it is possible to perform further inspection of the mechanical performances.

In this chapter, the static performances are analyzed and compared to the previous results obtained with the simplified Matlab® model. This ultimately proved the validity of the model, which is in accordance with the FEM simulations: a slight difference in the results is detected due to the presence of the interfaces, which add some flexibility to the system.

In addition, after the correctness of the model was verified for the static case, attention was given to the modal analysis of the multi-body system, in order to inspect the vibration modes and the effect of the elasticity of the bodies. Modal analysis is needed in order to investigate the dynamics of the contacts that can be simulated by the facility without incurring into excessive vibration or resonance: this can be ultimately used to design a range of contact experiments in terms of characteristics of the contact force (frequency, amplitude and direction).

A classic approach to solve elastodynamics problem uses the Finite Element Method (FEM), based on the discretization of the system in smaller slabs with elastic and inertia properties [21].

Most research omits the analysis of the dependency of the shift of the modal frequencies with respect to the joint configuration; this is fundamental for the development of an optimal control aimed to the reduction of vibrations [22].

A FEM model designed in Ansys® 17 allowed for the simulations of the modal behavior in a discrete number of points inside the workspace. Afterwards, with the aid of interpolation, using nonlinear regression tools, the spatial modal continuous distribution was estimated: the designed model provided a goodness of fit of $R^2$=0.93.

Finally, using the information that the modal analysis provided, slight mechanical modifications took place in order to augment the rigidity of the system without modifying significantly the compound mass.

## 4.2 STATIC LOADING VERIFICATION

In order to compare the FEM model against the Matlab® simulations for the deflection, we performed a cohort of different loading and configuration scenarios. We present below an extract of the verification campaign [10].

First of all, the unloaded case was taken into account. Configurations compared are, in terms of joint angles:

- $\theta_2 = 0°$; $\theta_3 = 0°$
- $\theta_2 = 30°$; $\theta_3 = -60°$
- $\theta_2 = 80°$; $\theta_3 = -160°$

After the unloaded case, a fictious weight was added to both model, Matlab® and Ansys®: this represents the payload at the robot tip, which will be typically represented by a mockup of a CubeSat and/or a docking interface. In order to take into account a realistic weight (frame, electronics, sensors, harness, etc.), the simulation campaign was performed with a loading of 1 kg [10]. In the following pages, printouts from Matlab® and Ansys® are presented: the black dot in the contour plots indicates the configuration-specific deflection. On the other hand, the maximum value in the ANSYS® plots can be found in the left-hand color-bar.

### 4.2.1 Case 1.0: unloaded end effector, $\theta_2 = \theta_3 = 0°$

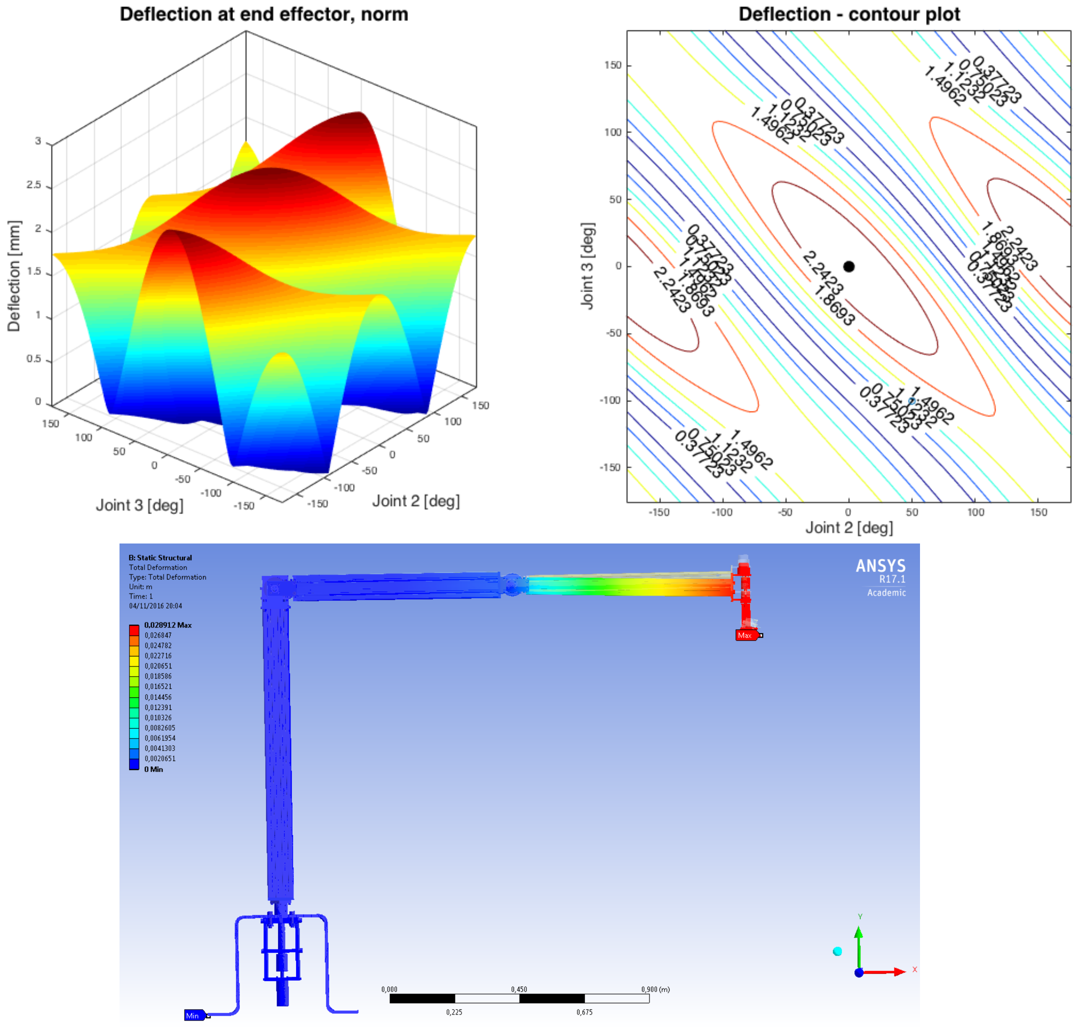


Figure 72: Unloaded robot, $\theta_2$=$\theta_3$=0°, Matlab® model and Ansys® model.

In this case, the angles for *joints* 2 and 3 are $\theta_2 = \theta_3 = 0°$ respectively: the end effector is not loaded with any payload. The comparison between the Matlab® model and the Ansys® simulation yielded the following results for the displacement:

$$\text{Matlab}^{®}: \quad 2.74\,\text{mm}$$

$$\text{Ansys}^{®}: \quad 2.89\,\text{mm} \quad \rightarrow \quad \Delta = 5.5\%$$

### 4.2.2 Case 2.0: unloaded end effector, $\theta_2 = 50°$, $\theta_3 = -100°$

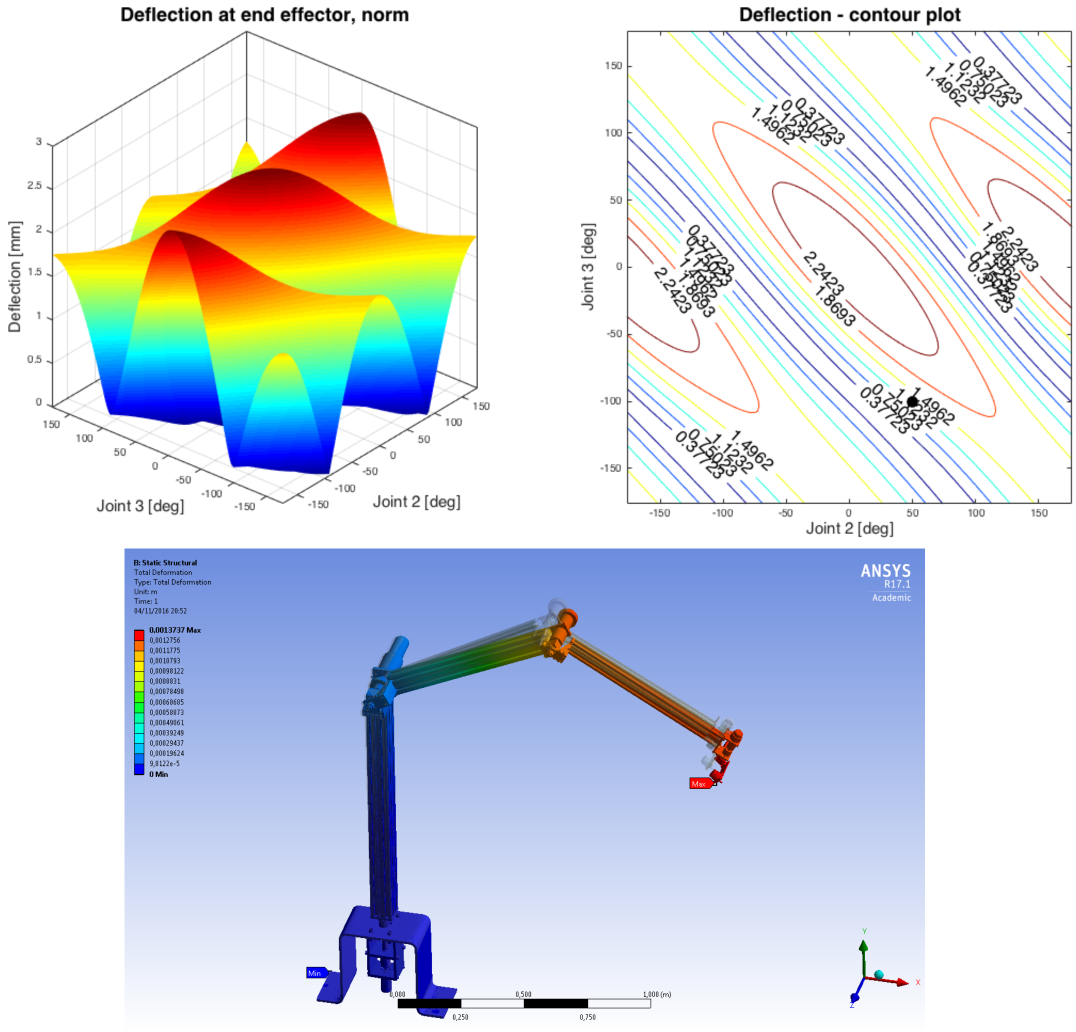


Figure 73: Unloaded case, $\theta_2 = 50°$, $\theta_3 = -100°$, Matlab® and Ansys® model.

In this case, the angles for *joints* 2 and 3 are $\theta_2 = 50°$, $\theta_3 = -100°$ respectively: the end effector is not loaded with any payload. The comparison between the Matlab® model and the Ansys® simulation yielded the following results for the displacement:

| | | | |
|---|---|---|---|
| Matlab®: | 1.31 mm | | |
| Ansys®: | 1.37 mm | → | $\Delta = 4.6\%$ |

### 4.2.3 Case 3.0: unloaded end effector, $\theta_2 = 80°$, $\theta_3 = -95°$

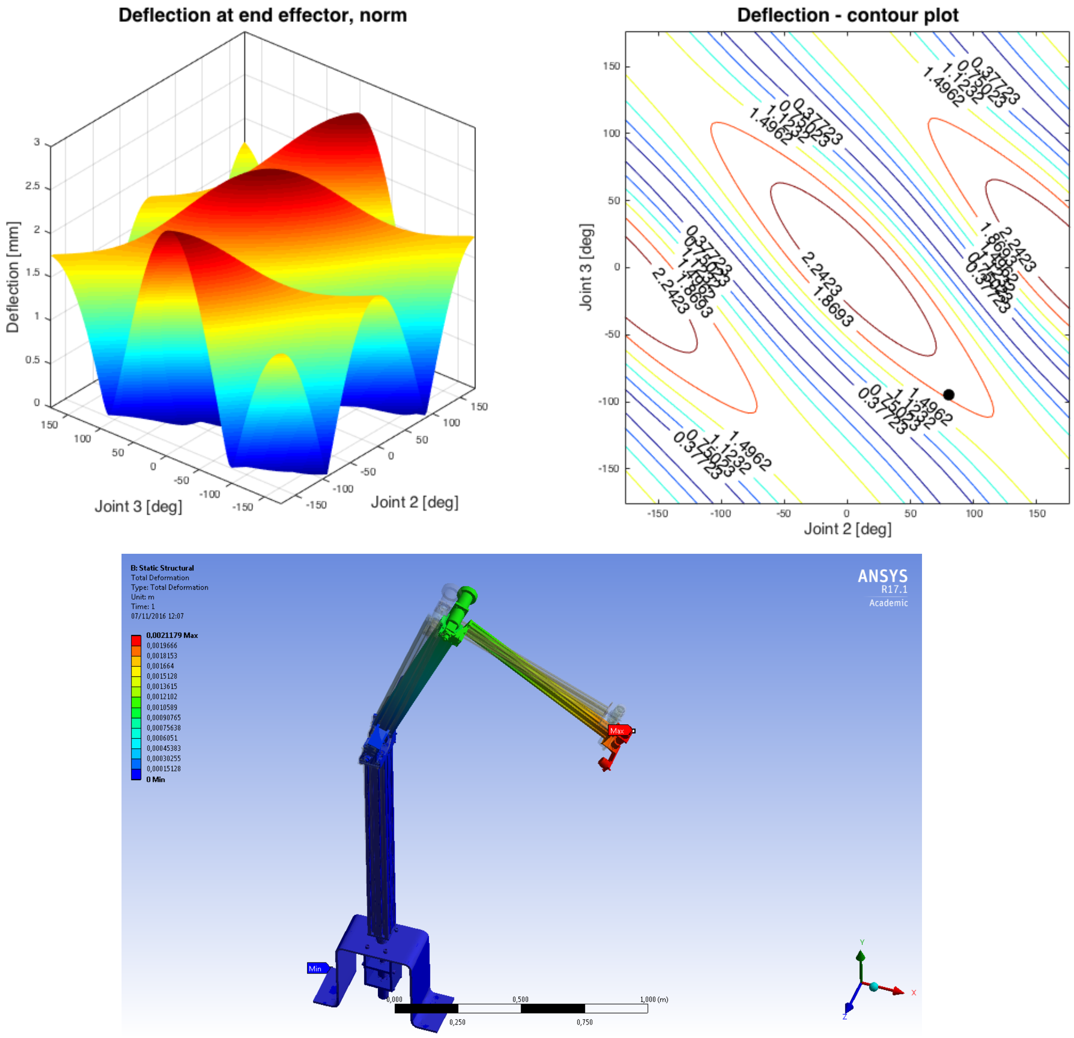


Figure 74: Unloaded case, $\theta_2 = 80°$, $\theta_3 = -950°$, Matlab® and Ansys® model.

In this case, the angles for *joints* 2 and 3 are $\theta_2 = 80°$, $\theta_3 = -95°$ respectively: the end effector is not loaded with any payload. The comparison between the Matlab® model and the Ansys® simulation yielded the following results for the displacement:

$$\text{Matlab}^{®}: \quad 2.03\,\text{mm}$$

$$\text{Ansys}^{®}: \quad 2.12\,\text{mm} \quad \rightarrow \quad \Delta = 4.4\%$$

### 4.2.4 Case 1.1: loaded e. e., $\theta_2 = \theta_3 = 0^\circ$, payload = 1 $kg$

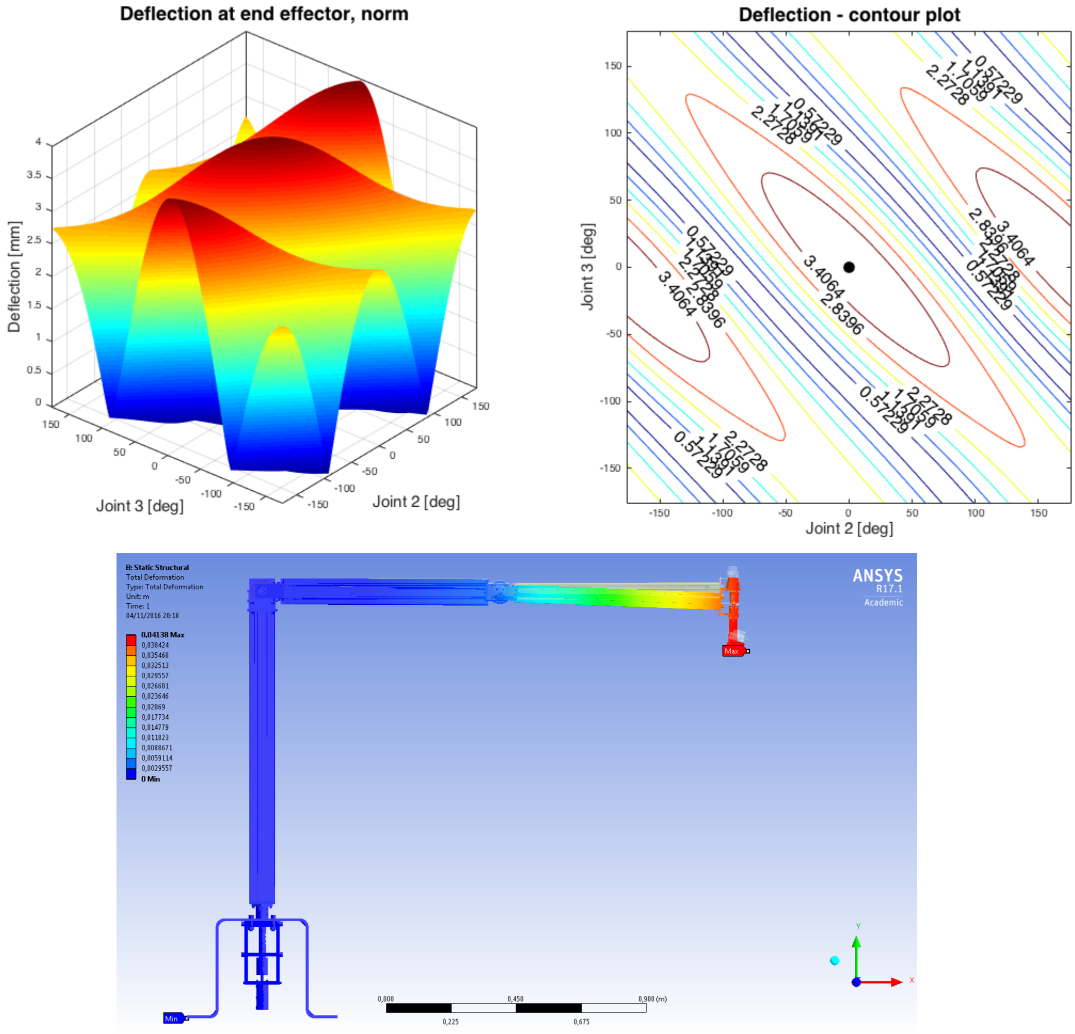


Figure 75: Loaded robot, $\theta_2$=$\theta_3$=0°, Matlab® and Ansys® model.

In this case, the angles for *joints* 2 and 3 are $\theta_2 = \theta_3 = 0^\circ$ respectively: the end effector is now loaded with a 1 kg payload. The comparison between the Matlab® model and the Ansys® simulation yielded the following results for the displacement:

| | | | |
|---|---|---|---|
| Matlab®: | 3.89 mm | | |
| Ansys®: | 4.14 mm | → | $\Delta = 6.4\%$ |

### 4.2.5 Case 2.1: loaded e. e., $\theta_2 = 50°$, $\theta_3 = -100°$, p/l = 1 $kg$

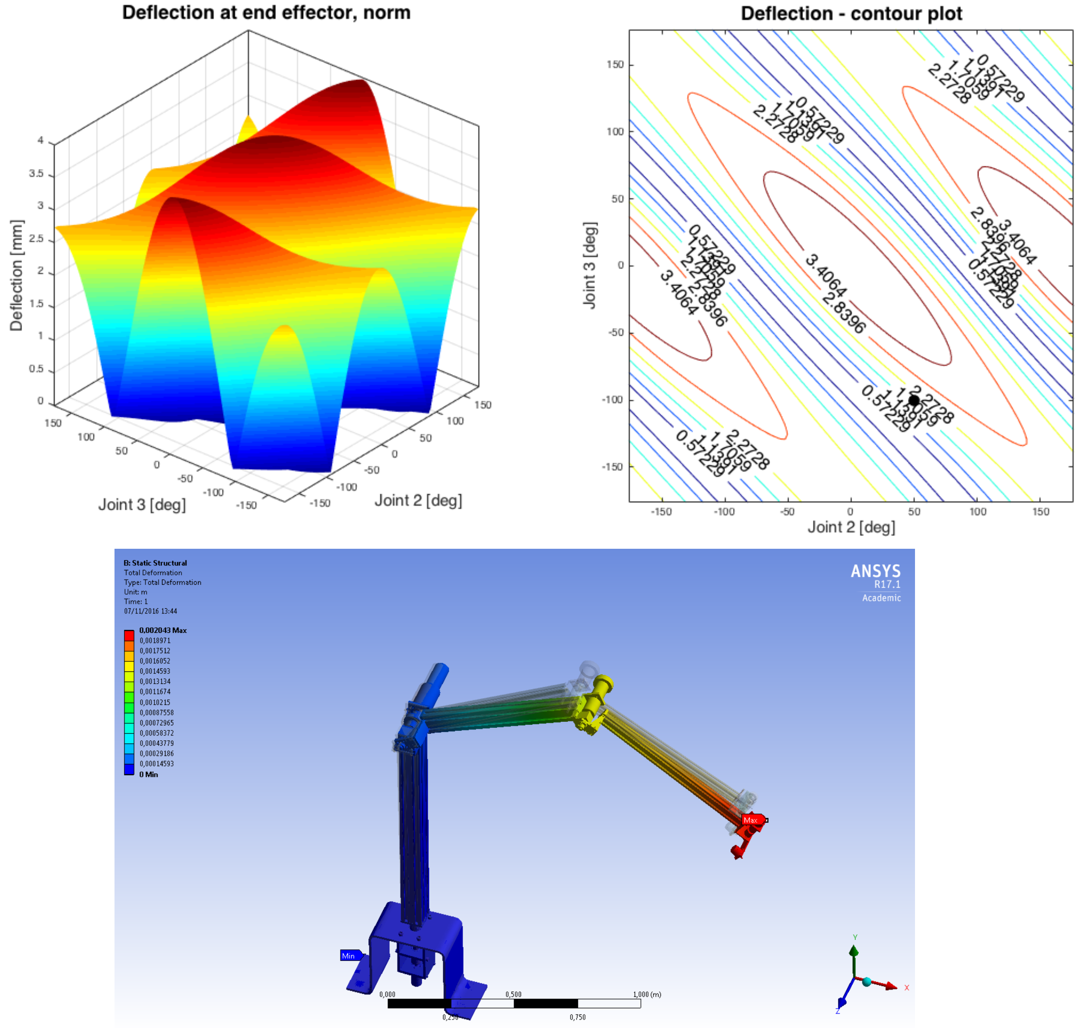


Figure 76: Loaded robot, $\theta_2 = 50°$, $\theta_3 = -100°$, MATLAB® and ANSYS® model.

In this case, the angles for *joints* 2 and 3 are $\theta_2 = 50°$, $\theta_3 = -100°$ respectively: the end effector is now loaded with a 1 kg payload. The comparison between the MATLAB® model and the ANSYS® simulation yielded the following results for the displacement:

$$\text{MATLAB}^{\circledR}: \quad 1.95\,\text{mm}$$

$$\text{ANSYS}^{\circledR}: \quad 2.04\,\text{mm} \quad \rightarrow \quad \Delta = 4.6\%$$

### 4.2.6 Case 3.1: loaded e. e., $\theta_2 = 80°$, $\theta_3 = -95°$, $p/l = 1\,kg$

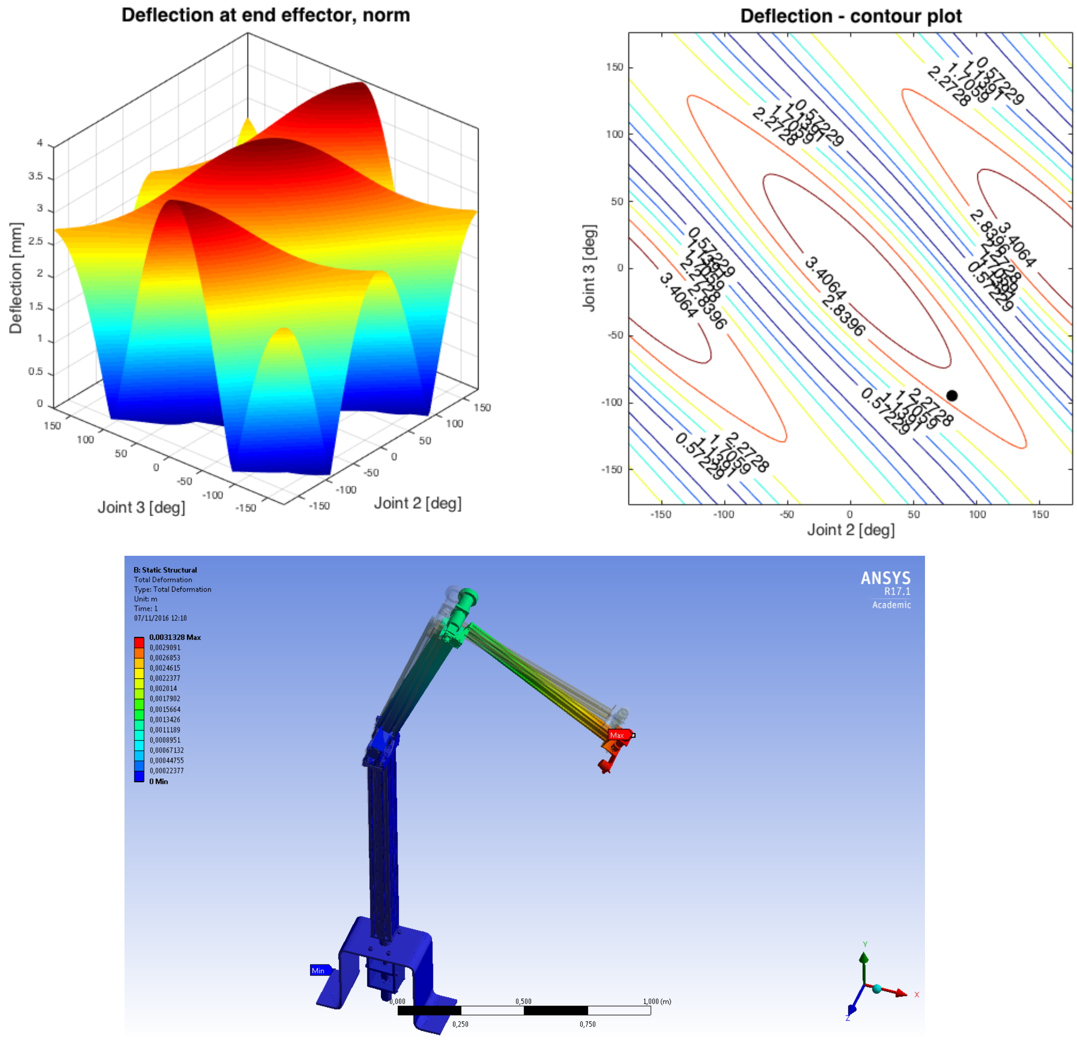


Figure 77: Loaded robot, $\theta_2 = 80°$, $\theta_3 = -95°$, Matlab® and Ansys® model.

In this case, the angles for *joints* 2 and 3 are $\theta_2 = 80°$, $\theta_3 = -95°$ respectively: the end effector is now loaded with a 1 kg payload. The comparison between the Matlab® model and the Ansys® simulation yielded the following results for the displacement:

| | | | |
|---|---|---|---|
| Matlab®: | 3.02 mm | | |
| Ansys®: | 3.13 mm | → | $\Delta = 3.6\%$ |

### 4.2.7 Static verification summary

The following table represents the percent error that the FEM model showed with respect to the MATLAB® simulation.

| | $\theta_2$ [deg] | $\theta_3$ [deg] | Load [kg] | MATLAB® [mm] | ANSYS® [mm] | Δ, % |
|---|---|---|---|---|---|---|
| Case 1.0 | 0 | 0 | NO | 2.74 | 2.89 | 5.5 |
| Case 2.0 | 50 | -100 | NO | 1.31 | 1.37 | 4.6 |
| Case 3.0 | 80 | -95 | NO | 2.03 | 2.12 | 4.4 |
| Case 1.1 | 0 | 0 | 1 | 3.89 | 4.14 | 6.4 |
| Case 2.1 | 50 | -100 | 1 | 1.95 | 2.04 | 4.6 |
| Case 3.1 | 80 | -95 | 1 | 3.02 | 3.13 | 3.6 |
| | | | | | Average Error | 4.9 |

Table 7: Static FEM analysis: results.

It can be seen that the error is on average 4.9%; in addition, it is possible to observe that the error is slightly larger when the robotic arm is fully extended. This discrepancy between the results can be traced down to the fact that the MATLAB® model doesn't take into account the contribution of the interface flexibility; moreover, this explanation is also in accordance with the fact that the error increases when the angles are zeros, condition for which the effect of the interface flexibility is at a maximum.

After the static analysis, which verified the correctness of the model and allowed to establish a realistic FEM framework, the software ANSYS® was used to investigate the modal response of the arm in different configurations.

## 4.3 MODAL ANALYSIS

This section contains the frequency response analysis of the system. The material is divided in three parts according to the following logic:

1. Firstly, a full 10 modes analysis is presented (Fig. 78–87). This provides a general idea of the vibration modes of the manipulator in the extended configuration.

2. Iterating the process in 1) for several configurations, a lookup table is created and an empirical model of the frequency response for the first 4 modes is generated by interpolating the discrete data.

3. Using the information of the analyses, slight mechanical modifications were performed in order to improve the performances in terms of frequency response.

The following pages, with the aid of printouts from ANSYS®, provide a detailed account of the mentioned steps; due to limited space, only a selection of the performed simulations is inserted.

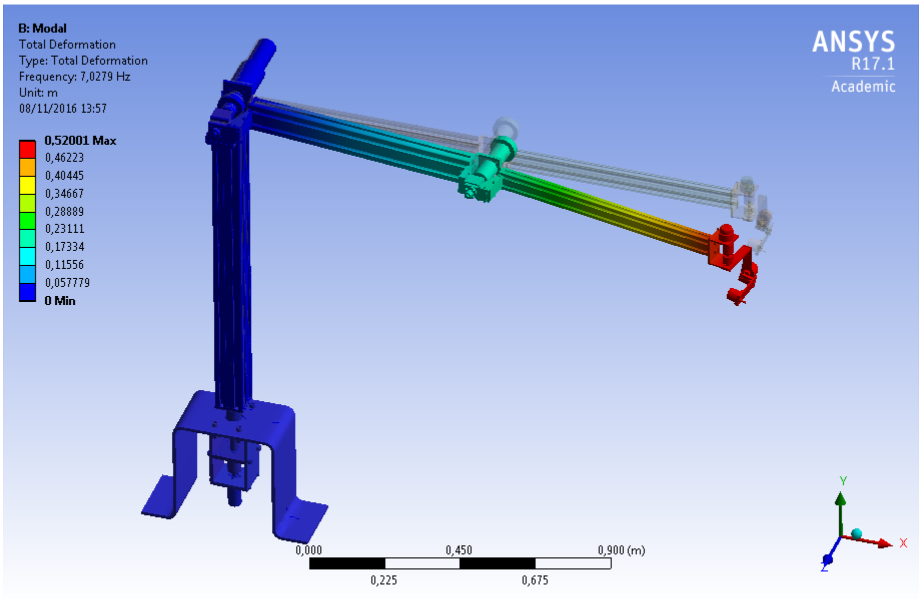


Figure 78: Modal analysis 1. Mode 1.

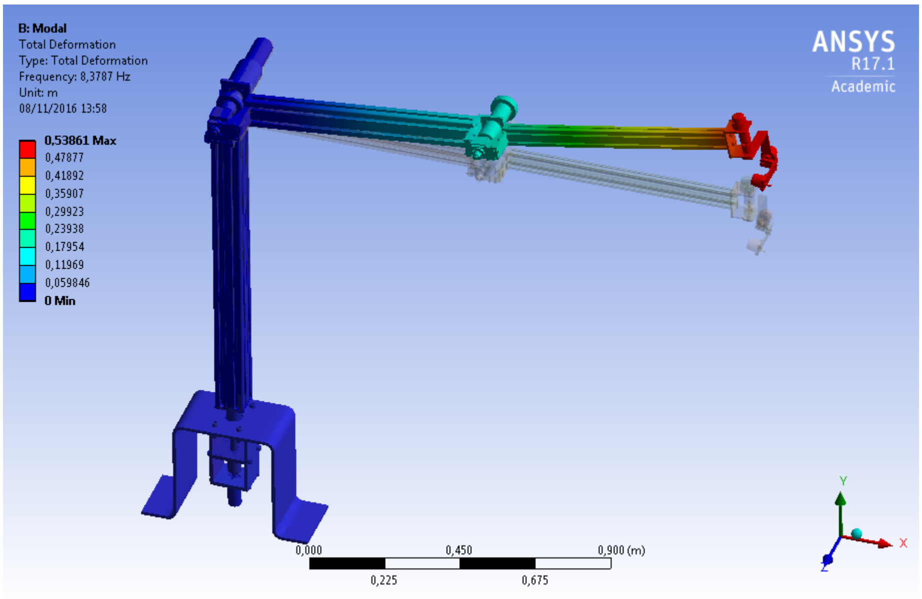


Figure 79: Modal analysis 1. Mode 2.

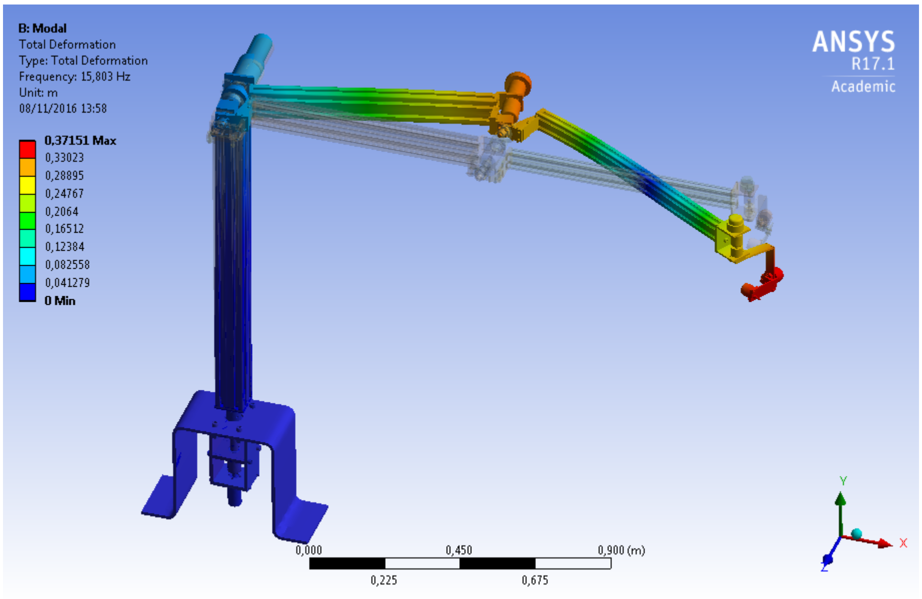


Figure 80: Modal analysis 1. Mode 3.

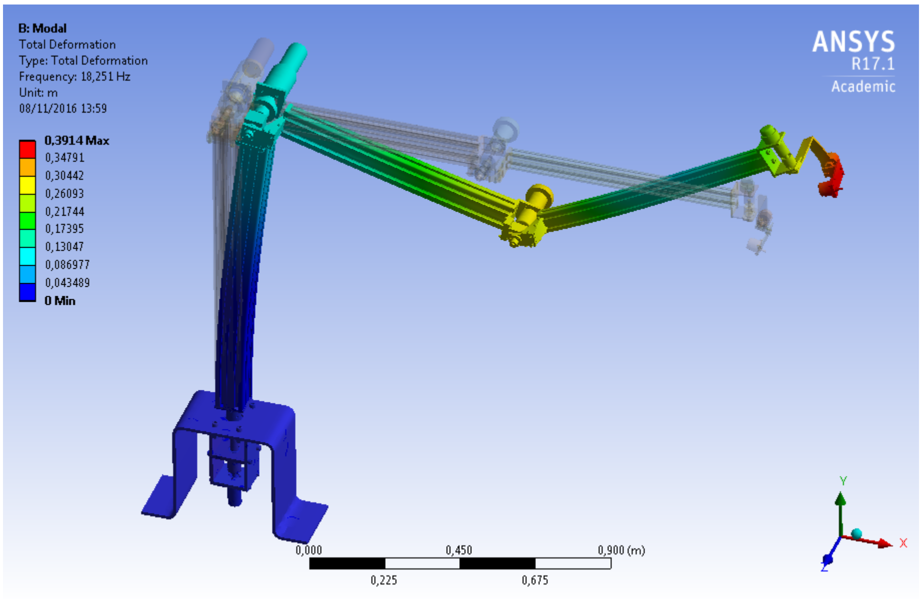


Figure 81: Modal analysis 1. Mode 4.

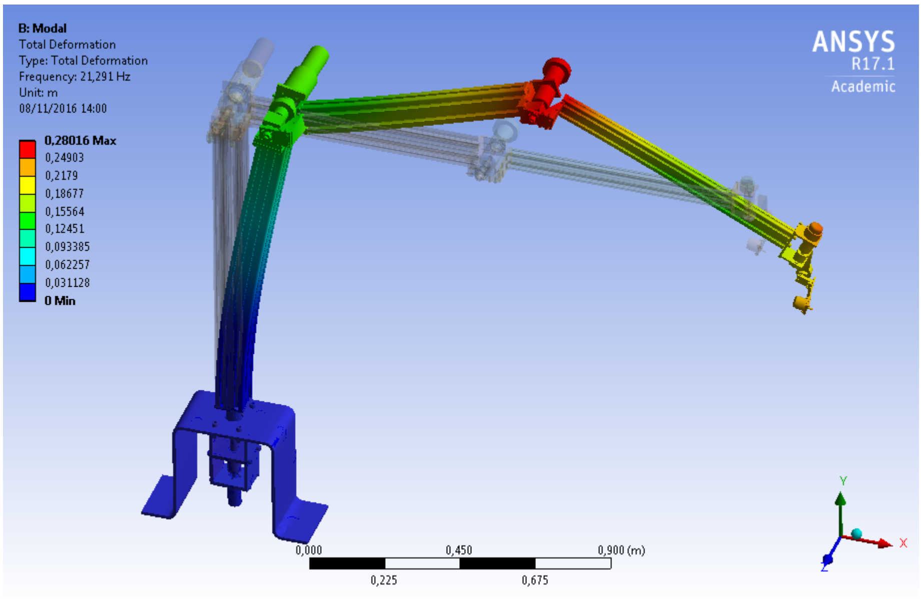


Figure 82: Modal analysis 1. Mode 5.

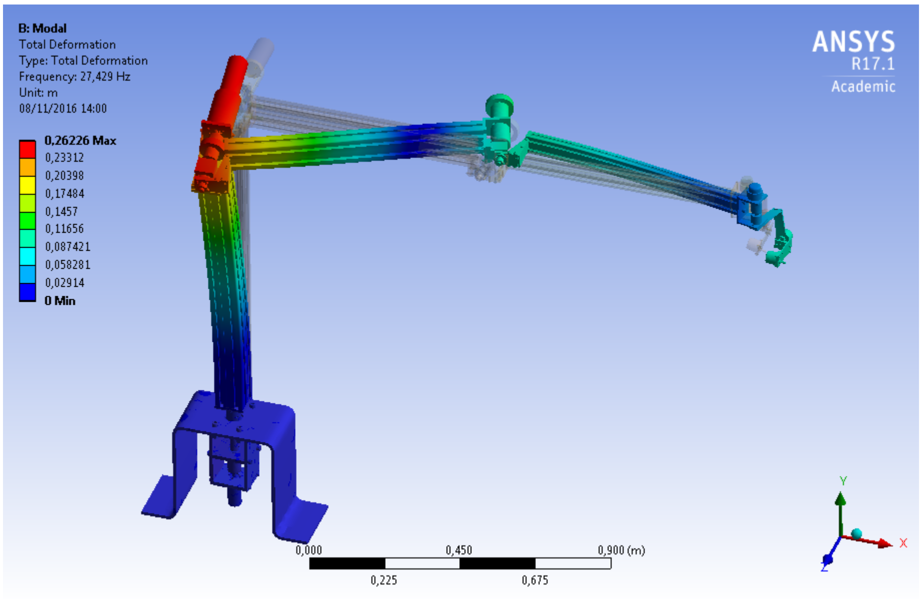
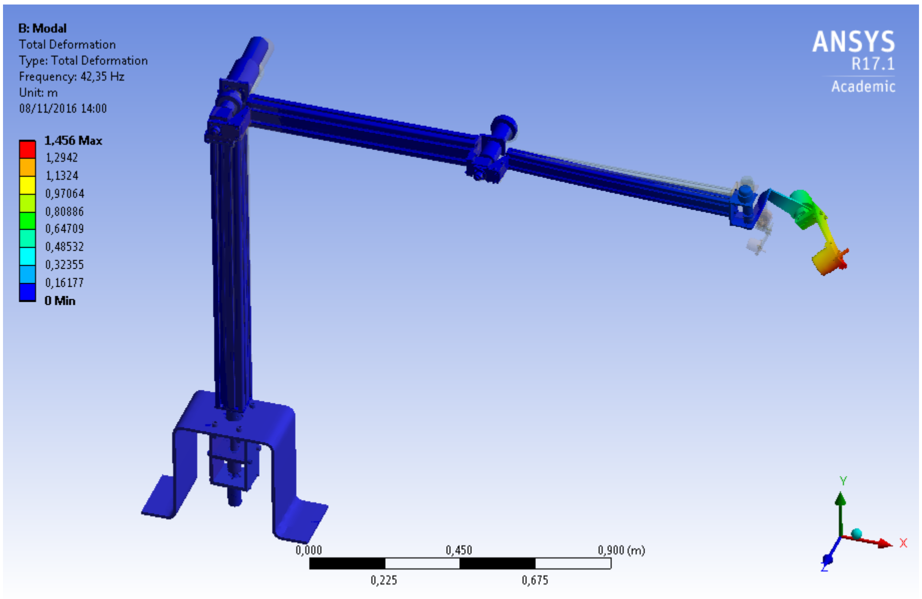
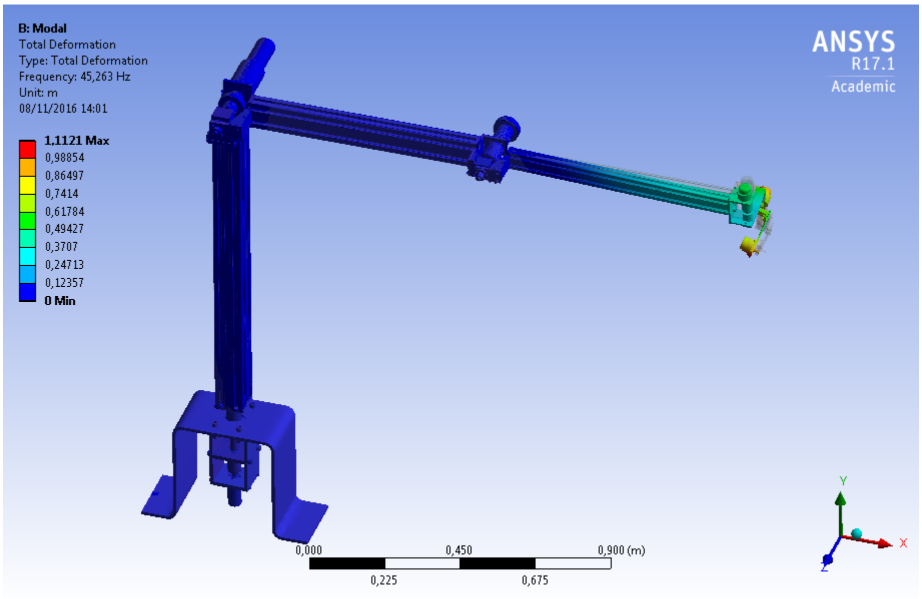


Figure 83: Modal analysis 1. Mode 6.



B: Modal
Total Deformation
Type: Total Deformation
Frequency: 42,35 Hz
Unit: m
08/11/2016 14:00

Figure 84: Modal analysis 1. Mode 7.



B: Modal
Total Deformation
Type: Total Deformation
Frequency: 45,263 Hz
Unit: m
08/11/2016 14:01

Figure 85: Modal analysis 1. Mode 8.

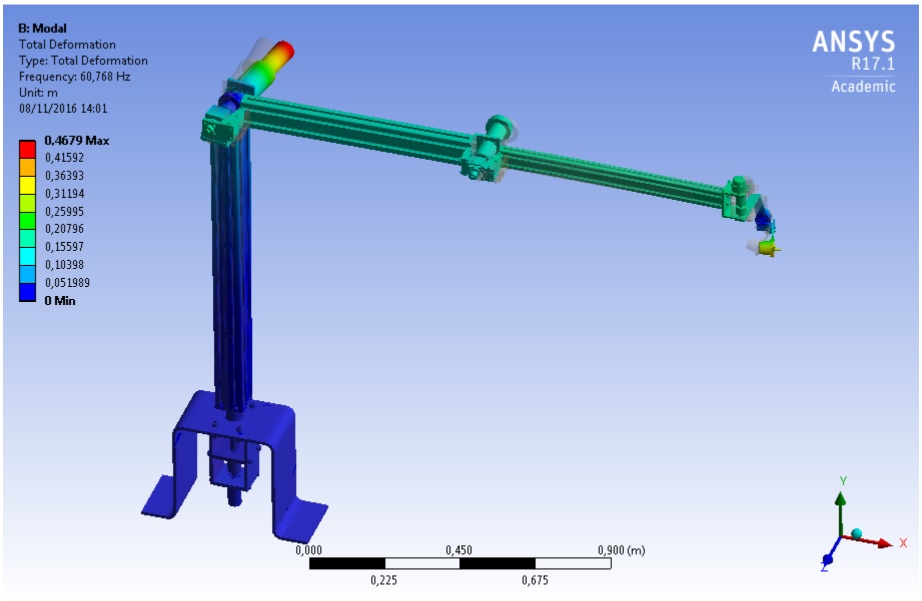


Figure 86: Modal analysis 1. Mode 9.

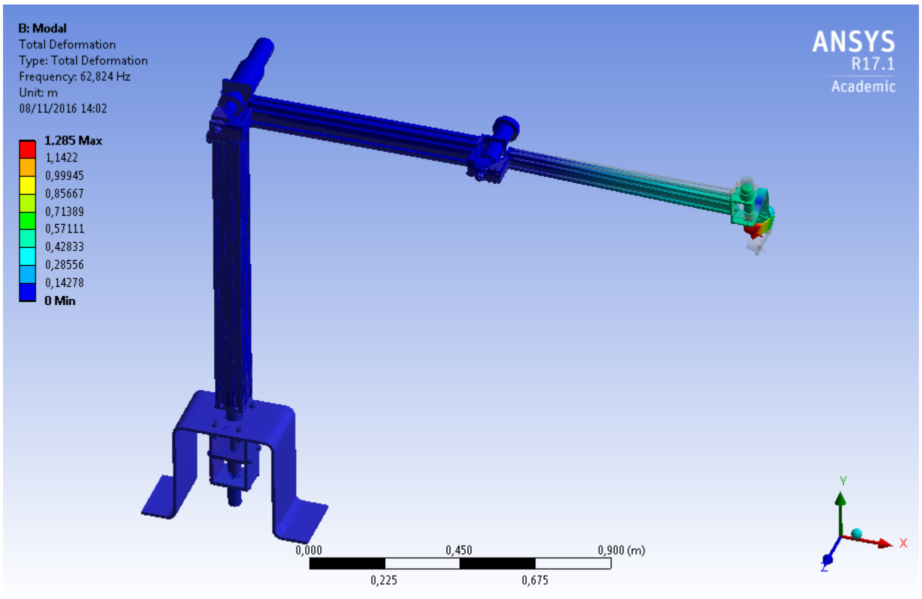


Figure 87: Modal analysis 1. Mode 10.

### 4.3.1 Lookup table for modal response

In order to design a continuous model of the frequency response, a lookup table has been created using some selected configurations that are likely to be used in a laboratory scenario and which are compatible with the manipulator angular limits.

The following table summarizes the frequency response of the first 4 vibrating modes of the system for 16 different angular configurations. The behaviors are represented in Fig. 88–91.

| $\theta_2$ [deg] | $\theta_3$ [deg] | **Mode 1** [Hz] | **Mode 2** [Hz] | **Mode 3** [Hz] | **Mode 4** [Hz] |
|---|---|---|---|---|---|
| 0 | 0 | 7.03 | 8.38 | 15.8 | 18.25 |
| 0 | -30 | 6.54 | 8.17 | 14.83 | 15.50 |
| 0 | -60 | 7.21 | 9.31 | 13.57 | 15.07 |
| 0 | -90 | 7.59 | 10.52 | 11.89 | 14.33 |
| 30 | 0 | 6.53 | 7.73 | 15.19 | 17.12 |
| 30 | -30 | 6.56 | 7.87 | 14.56 | 14.96 |
| 30 | -60 | 6.91 | 8.78 | 12.67 | 14.34 |
| 30 | -90 | 7.33 | 10.20 | 11.07 | 13.68 |
| 60 | 0 | 6.32 | 7.54 | 14.79 | 16.91 |
| 60 | -30 | 6.31 | 7.38 | 14.19 | 14.69 |
| 60 | -60 | 6.55 | 8.08 | 12.45 | 13.93 |
| 60 | -90 | 7.28 | 9.41 | 11.78 | 13.94 |
| 60 | 0 | 6.22 | 7.09 | 14.96 | 16.67 |
| 60 | -30 | 6.21 | 7.46 | 14.19 | 15.47 |
| 60 | -60 | 6.69 | 7.98 | 12.96 | 14.09 |
| 60 | -90 | 7.16 | 9.03 | 11.65 | 13.62 |

Table 8: Frequency response of the first 4 vibrating modes.

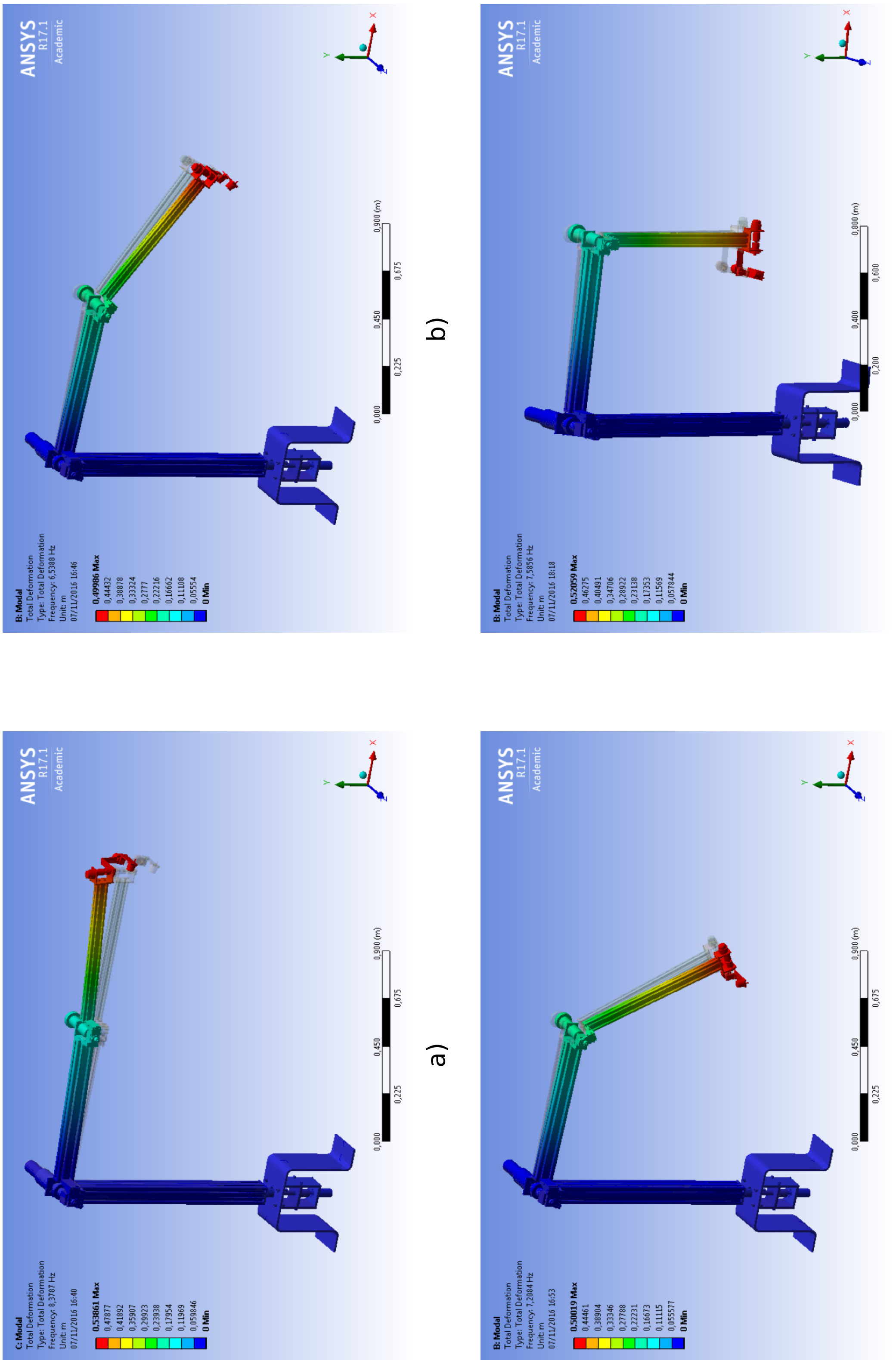


Figure 88: Modal analysis, 4 different configurations for $\theta_2 = 0°$

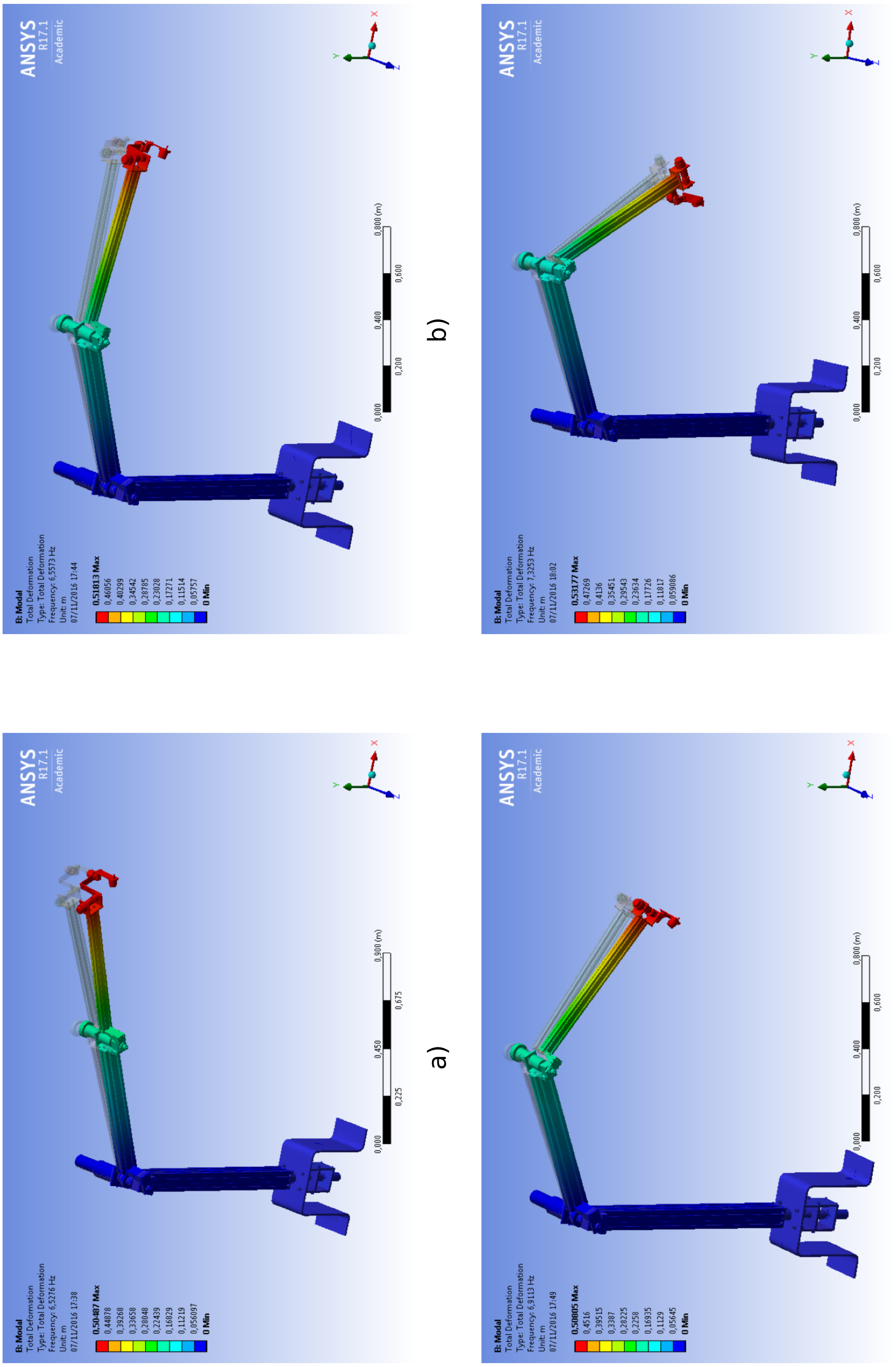


Figure 89: Modal analysis, 4 different configurations for $\theta_2 = 30°$

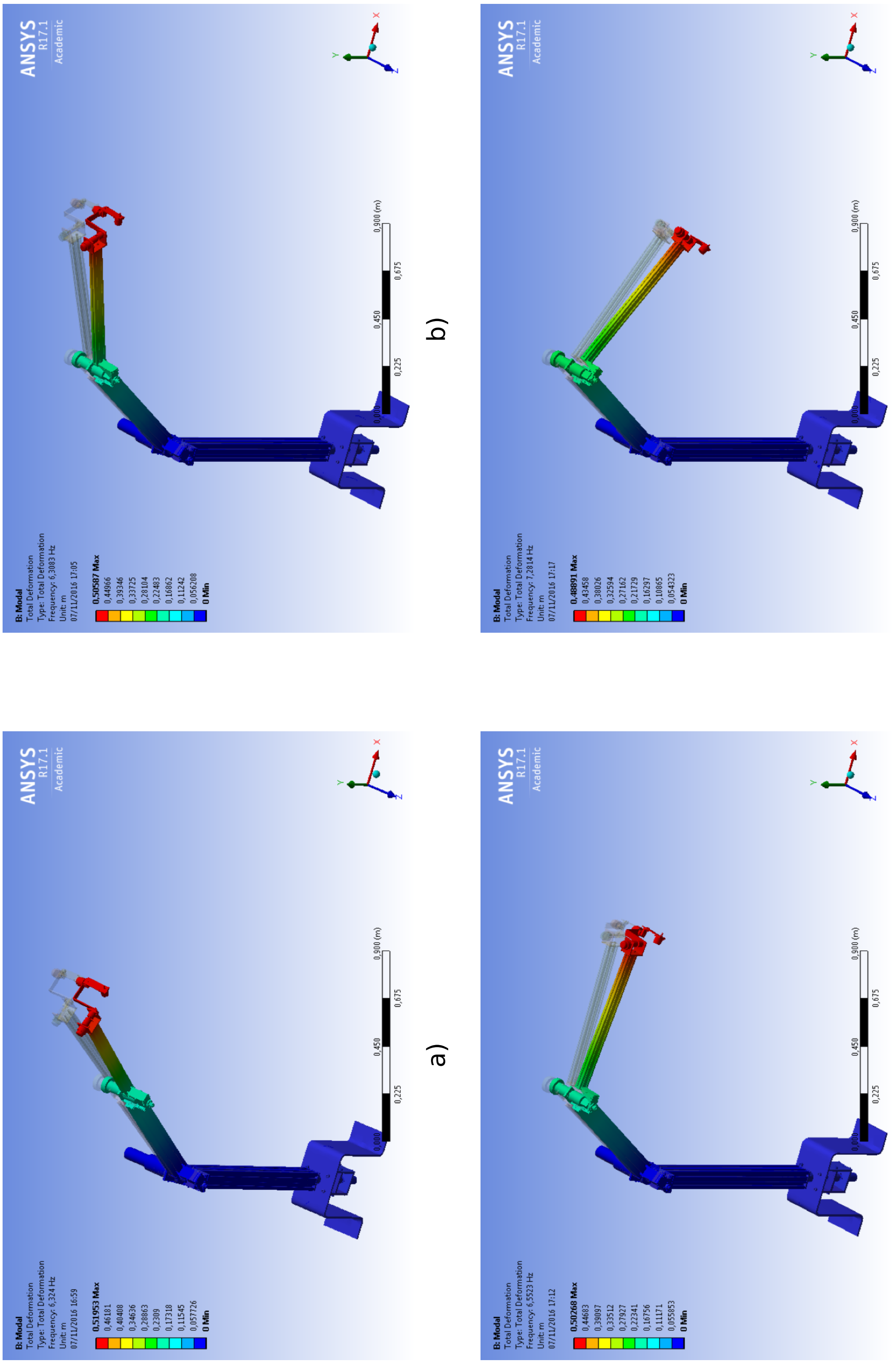


Figure 90: Modal analysis, 4 different configurations for $\theta_2 = 60°$

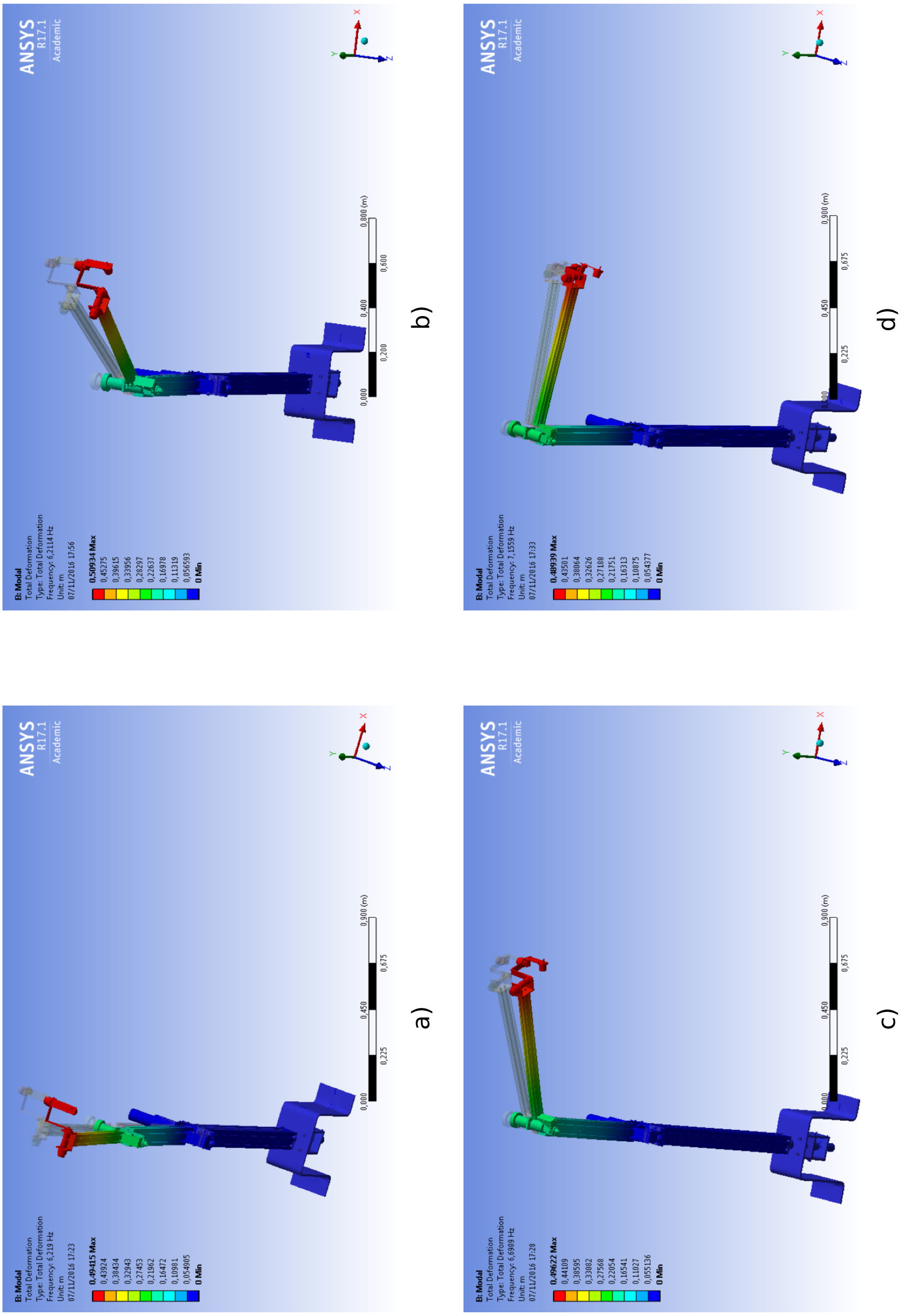


Figure 91: Modal analysis, 4 different configurations for $\theta_2 = 90°$

### 4.3.2 Analytical model of the configuration dependent modes

By using the data available in the previous chapter it is possible to use non-linear regression techniques to find a fitting equation in the independent parameters $\theta_2$ and $\Theta_3$ (where, for simplicity, we take $\Theta_3$ as the absolute value of $\theta_3$ in our angular convention). The expression that was found to best fit the simulation data for the calculation of the resonating frequency $\nu$ is:

$$\nu = a_1 \cdot {\Theta_3}^2 + a_2 \cdot \Theta_3 + a_3 \cdot \theta_2 + a_4 \tag{109}$$

where:

$$\begin{aligned} a_1 &= 1.714 \cdot 10^{-4} \\ a_2 &= -5.844 \cdot 10^{-3} \\ a_3 &= -5.922 \cdot 10^{-3} \\ a_4 &= 6.6766 \cdot 10^{0} \end{aligned}$$

The resulting model, even though it is obtained using a limited number of points, shows good performances in terms of goodness of fit, with the uncertainties of the parameters being:

$$\begin{aligned} \sigma_{a_1} &= 3.903 \cdot 10^{-5} \\ \sigma_{a_2} &= 3.665 \cdot 10^{-3} \\ \sigma_{a_3} &= 1.047 \cdot 10^{-3} \\ \sigma_{a_4} &= 8.312 \cdot 10^{-2} \end{aligned}$$

The resulting $R^2$ of the model is 0.9283. Ultimately, the analytical model, plotted against the actual simulated points, is presented in Fig. 92 (a). The residuals are expressed in terms of percentage error and are plot in Fig. 92 (b).

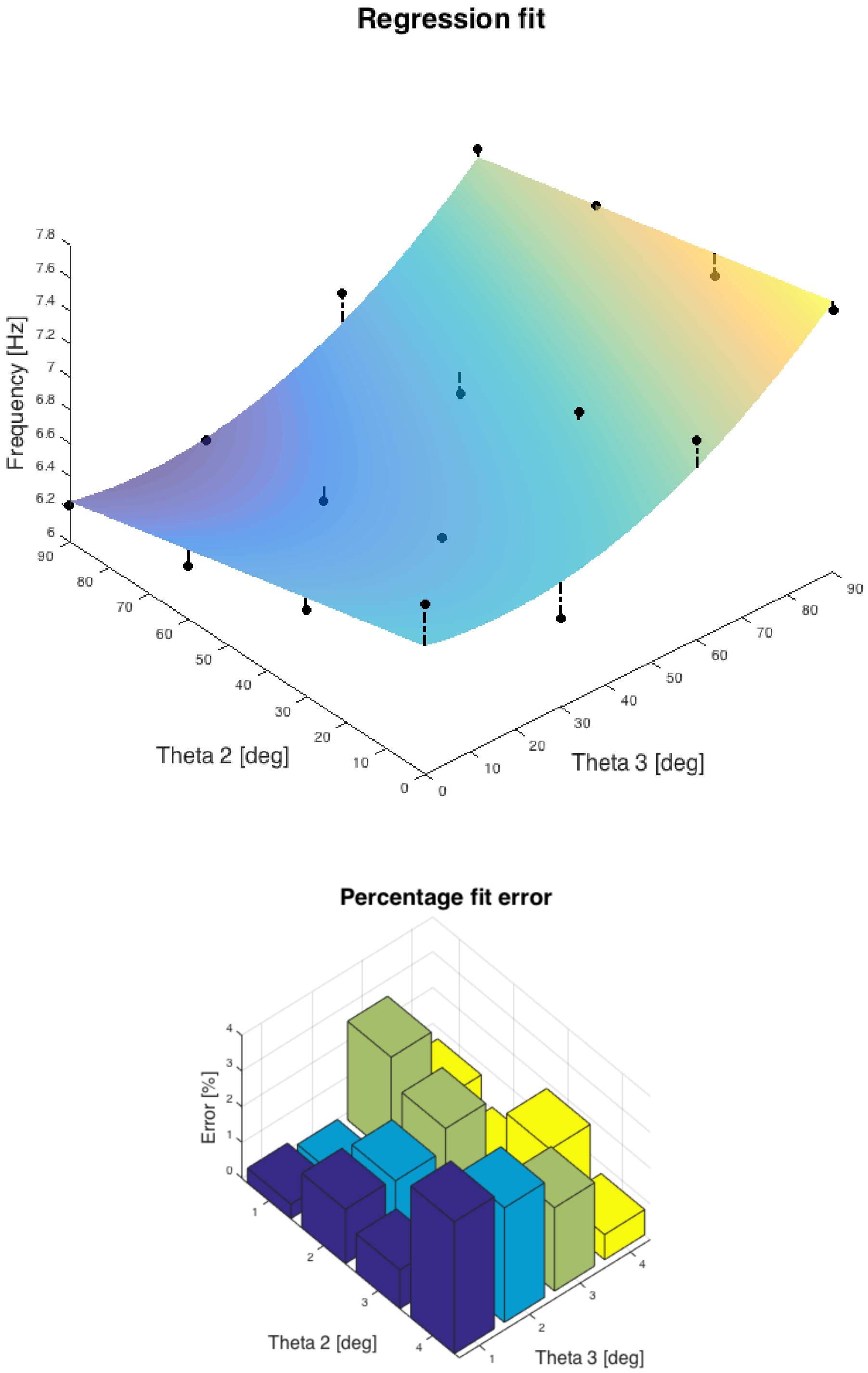


Figure 92: Regression model: fitting surface and percentage errors.

## 4.4 MECHANICAL MODIFICATIONS

From the modal analysis presented at the beginning of the chapter, it can be noted that the flexibility of the links is the cause of the first four modes. The characteristic of the joint blocks, on the other hand, have an effect on the remaining modes. Since it is not desirable at this point of the design process to modify the link design, the analysis was used to perform some mechanical modification to the joint blocks. In particular, the goal was to improve the rigidity of the system (in terms of the increase of the first modal response) with the tradeoff of not increasing the system's mass.

This resulted in the following minor mechanical modifications:

1. Interface joint block triangles for *joint* 2 and 3 have been augmented to a thickness of 2 mm from an original 0.5 mm

2. End effector sheet metal thickness has been increased from 1.5 mm to 2 mm

3. Diameter of the shaft connecting *link* 1 to the base joint has been changed to 25 mm from the original 20 mm

4. Additional fixtures have been added to the triangles connecting the base interface to the motor flange in *joint* 2 and 3, from an original number of 4xM5 to 8xM6

The immediate effect of this change was the improvement of the modal response related to the joint blocks, but this modification influenced also the first modal frequency of the entire system. In the following tables it is possible to appreciate the improvement in the modal profiles for the cases with $\theta_2 = 0$. The results are compared with the original design. Fig. 93–96 depict the simulations for these selected cases.

Modal response pre-modifications

| $\theta_2$ [deg] | $\theta_3$ [deg] | **Mode 1** [Hz] | **Mode 2** [Hz] | **Mode 3** [Hz] | **Mode 4** [Hz] |
|---|---|---|---|---|---|
| 0 | 0 | 7.03 | 8.38 | 15.8 | 18.25 |
| 0 | -30 | 6.54 | 8.17 | 14.83 | 15.50 |
| 0 | -60 | 7.21 | 9.31 | 13.57 | 15.07 |
| 0 | -90 | 7.59 | 10.52 | 11.89 | 14.33 |

Modal response post-modifications

| $\theta_2$ [deg] | $\theta_3$ [deg] | **Mode 1** [Hz] | **Mode 2** [Hz] | **Mode 3** [Hz] | **Mode 4** [Hz] |
|---|---|---|---|---|---|
| 0 | 0 | 8.37 | 9.79 | 29.96 | 38.06 |
| 0 | -30 | 9.44 | 11.14 | 32.04 | 42.01 |
| 0 | -60 | 9.77 | 11.12 | 32.18 | 39.89 |
| 0 | -90 | 9.31 | 10.89 | 31.84 | 38.71 |

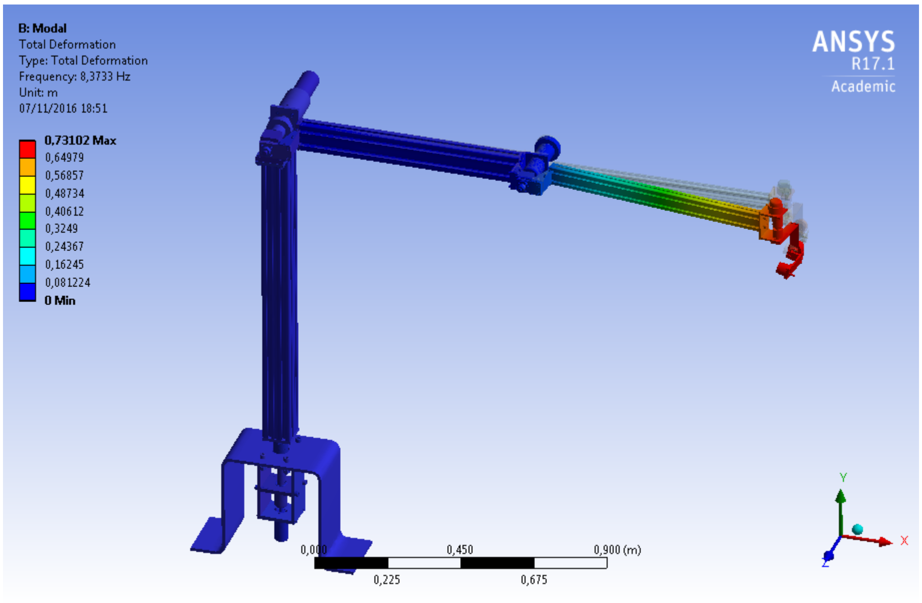


Figure 93: Reinforced structure, configuration $\theta_2 = 0$, $\theta_3 = 0$. Mode 1.

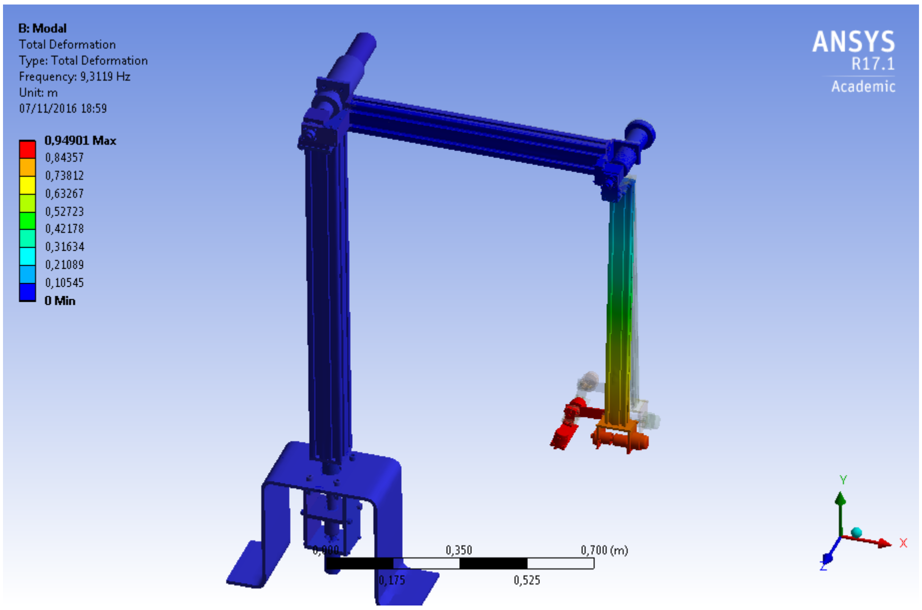


Figure 94: Reinforced structure, configuration $\theta_2 = 0$, $\theta_3 = -30$. Mode 1.

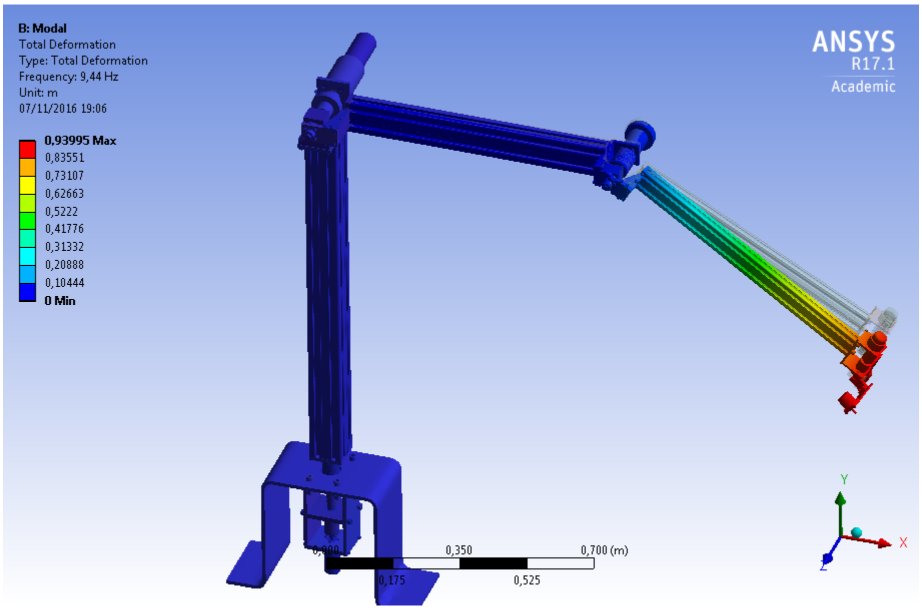


Figure 95: Reinforced structure, configuration $\theta_2 = 0$, $\theta_3 = -60$. Mode 1.

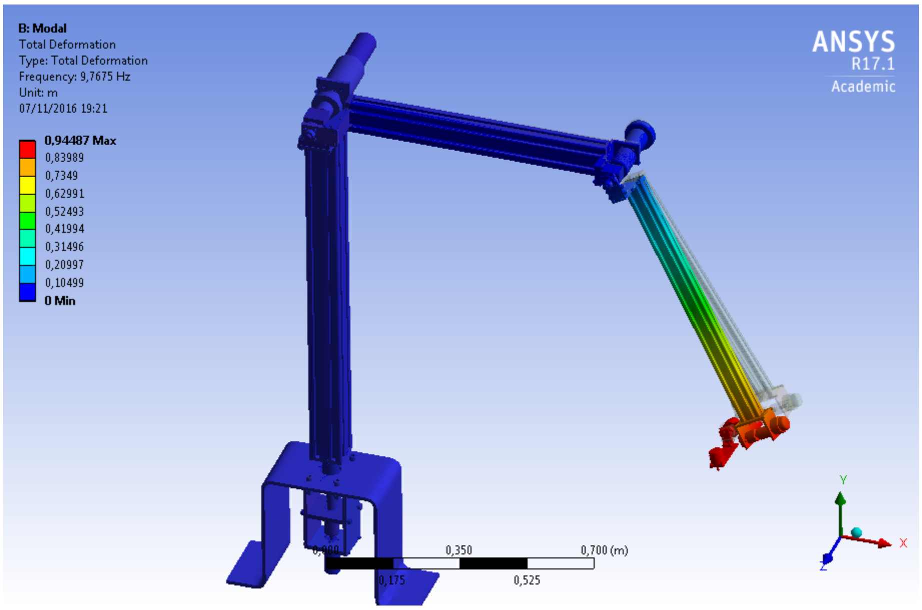


Figure 96: Reinforced structure, configuration $\theta_2 = 0$, $\theta_3 = -90$. Mode 1.

# 5 INSTRUMENTING THE END EFFECTOR FOR REALISTIC ORBITAL OPERATIONS

## 5.1 EMBEDDING AN ATTITUDE DETERMINATION SYSTEM ON THE END EFFECTOR

Following the mechanical design chapter, the focus will be now directed on the assessment of the capabilities of the end effector.

State of the art solutions in terms of robotic facilities for the simulation of OOS do not comprise sensors for the determination of attitude. The EPOS experiment from DLR, for example, provides realistic simulations of environmental conditions (e.g. sun illumination effects) but does not currently perform self-attitude determination.

Attitude information, that need to be used during the manipulator's maneuver, can be provided to the algorithm in two ways: via software or via hardware. In this application, the hardware-in-the-loop solution was chosen and was believed to be the most appropriate in order to have a realistic system with its realistic cohort of related peculiarities (sensing noise, delay, outliers measures). Having a real system for the measurement of the attitude allowed for the further development of the Cross Entropy theory detailed in Chapter .

The robotic facility presented in the previous chapter has been designed with the goal of performing not only realistic OOS, but to serve as a benchmark for navigation and docking technologies, both software and hardware. Our team at CISAS Research Center from the University of Padova has provided a consistent contribution to this research field in recent years [23, 24, 25, 26, 27].

The technology presented in this chapter has been designed to fill a gap in the CubeSat attitude determination sensors segment, and which ultimately led to the manufacturing of a sensor that will enlarge the capabilities of the final robotic facility.

## 5.2 DEVELOPMENT OF A SUN SENSOR FOR CUBESATS

In the flourishing small satellites market various solutions for sun sensors are available, most of them derived and miniaturized from larger satellites hardware. Such components are therefore not totally optimized for small spacecraft, and present heavy requirements and high cost (up to several thousands of Euro) that are not always balanced with good accuracy and precision. In particular, one of the best off-the-shelf sensors reaches an accuracy of 0.3°and a precision of 0.05°, with a size of 40x30 mm and a weight of 25 g [28].
A simplified classification of sun sensors solutions is here briefly reported: 1-D sensors are able to give a single angular information regarding the sun direction, and their measure can be performed with an analog [29] or a digital system [30, 31] with a theoretical resolution of up to 0.07°[31]. To obtain a complete information of the sun relative position (i.e. the sun vector) it is possible to use two 1-D elements or to implement 2-D sensors, usually employing a photo-sensible surface instead of a linear array of photodiodes. Their working principle is simple: Sun rays passing through a mask mounted in front of the photo-sensible sensor illuminate a spot that can be detected, allowing the reconstruction of the Sun vector [32, 33]. In the cited case, average accuracy can reach less than 0.01°, but with a mass

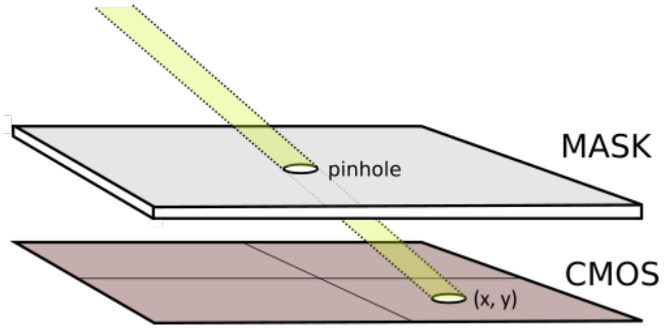


Figure 97: Sensor working principle: Sun rays passing through the mask illuminate a spot on the CMOS surface.

of 2 kg [34].
In the framework of its investigation on CubeSat technologies, the University of Padova is developing a cohort of sensors for absolute and relative attitude and position determination. To this day, relative navigation sensors have been developed, and a new sun sensor is under investigation [35] The idea behind the proposed device is to have a small yet precise attitude sensor which can be place on a CubeSat with a very limited footprint. The main philosophy behind the project is to use off-the-shelf components and custom software to obtain a reliable piece of equipment that could be feasible for a multitude of applications, from miniaturized commercial spacecraft to academic demonstrators.

The layout and the working principle of the proposed sensor are visible Fig. 97: the sun rays can filtrate through a circular hole on the front mask, to be detected by an active pixel sensor (based CMOS technology) behind it. By knowing the coordinates of the spot on the CMOS and the sensor geometry, it is possible to use simple trigonometry in order to infer the sun vector in terms of the azimuth and elevation angles with respect to the CMOS plane.

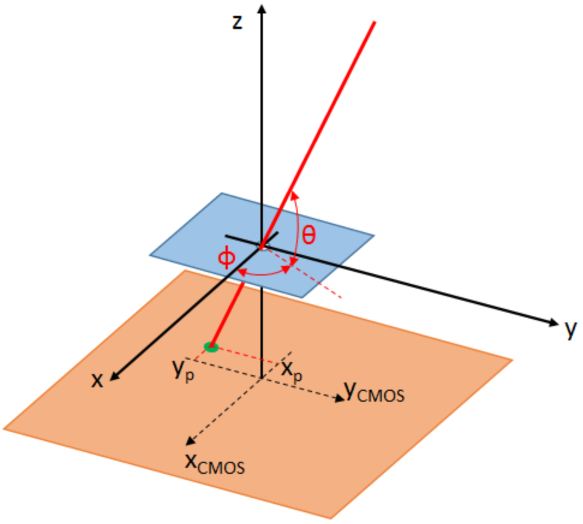


Figure 98: Sensor reference geometry: the Sun vector with respect to the reference plane XY is defined by azimuth Φ and elevation $\alpha$. The illuminated spot centre coordinates is $x_p$, $y_p$.

## 5.3 geometrical model and simulations

In Fig. 98 the geometrical model of the proposed sensor is reported, with the reference frames on the mask (blue) and the CMOS-plane (orange). The direction of the Sun vector (i.e the relative direction of the Sun in the field of view, in red in Fig. 98) can be defined both with the two angles of azimuth Φ and elevation $\alpha$ or with the related unit vector $\mathbf{v} = (\cos\Phi\cos\alpha, \sin\Phi\cos\alpha, \sin\alpha)$. Both formulations can be derived knowing the light spot position $(x_{p,CMOS}, y_{p,CMOS})$ with respect to the CMOS centre, as the distance $h$ between the mask and the CMOS is noted:

$$\mathbf{v} = \frac{(x_{CMOS}, y_{CMOS}, h)}{\sqrt{{x_{CMOS}}^2 + {y_{CMOS}}^2 + h^2}} \tag{112}$$

The formulation reported in Eq. 112 has the advantage of not involving any trigonometric function; furthermore, an unique and real solution exists for any position of the light spot on the CMOS. Knowing the size

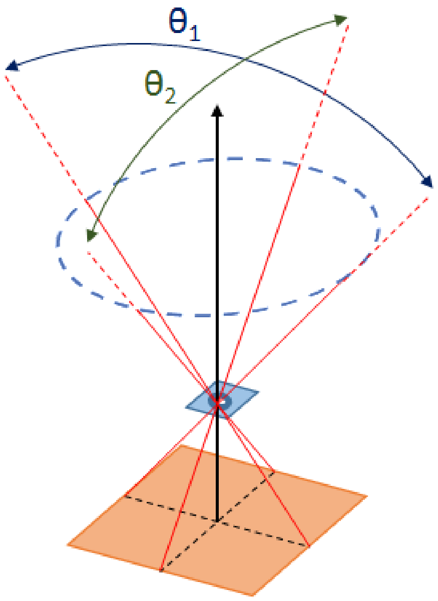


Figure 99: The sensor field of view, represented by the dashed blue ellipsoid, can be described by the two view angles $\theta_1$ and $\theta_2$.

of the CMOS and the mask mounting distance it is possible to define the sensor theoretical field of view, considering an ideal mask with negligible thickness and no diffraction. For the sensor described in this work, the two field of view angles $\alpha_1$ and $\alpha_2$ represented in Fig. 99 are respectively of 66.2°and 51.1°.

### 5.3.1 Simulations

In this section the aforementioned geometrical description is implemented to develop three different sensor models of increasing sophistication, as reported in Fig. 101.

In the ideal case of a mask with negligible thickness (Fig. 101, left), the projected light spot has the exact size and shape of the mask hole: the Sun vector can be directly calculated by measuring the light spot centre. Due

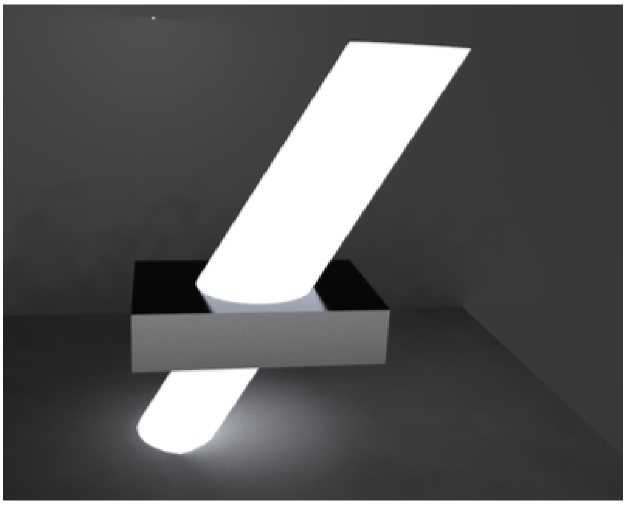


Figure 100: Effect of the mask thickness on the light spot: the incoming sunlight can be partially obstructed by the mask

to miniaturization constraints, the employed mask thickness is comparable to the hole diameter and part of the incoming radiation is stopped by the mask border (Fig. 100), modifying the shape of the light spot detected by the CMOS (Fig. 101, centre). Comparing the new shape with the circle from the previous model (no thickness effect), it is possible to note a translation of the light spot centre.

The geometrical problem, reported in Fig. 102, can be simplified due to its axial-symmetry. The centers of the two light-spot circles sections $(x_0, y_0)$ (i.e. the negligible thickness light-spot circle center, caused by the mask upper surface shading) and $(x_1, y_1)$ (caused by the mask lower surface shade) can be defined knowing the two angles $\phi$ and $\theta$, as well as the distance $h$ between the mask and the CMOS:

$$\begin{cases} x_0 = (t+h) \cdot \tan(\pi/2 - \theta) \cdot \cos(\phi + \pi) \\ y_0 = (t+h) \cdot \tan(\pi/2 - \theta) \cdot \sin(\phi + \pi) \end{cases}$$

$$\begin{cases} x_1 = h \cdot \tan(\pi/2 - \theta) \cdot \cos(\phi + \pi) \\ y_1 = h \cdot \tan(\pi/2 - \theta) \cdot \sin(\phi + \pi) \end{cases}$$

The two points distance from the CMOS reference frame origin is:

$$\begin{cases} R_{C0} = \sqrt{x_0^2 + y_0^2} = (t+h)\tan\left(\frac{\pi}{2} - \theta\right) \\ R_{C1} = \sqrt{x_1^2 + y_1^2} = h\tan\left(\frac{\pi}{2} - \theta\right) \end{cases}$$

The distance between the two arcs centers can therefore be defined as reported:

$$\Delta C = h\tan\left(\frac{\pi}{2} - \alpha\right)$$

The piecewise equation describing the shape of the projected sunspot, in polar coordinates (further expressed in terms of pinhole diameter $d$, azimuth *Phi* and elevation *Theta*), where $r = x^2 + y^2$ is:

$$\begin{cases} R_{C0}^2 - 2R_{C0}(x\cos\Phi + y\sin\Phi) + x^2 + y^2 = d^2/4 \\ \qquad for \;\; R_{C0} - \frac{d}{2} < r < R_{C0} + \frac{\Delta C}{2} \\ \\ R_{C1}^2 - 2R_{C1}(x\cos\Phi + y\sin\Phi) + x^2 + y^2 = d^2/4 \\ \qquad for \;\; R_{C1} - \frac{\Delta C}{2} < r < R_{C1} + \frac{d}{2} \end{cases}$$

Both of the aforementioned models have been developed using an ideal punctiform light source at near-infinite distance: the incident rays have been simulated as parallel. However, in near-Earth orbit and on ground the Sun angular diameter is not negligible and presents an incoming radiation aperture of about 0.5°. The third model considers such effect, yielding the results visible in Fig. 101 on the left: the spot is consistently larger than the previous one, and the center translation with respect to the first model is still visible.

The centre translation due to the aforementioned effects can be evaluated: the maximum bias is of about 5 pixels, which is equivalent to 0.12°; the sensor's software will be designed to evaluate the translation and calculate the right value of the Sun vector.

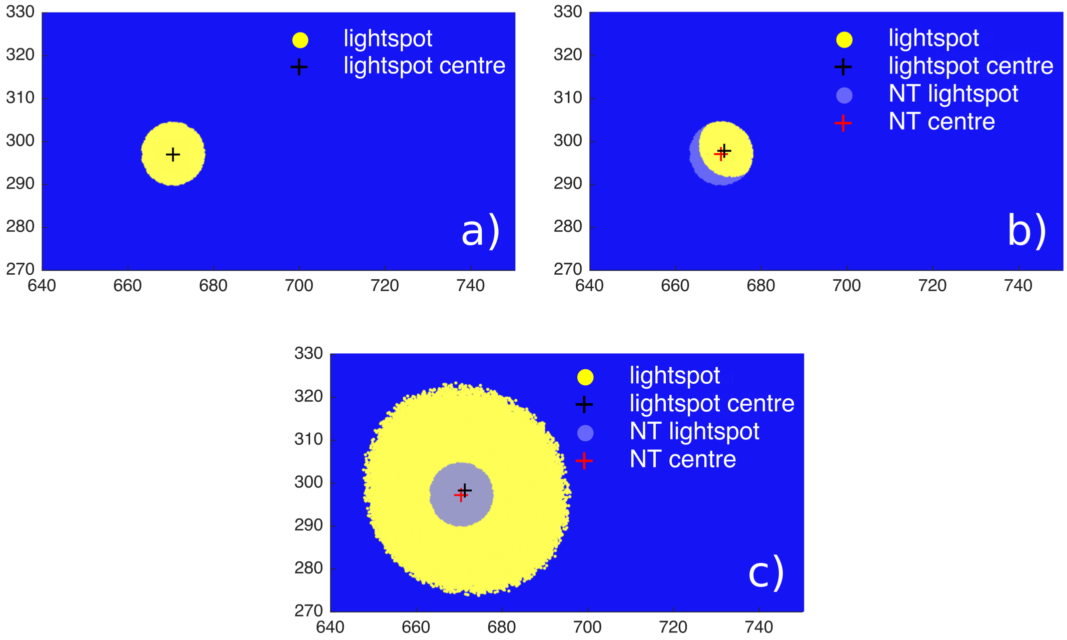


Figure 101: Sun sensor simulations, the three different models: from left to right, the spot projection considering (a) no mask thickness, (b) mask thickness, and (c) the effect of the Sun angular diameter with respect to a punctiform origin.

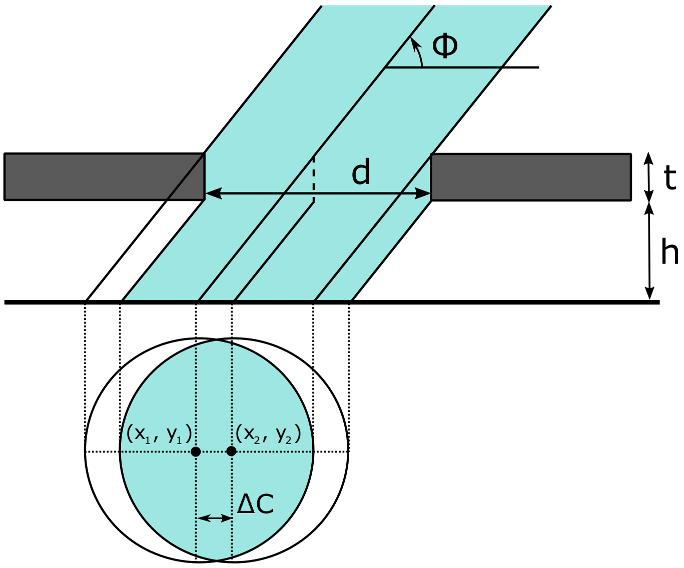


Figure 102: Schematic representation of the effects that the mask thickness has in the perturbation of the projected light spot.

### 5.3.2 Resolution

The sensor resolution is driven by the pixel size, which, for the chosen sensor, is 1.4 $\mu$m. A simulation has been performed in order to get an insight of the performances of the system across the CMOS surface. The angular accuracy, expressed in polar coordinates, is:

$$a(r, \alpha) = a(r) \doteq tan^{-1}\left(\frac{r}{t+h}\right) - tan^{-1}\left(\frac{r-px}{t+h}\right) \tag{113}$$

where $h$ is the CMOS-mask distance, $r$ is the distance from the pinhole center, $px$ is the size of a pixel, $t$ is the thickness of the mask. Fig. 103 depicts the resolutions expressed in terms of arcminutes. It can be seen that the maximum resolution uncertainty occurs in the proximity of the center of the sensor (under the hypothesis that this is collinear with the pinhole center). The maximum value is 1.55 arcmins, which, with the current setup, corresponds to a maximum resolution uncertainty of 0.023°. This value could be eventually lowered by decreasing the distance between the sensor and the CMOS and by using a sensor with a smaller pixel size.

## 5.4 EXPERIMENTAL SETUP

When it comes to active pixels sensors, two choices are available: CCDs and CMOS. Nowadays, CMOS represent the commercial standard, they are less expensive and have a limited power consumption when compared to CCDs. On the other hand, CCDs have better SNR profiles and are much simpler to handle and to interface with the acquisition hardware. For this sensor, we chose a commercial camera module based on CMOS technology (Raspberry Pi Camera Module), whose characteristics (in terms of pixel size and footprint) are suited to our application.

The experimental setup is composed by:

- acquisition PC (2.4 GHz i5, 8GB RAM)
- interfacing board (Raspberry PI 2 Model B)
- CMOS sensor (Raspberry PI Camera Module)
- pinhole aperture (Edmund Optics)
- enclosing hardware (3D printed)

This type of camera, however, is commercially distributed with a mounted lens. In order for the device to be used, the optics was removed, leaving an exposed CMOS. Then, by using a 3D printed custom adapter, the pinhole was placed on top of the CMOS, at a distance of 3 mm from the active surface. The pinhole, which is a commercial device manufactured by Edmund Optics, consists of a thin metal film (25 $\mu$m) on which a circular hole of diameter 20 $\mu$m has been etched (Fig. 106). The circularity and size of the hole has been verified through microscopy. The complete assembly is pictured in Fig. 104.

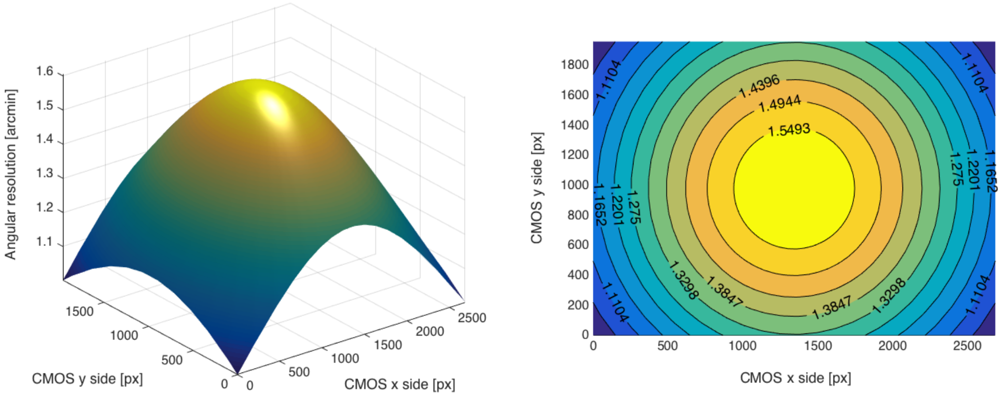


Figure 103: Resolution characteristics of the CMOS sensor under analysis. Three dimensional plot and contour line plot.

### 5.4.1 Precision

In order to obtain the metrological characteristics of the system, several tests were performed. The precision of the sensor has been obtained experimentally in order to get an insight of the reproducibility. To simulate in a laboratory environment the emittance of the Sun, we used an array of high power white LEDs arranged in a square pattern, whose emittance was computed to be close to $1200\ \mathrm{W/m^2}$ in the area invested by the light. By using such a device, placed 2 meters from the pinhole aperture, perpendicular to the CMOS-plane, we took 300 measures to investigate the accuracy of the sensor, Fig. 107: the scatter plot has been offset to the mean value of the acquisition and 90%, 95% and 99% confidence ellipses have been plotted.

By analyzing the eigenvectors of the covariance matrix scaled by the square root of the corresponding eigenvalue we can state that there is no evident correlation between the two pixel axes. The 99% confidence ellipse has semi axis lengths of [1.21; 1.14] px. This corresponds to a precision, translated into meters, of [1.69; 1.59] $\mu$m in the x-y directions respectively. In terms of angular displacement, this means an average precision

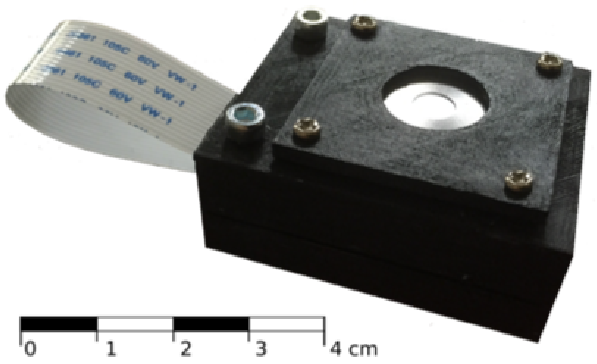


Figure 104: Sun-sensor prototype.

of ±0.024°.

### 5.4.2 Acquisition strategy

The sunspot projection on the CMOS sensor constitutes a very small footprint with respect to the entire sensing area. In addition, for a non-spinning satellite, the position of the sunspot can be considered quasi-fixed from one image acquisition to the next. This suggests for the implementation of a tracking algorithm in the image processing, in order to optimize the resources and boost the performances. In the case of a spinning satellite, this observation still holds if the image analysis is corroborated with information regarding the vehicle's attitude.

We assume the CMOS pictures come in a row-major format, that is, each pixel lies in memory next to its left and right neighbours, whereas the top and bottom neighbours are one row width of pixels away. Since memory bandwidth represents the limiting factor for simple operations on data, it's always important to minimize the times new memory is fetched: data in the CPU registers can be directly used for calculations, whereas data in the RAM may take several hundred of multiplications worth of CPU time in

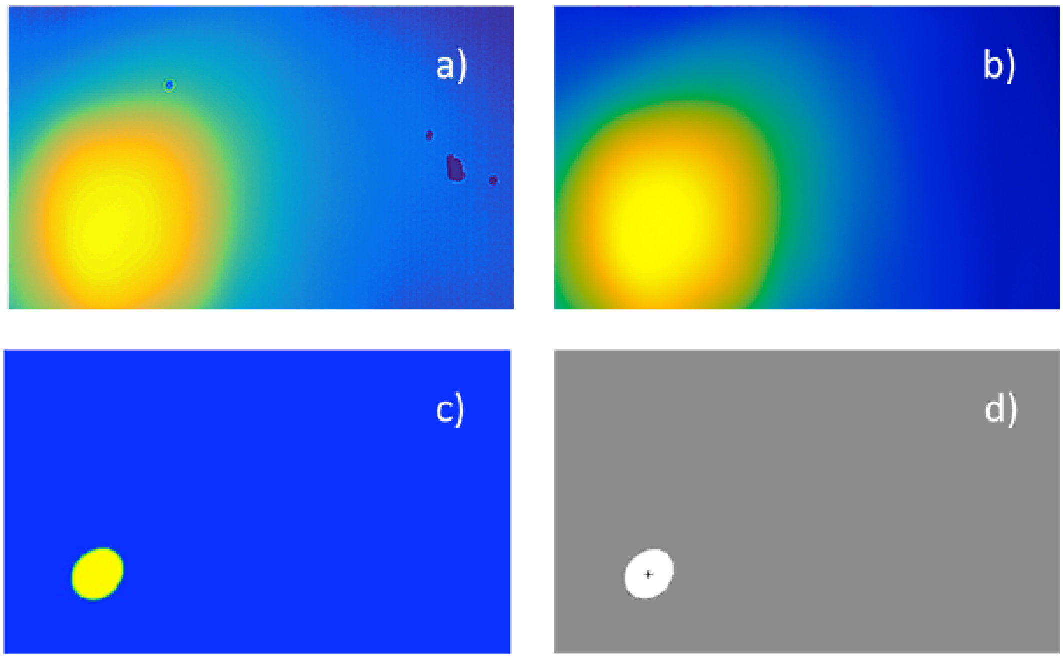


Figure 105: Image analysis: original image (a), noise reduction and surface defects elimination (b), thresholding (c) and light-spot center computation (d).

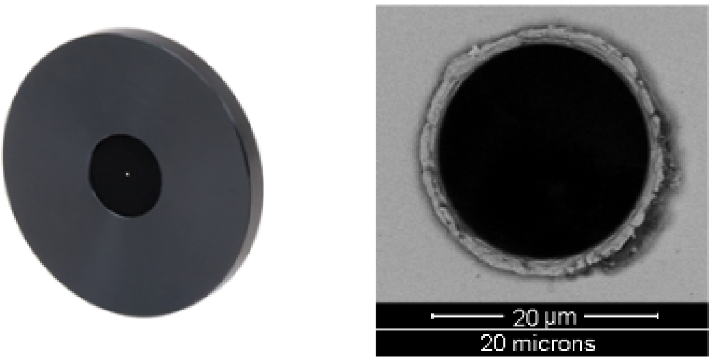


Figure 106: Pinhole aperture in mounted position and under SEM microscopy to verify circularity and diameter.

order to get loaded. In the following paragraphs, we take advantage of the intermediate RAM cache in the CPU to perform efficient image analysis.

First of all, the algorithm performs an engaging routine: the image is converted into a grayscale array; then, the brightest pixel(s) are searched in the first row; a variable stores both the value (which is needed for searching) and the index of that pixel (column); also, the sum of the pixels is stored; at the end of the row, the sum and the index of the brightest pixel are stored. The procedure is repeated for all the rows. In order to further improve the success of the engaging routine in the identification of the active pixels, the initial grayscale image is the result of a burst of N pictures averaged to obtain a single array. Averaging multiple shots allow to remove the background noise and to detect defects on the surface of the CMOS (such as dust or other debris). In Fig. 105 it is possible to notice how the algorithm removes some dust particles that were purposely added to the surface of the sensor; in addition, the image is filtered and noise is reduced. This allows to discard any disturbance due, for example, to reflections inside the casing of the sensor The engaging routine terminates with the identification of the Region of Interest (ROI), discarding the noisy pixels. By having the brightest pixel column indices and by knowing in which rows that happened, region of interest is known in linear time, without thrashing the CPU cache.

### 5.4.3 Region of Interest tracking

Once the Region of Interest has been identified in terms of pixel coordinates, we propose a strategy to perform the tracking. The light track on the CMOS can be described with the stochastic process:

$$d\mathbf{x}_c = \mathbf{f}(\mathbf{x_c})\delta t + g(x)d\mathbf{w} \tag{114}$$

where $\mathbf{x} \in \mathbb{R}^2$ represents the sunspot center coordinates and $\mathbf{w} \in \mathbb{R}^2$ is a zero-mean Gaussian process with covariance $\mathbf{\Sigma_w}$.

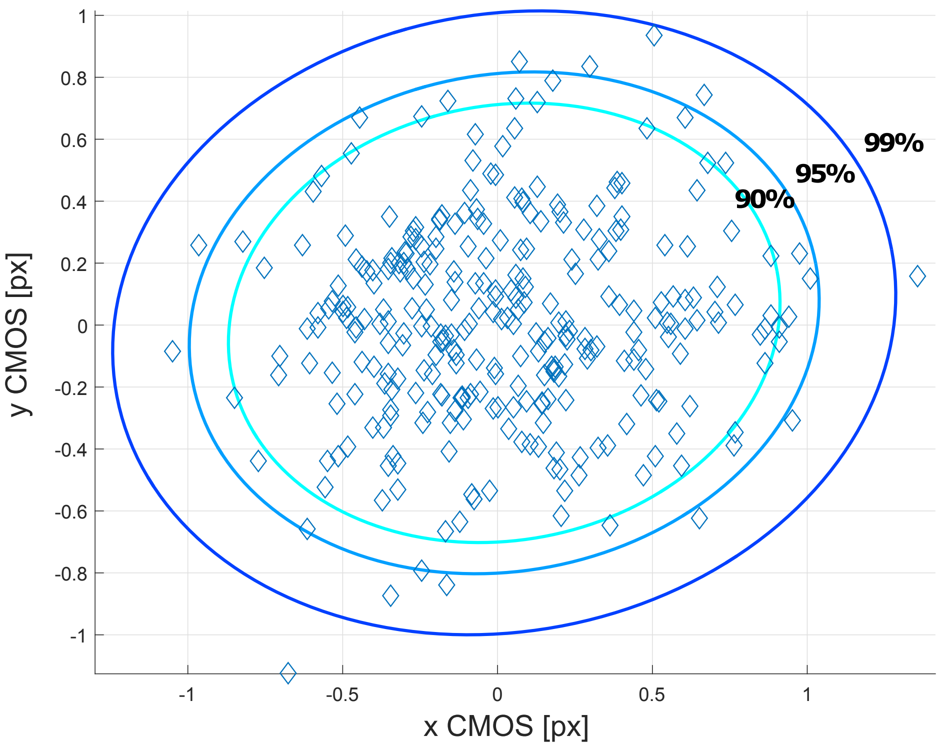


Figure 107: Calibration acquisition results, with 90%, 95% and 99% confidence ellipses.

The function $\mathbf{f}(\mathbf{x_c})$ contains the velocity of the sun projection on the CMOS and is determined by the satellite orbit, attitude and sensor installation. In our case, for Earth testing, $\mathbf{f}(\mathbf{x_c})$ is determined analytically by knowing the Ephemeris relative to the test location.

Granted that $\mathbf{f}(\mathbf{x_c})$ is a known function, it is possible to outline a tracking strategy: the image acquisition will be concentrated on a circular area whose center is determined by:

$$\begin{bmatrix} \mathrm{d}x(t) \\ \mathrm{d}y(t) \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \end{bmatrix} \mathrm{d}t + \mathbf{I}_2 \mathrm{d}\mathbf{w}(t) \tag{115}$$

Since the shape of the projection is a function of the sunspot location on the CMOS, the circular radius can be defined as a function of the azimuth and elevation. In other words, the circular area to be searched for bright pixels changes according to the location. On the other hand, since manufacturing uncertainty is present and the shape of the projection is afflicted by noise (CMOS, dust, diffraction, etc.), a proposed strategy for the search radius is based on the measured data:

$$r_i = \gamma \cdot \max(h_i, w_i) \tag{116}$$

where $h_i$ and $w_i$ are defined as the observed maximum height and width of the pixel cluster (see Fig. 108), premultiplied by a constant $\gamma$ that adds some extra outer pixels for safety. In addition, we designed a watchdog procedure that is extremely light in terms of computation burden but prevents bright pixel from being ignored by an incorrect radius-center tuple: this consists of 4 pixels, placed 90°apart from each other at $\widetilde{r} = \beta\, r$, $\beta < 1$. At the beginning of the acquisition, these 4 pixels are checked: if they are measured as *bright*, the algorithm increases the search radius $r$, Fig. 108 c); the procedure is repeated until all the watchdog pixels return a *dark* state.

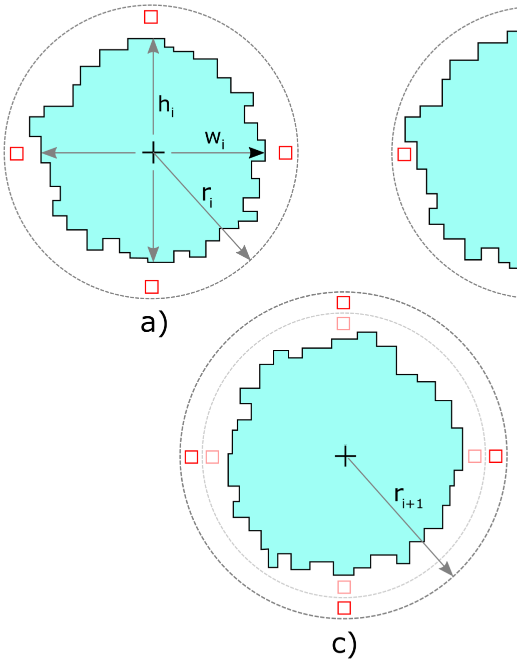


Figure 108: Sunspot adaptive identification strategy.

Further development will entail the addition of an Extend Kalman Filter or a Particle Filter to track the sunspot and integrate the measurements with the prediction model.

### 5.4.4 Led matrix calibration

One of the driving dimension for the computation of the radiation incidence angle is represented by the standoff between the CMOS surface and the pinhole, referred to as $h$ in Fig. 102. Due to tolerances in the manufacturing of the casing, it is mandatory to verify this distance. A proposed solution consists in the acquisition of the light coming from an array of high brightness LEDs: the purpose of this test is twofold. First of all, by knowing the array dimensions and the distance from the mask $k$, the dis-

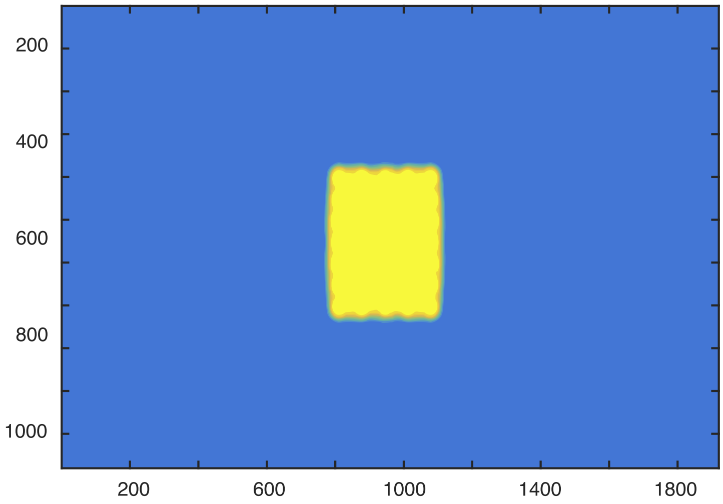


Figure 109: Calibration LED matrix, as acquired from the CMOS.

tance $h$ (or focal length) can be measured by using the perspective camera equation:

$$\begin{bmatrix} x_{px} \\ y_{px} \end{bmatrix} = -\frac{1}{k} \begin{bmatrix} h & 0 \\ 0 & h \end{bmatrix} \begin{bmatrix} x_{led} \\ y_{led} \end{bmatrix} \tag{117}$$

which maps the points in the led matrix, $\mathbf{x}_{led}$ into their projection in the CMOS, $\mathbf{x}_{px}$. In addition, the LED matrix is used for the exposure parameter calibration in the native camera firmware. An example of the acquisition during the calibration is shown in Fig. 109.

### 5.4.5 Image post-processing

After the noise background removal and the defects correction (operations performed at the beginning of each acquisition), the next part of the processing algorithm computes the center of the projected sunspot. This is done by first converting the acquisition into a gray scale image; then, by setting a custom threshold (which is a function of the camera exposure settings), the image is further converted into a binary color matrix, resulting

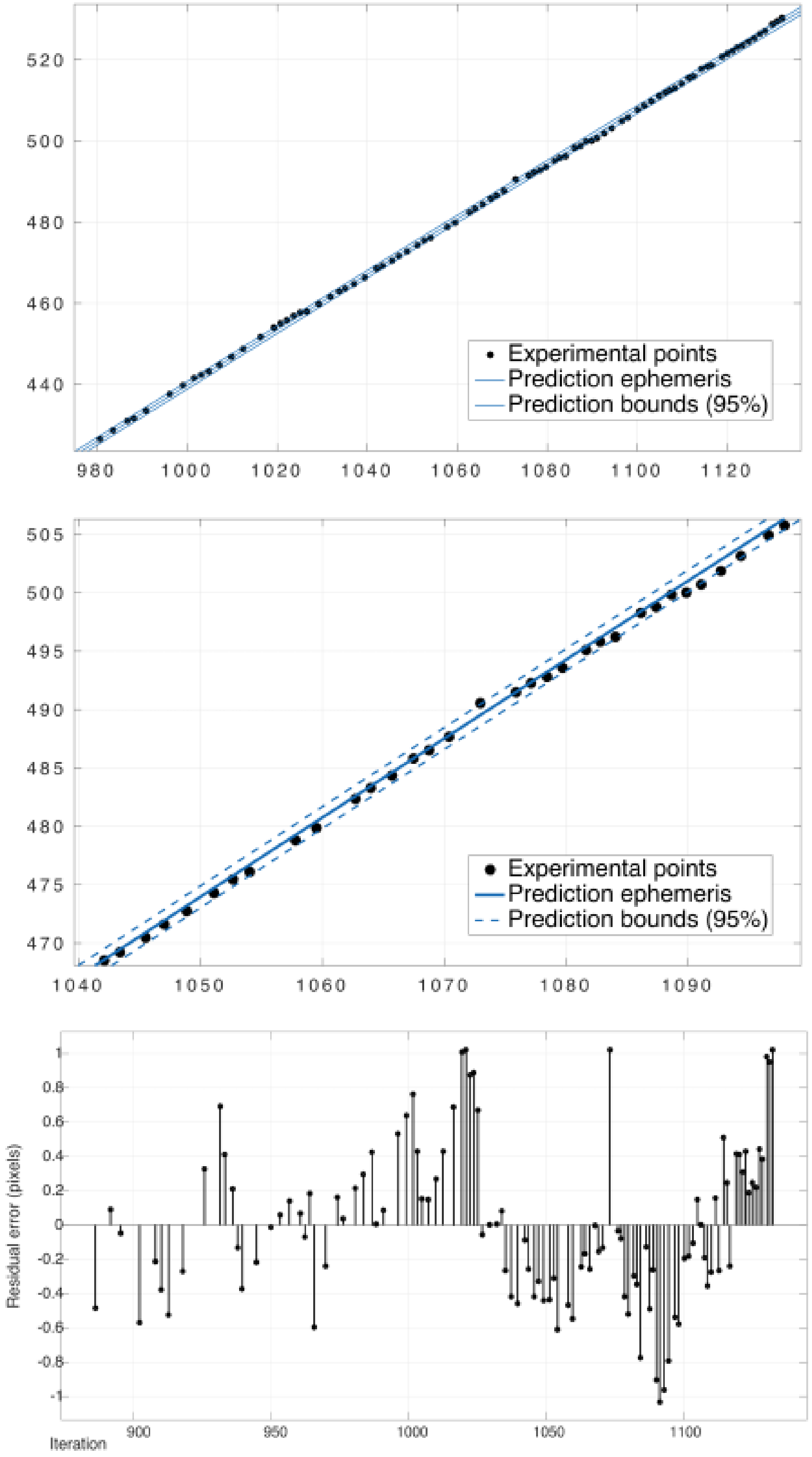


Figure 110: Measured and predicted sun trace on CMOS (a), zoomed (b). Residual errors plot and 95% bounds (c).

| LINEAR MODEL: $f(x) = p_1x^2 + p_2x + p_3$ | |
|---|---|
| Coefficients | 95% confidence bounds |
| $p_1$ = -9.216e-05 | (-1.033e-04, -6.002e-05) |
| $p_2$ = 0.8456 | (0.7905, 0.8132) |
| $p_3$ = -329.6 | (-3336.4, -310.2) |

Table 9: Linear model

in the final projected sunspot ellipse (Fig. 105); the center of this ellipse ($x_c$, $y_c$) is then computed by simply averaging the $x_i$ and $y_j$ coordinates of the ellipse's pixels:

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} \frac{\sum_{ellipse} x_i}{n_x} \\ \frac{\sum_{ellipse} y_i}{n_y} \end{bmatrix} \tag{118}$$

### 5.4.6 Accuracy

In order to estimate the accuracy, we performed several test by exposing the sensor under direct sunlight in clear sky conditions. By knowing the ephemeris of the Sun and by measuring the position and orientation of the sensor with respect to the Earth, it was possible to compare the predicted Sun trace with the measured one. We present here a sample acquisition obtained with the device. Total acquisition time is 125 minutes.

The predicted trajectory can be expressed with a polynomial model and has the following characteristics:

The experimental data, when fitted with the model, provided the following regression parameters, showing extremely good accordance with the prediction.

It can be seen that the residual orthogonal errors between the projected track and the experimental points are bounded in a ±0.7 px band at 95%

| Regression parameters | |
|---|---|
| SSE | 23.22 |
| R-square | 0.9999 |
| Adjusted R-square | 0.9998 |
| RMSE | 0.3525 |

Table 10: Regression parameters

confidence. This means that experimental accuracy obtainable with the current setup is ±0.02°, provided that the mounting process does not introduce alignment uncertainties and that, if these are present, a calibration procedure with the aid of a known light source can offset them.

## 5.5 self-power assessment

One of the design drivers in the development of the sensor is represented by the power budget, that is constrained to the low resources available on CubeSats and small satellites. First, the CMOS technology was preferred as it is demonstrated to consume less power than CCDs; a further reduction is expected with the technology evolution.

The camera selected for the first prototype has low power requirements (5V, 200 mA), and further reduction is expected using new-generation sensors; the goal is to make it possible to self-power the whole sensor with dedicated solar cells mounted on the pin-hole frame. Commercial multi-junction solar cells for space applications have reached an efficiency of more than 30%, with a net power production of about 400 $W/m^2$ (330 $W/m^2$ at the Sun sensor FOV limits) in Earth orbit; experimental cells reached up to 35% in 2014 [36] and the research is constantly increasing these values. Considering a surface of about 3x3 cm, the current technologies can furnish 0.3 W.

This result is encouraging, as the aforementioned expected trend of cells efficiency and CMOS power consumption reduction will make it possible to self-power the sensor with commercial cells in about 5-10 years.

## 5.6 robotic arm implementation

The finalized sensor has been designed to be mounted on the end effector $5^{th}$ link flange of the robotic arm. The acquired attitude information will be used to perform proximity navigation with a feedback on the attitude: the most immediate integration of the sensor in the HIL control loop will be by inserting the measurements in an Extend Kalman Filter. The complete end effector, with the addition of the sensor, can be appreciated in Fig. 111.

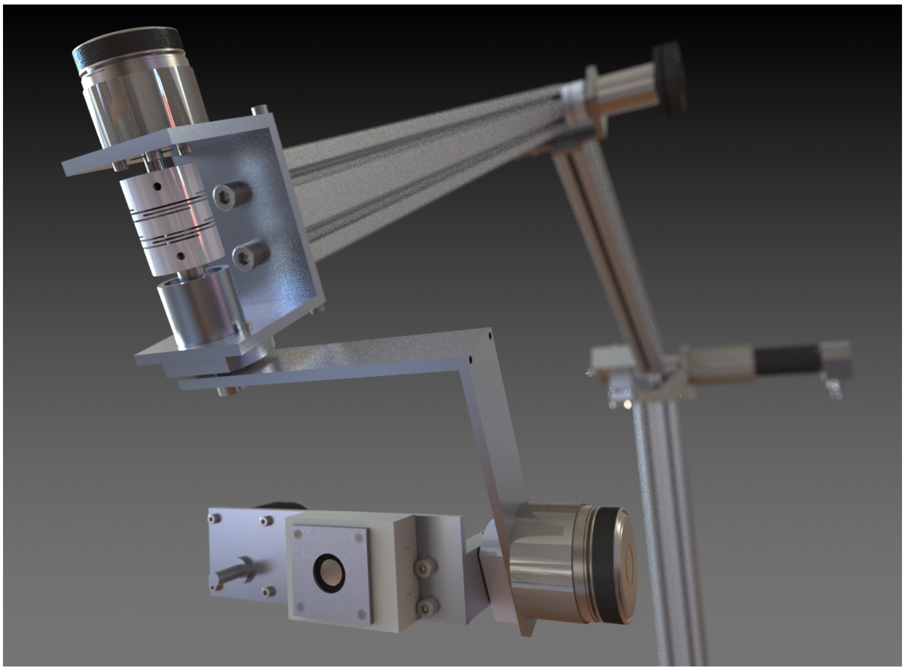


Figure 111: Render of the end effector with mounted sun sensor.

# 6 CONTROL ARCHITECTURE

## 6.1 THE CONTROL PROBLEM

The problem of controlling a robot can be formulated as that to determine the time history of the generalized forces (forces or torques) to be developed by the joint actuators, in order to guarantee the execution of the predefined tasks.

The problem of motion control of a manipulator is the topic of this chapter and is the natural extension of the previous work by this author [10]. Several techniques are available, and their main distinction is due to the way the operate: joint space or operational space. The most common techniques, due to their simplicity, belong to the first category and can be further divided according to the approach taken towards the dynamic contributions. In the following paragraph, both families of control are presented, with a particular focus on joint space procedures

## 6.2 JOINT SPACE CONTROL

In joint space techniques, the control is focused on the *q(t)* values to track the reference inputs, calculated from the desired trajectory with the aid of an inverse kinematics procedure. However, the drawback of this solution is that a joint space control does not have effects on the operational space variables, which are controlled in an open-loop fashion through the manipulator mechanical structure. It follows that any uncertainty in the structure (backlash, play, stiffness, added end effector weight) or any discrepancy between the known geometric data and the real ones causes a decrease of the accuracy on the operational space trajectory.

In a control system design process, several parameters are required in order to come up with a correct model of the system. First of all, it is mandatory to know the mechanical design of the structure.

Furthermore, the way the motion is transferred through the joints has its influence as well; if the motors, for example, are coupled with high-ratio reduction gears, it is possible to linearize the problem. This means that the analysis of the joints can take advantage of the superposition of the effects principle, and the solution is dramatically simplified. The disadvantage of this approximation is that all the nonlinear effects (such as friction, backlashes, elasticity) might affect the performances of the control.

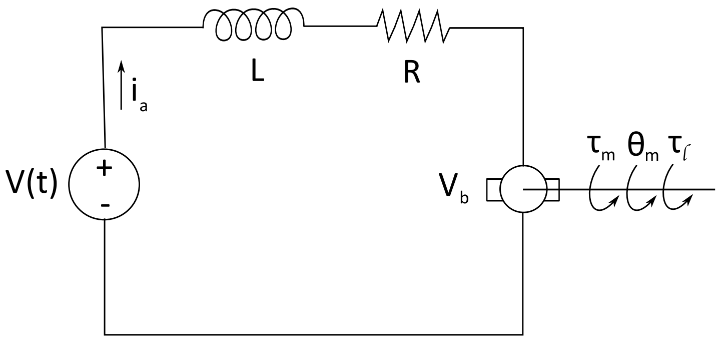
### 6.2.1 Decentralized control

In our case, all the motors come with a reduction gear, whose transmission ratio is relatively high (refer to the Mechanical Design chapter). Under this property, a linear approximation can take place: a control of this type is often referred to as *decentralized control* [2], since each linked is analyzed as a SISO independent system.

We recall, from dynamics, the differential equations describing the motion of a $n$ degrees of freedom robot [15].

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + g(\mathbf{q}) = \tau \tag{119}$$

This represents the dynamics of an multi-body system when some generalized forces $\tau$ are acting. These torques are produced by an actuator, which can be electric, hydraulic or pneumatic. An armature controlled DC motor presents the following electric diagram [2]:



Due to the presence of a movable rotor inside the stator (which creates a radial magnetic flux $\Phi$), if there is a current $i_a$ flowing, there will be a torque on the rotor:

$$\tau_m = K_1 \, \Phi \, i_a \tag{120}$$

When the rotor starts to rotate, however, an electromagnetic field arises (back **emf**), trying to oppose the current flow in the conductor. This can be expressed with:

$$V_b = K_2 \, \Phi \, \omega_m \tag{121}$$

The differential equation for the armature current is:

$$L \, \frac{di_a}{dt} + R \, i_a = V - V_b \tag{122}$$

Since the flux $\Phi$ is constant, we can rewrite $\tau_m$ as (where $K_m$ is the torque constant of the motor):

$$\tau_m = K_1 \, \Phi \, i_a = K_m \, i_a \tag{123}$$

From Eq. 121, with $K_b$ being the back **emf** constant, we have:

$$V_b = K_2 \, \Phi \, \omega_m = K_b \, \omega_m = K_b \, \frac{d\theta_m}{dt} \tag{124}$$

When the motor stalls, the corresponding torque is denoted with $\tau_0$; evaluating Eq. 122 for $V_b$=0 and $\frac{di_a}{dt}$=0:

$$V_r = R \, i_a = \frac{T \, \tau_0}{K_m} \tag{125}$$

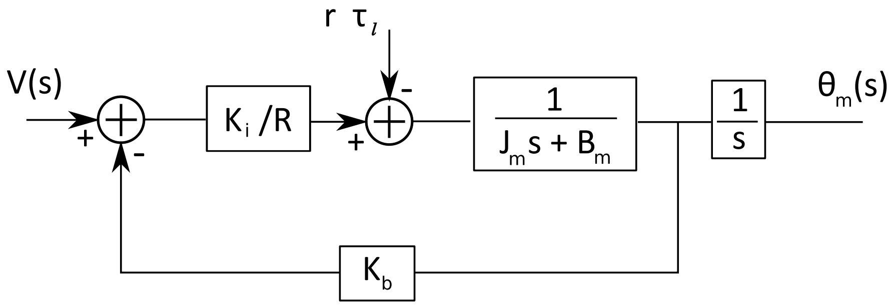


Figure 112: Block diagram of DC motor.

Which yields:

$$K_m = \frac{R\,\tau_0}{V_r} \tag{126}$$

If we couple the motor with a gear train, the differential equation describing the assemby is:

$$J_m\,\frac{d^2\theta_m}{dt^2} + B_m\,\frac{d\theta_m}{dt} = \tau_m - r\,\tau_l \tag{127}$$

Where $I_m = I_a + I_g$, that is, the sum of the actuator and the gear-train inertias. The torque at the output of the gear is $\tau_l$. The block diagram of the DC motor is pictured in Fig. 112.

At this point, we can switch from the time domain to the Laplace domain, and rewrite Eq. 124 and Eq. 127 as:

$$(Ls + R)\,I_a(s) = V(s) - K_b\,s\,\Theta_m(s) \tag{128}$$

$$(J_m\,s^2 + B_m\,s)\,\Theta_m(s) = K_i\,I_a(s) - r\,\tau_l(s) \tag{129}$$

It is straightforward to obtain the transfer function between the armature voltage $V(s)$ and the angle $\Theta_m(s)$ (imposing $\tau_l = 0$):

$$\frac{\Theta(s)}{V(s)} = \frac{K_m}{s[(L\,s + R)(J_m\,s + B_m) + K_b\,K_m]} \tag{130}$$

If $V = 0$, the transfer function between the torque $\tau_l$ and $\Theta_m(s)$ is:

$$\frac{\Theta(s)}{\tau_l(s)} = \frac{-r\,(L\,s + R)}{s[(L\,s + R)(J_m\,s + B_m) + K_b\,K_m]} \tag{131}$$

Dividing everything by $R$ and assuming that the electrical constant $L/R$ is much smaller than the mechanical constant $J_m/B_m$, we approximate the previous expressions to:

$$\frac{\Theta(s)}{V(s)} = \frac{K_m/R}{s\left(J_m\, s + B_m + \dfrac{K_b\, K_m}{R}\right)} \tag{132}$$

And:

$$\frac{\Theta(s)}{\tau_l(s)} = -\frac{r}{s\left(J_m\, s + B_m + \dfrac{K_b\, K_m}{R}\right)} \tag{133}$$

Returning for a moment to the time domain, Eq. 133 and 134 can be expressed, using the superposition of the effects, with the following differential linear equation:

$$J_m\, \ddot{\theta}_m(t) + (B_m + K_b\, K_m/R)\, \dot{\theta}_m(t) = (K_m/R)\, V(t) - r\, \tau_l(t) \tag{134}$$

At this point, we need to provide further assumptions and simplifications in order to obtain the solution. Since the output of the gear is directly connected to the link, then the generalized coordinate $q_i$ is given by (with $r_i$ being the *i-th* reduction ratio):

$$q_i = r_i\, \theta_{m_i} \tag{135}$$

It follows that the torques provided by the actuators and the load torques of the actuators share the following relationship:

$$\tau_{l_i} = \tau_i \tag{136}$$

Finally, the equations of motion of the manipulator become:

$$\sum_{j=1}^{n} d_{ji}(\mathbf{q})\ddot{q}_j + \sum_{j,k=1}^{n} c_{jki}(\mathbf{q})\dot{q}_j\dot{q}_k + g_i(\mathbf{q}) = \tau_i \tag{137}$$

$$J_m\, \ddot{\theta}_{m_i} + (B_m + K_b\, K_m/R)\, \dot{\theta}_{m_i} = (K_m/R)\, V_i - r_i\, \tau_{l_i} \tag{138}$$

If we take a closer look to the last two equations, we can note that the first one represents the nonlinear inertial, Coriolis, centripetal and gravitational coupling contributions due to the motion of the robot, whereas the second one models the actuator dynamics.

If we have to control this kind of system, a first good consideration would be to treat the nonlinear term $\tau_i$ as a disturbance entering into Eq. 138: this is extremely convenient, since Eq. 138 is linear.
After this substitution, however, the term $r_i^2 d_i i(\mathbf{q})$ appears in the coefficient $\ddot{\Theta}_{m_i}$, which hence becomes:

$$J_m + r_i^2 d_i i(\mathbf{q}) \tag{139}$$

That is, this coefficient is configuration dependent. For the purpose of the control, however, we can approximate this value with an effective value, called **effective inertia** $J_{eff}$. For the moment, we can suppose $J_{eff}$ to be the simple mean average between the value of the inertia at its minimum ($J_{min}$) and at its maximum ($J_{max}$), that is:

$$J_{eff} = \frac{J_{min} + J_{max}}{2} \tag{140}$$

We also define $B_{eff}$ as:

$$B_{eff} = B_m + K_b\, K_m / R \tag{141}$$

$$K = \frac{K_m}{R} \tag{142}$$

Thus, Eq. 138 becomes:

$$J_{eff}\, \ddot{\theta}_{m_i} + B_{eff}\, \dot{\theta}_{m_i} = K\, V_i - r_i\, d_i \tag{143}$$

In which $d_i$ is taken as a disturbance and is made up by:

$$d_i := \sum_{j \neq i}^{n} d_{ji} \ddot{q}_j + \sum_{j,k}^{n} c_{jik} \dot{q}_j \dot{q}_k + g_i \tag{144}$$

Translating this result into a block diagram, we finally obtain the scheme of Fig. 113, which is clearly an open loop system.

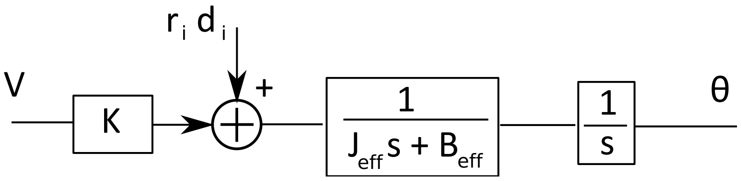


Figure 113: Open loop block diagram of manipulator link.

### 6.2.2 Design of the PD compensator

The open loop transfer function in the Laplace domain can be obtained as [37]:

$$s^2\, J_{eff}\, \Theta(s) + s\, B_{eff}\, \Theta(s) = K\, V(s) - r\, D(s) \tag{145}$$

The input $V(s)$ can be substituted by a PD control law:

$$V(s) = K_p\, [\Theta_r(s) - \Theta(s)] - s\, K_d\, [\Theta(s)] \tag{146}$$

Where $\Theta_r(s)$ is the reference command that needs to be followed by the system. Combining these two expressions, we get:

$$\Theta_m(s) = \frac{K\, K_p}{\alpha(s)}\, \Theta_r(s) - \frac{r}{\alpha(s)}\, D(s) \tag{147}$$

With $\alpha(s)$ being the characteristic equation:

$$\alpha(s) = J_{eff}\, s^2 + (B_{eff} + K\, K_d)\, s + K\, K_p \tag{148}$$

The feedback control loop can then be described by the block in Fig. 114.

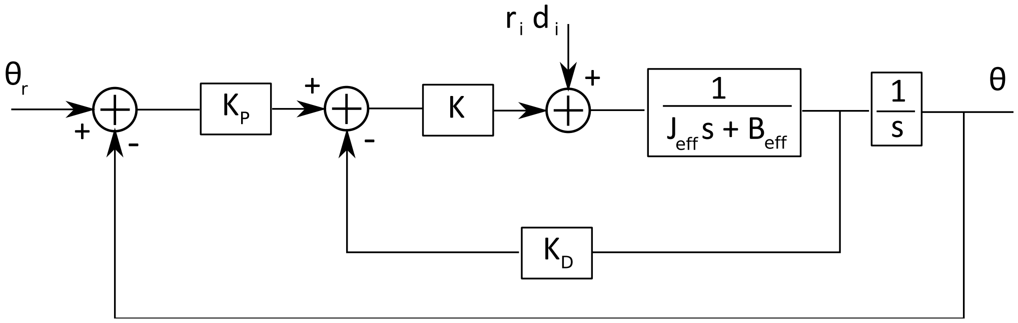


Figure 114: Block diagram of PD control system.

### 6.2.3 Design of the PID compensator

The PD controller described in the previous section is sensitive to external disturbances and high gains are typically needed in order to limit the steady state error [10, 37].

A typical upgrade to the previous system is the addition of an integral term to the PD compensator law $C(s)$:

$$C(s) = K_p + K_d\, s + \frac{K_i}{s} \tag{149}$$

As far as concerns the closed loop expression, we have:

$$\Theta_m(s) = \frac{K_d\, s^2 + K_p\, s + K_i}{\beta(s)}\, \Theta_r(s) - \frac{r\, s}{\beta(s)}\, D(s) \tag{150}$$

The characteristic equation, in this case, is the following $3^{rd}$ order polynomial:

$$\beta(s) = J_{eff}\, s^3 + (B_{eff} + K\, K_d)\, s^2 + K\, K_p\, s + K\, K_i \tag{151}$$

And the modified block diagram is pictured in Fig. 115. Note the addition of the feedforward integral part.

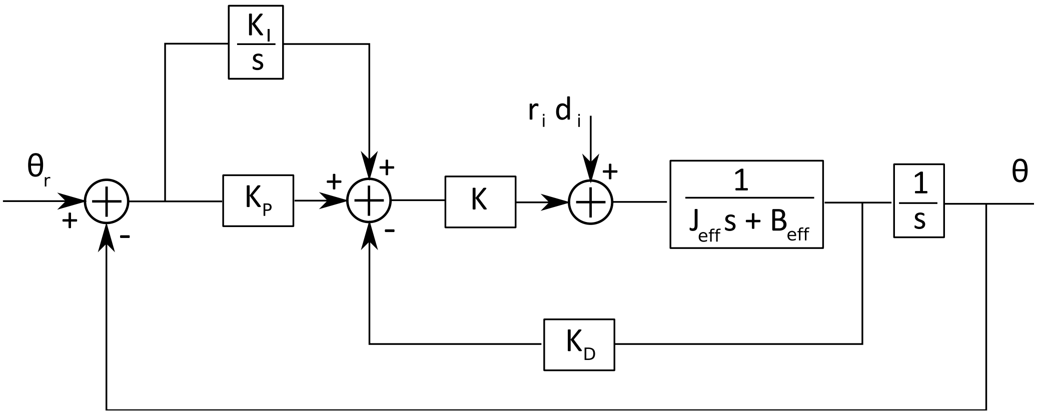


Figure 115: Block diagram of PDI control system.

*Stability analysis*

Stability can be inferred from the characteristic equation by using Routh-Hurwitz criterion. For a third order system, whose generic equation is:

$$q(s) = a_0 + a_1\,s + a_2\,s^2 + a_3\,s^3 \tag{152}$$

We have that the Routh array is:

$$\begin{array}{lll} s^3 & a_3 & a_1 \\ s^2 & a_2 & a_0 \\ s^1 & b_1 & 0 \\ s^0 & c_1 & 0 \end{array} \tag{153}$$

Where:

$$b_1 = \frac{a_2\,a_1 - a_0\,a_3}{a_2} \qquad c_1 = \frac{b_1\,a_0}{b_1} = a_0 \begin{vmatrix} a_2 & a_0 \\ a_1 & 0 \end{vmatrix} = a_0 \tag{154}$$

It follows that, since stability occurs when all the elements of Routh matrix's first column are positive, the system is stable if:

$$a_2\,a_1 > a_0\,a_3 \tag{155}$$

Hence:

$$K_i < \frac{(B_{eff} + K\,K_d)}{J_{eff}} \cdot K_p \tag{156}$$

for such a controller, it can be seen that, after the overshoot, the response tends to the imposed set-point. The error at steady state, hence, is zero.

The addition of the integral part to the control law is mandatory if we are looking for a system which is able to reject external disturbances (that are always present).

### 6.2.4 Extension to a multibody system

The previous analysis of the PD and PID control system concerned the control of a single link. In the case of a multi-body structure, as in our case, the problem can be solved by invoking the linearity of the model (if the hypotheses on the high gear ratio and the slow dynamics are fulfilled [10]).

This extension is fairly straightforward, since any dependency among the bodies has been removed. Hence, every link will be modeled by following the approach explained in the previous paragraphs, and the gains will be tuned in order to obtain the best overall performances.

With these simplifications, the only actual way to verify the performances of system would be to simulate a control and to post-analyze the results; this is due, first of all, to the fact that the inertia seen by each link has been approximated with the effective inertia $J_{eff}$, even if this parameter is clearly configuration dependent.

Moreover, the input disturbance, that should take into account all the nonlinear effects, cannot be known exactly at each step, and an educated guess on its value has to be made, introducing another relevant source of uncertainties. Note that, again, the slower the dynamics of the object, the better this simplified model will control the system adequately.

## 6.3 operational space control

The operational space control enables the manipulator to obtain a greater degree of precision in the cartesian space: that is, the end effector position is actively controlled and is not longer a byproduct of the accuracy with which the geometry of the manipulator is known.

However, this global approach requires a greater complexity; notice, in fact, that the inverse kinematics algorithm is now embedded into the feedback control loop. This slows down the algorithm and requires higher computational performances. Moreover, the afore-mentioned advantage on the end effector position presents actually an obvious limit. The measurement of the cartesian variables, in fact, is not always[1] performed directly, but via the application of the direct kinematic algorithm to the encoders' readings.

Hence, since this technique does not clear the need of a having good knowledge of the robot parameters, it doesn't makes sense to go to the trouble of implementing such a cumbersome and CPU consuming control law. The need for the extra computing power to run the model at a sufficient rate might not be worthwhile.

The most common industrial robots, for economic reasons, do not use this technique: instead, present-day manipulators are controlled with very simple control laws that generally are just error driven.

For all the above reasons, in this section we are just going to introduce the schematics of the principal control blocks without diving too deep into the details, leaving any further analysis to the appropriate references [13, 16, 38, 14].

[1] This is not valid if there exists a cartesian sensor which avoids the need for the direct kinematics transformation: for example, cameras or vision sensors.

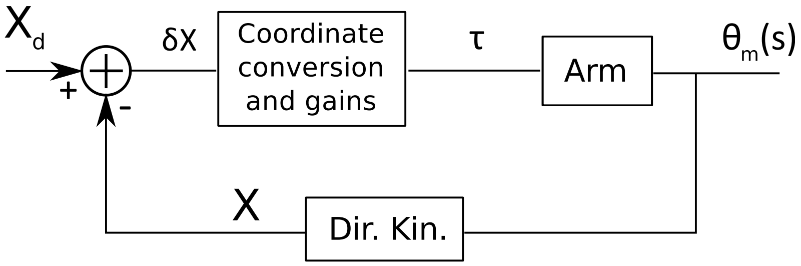


Figure 116: Block diagram of a general cartesian based control loop.

### 6.3.1 An overview

The general scheme of a control based on operational space techniques is presented in Fig. 116. The input to the close loop block is not longer the generalized coordinate $q(t)$, but it's simply the cartesian trajectory needed.

Thus, all the cartesian transformation into the joint space variables need to be performed inside the loop; this is an important drawback, that results in a lower sampling frequency if compared to joint based controls, degrading the stability and the disturbance-rejection ability of the loop [14].

Note that, even if we are talking about a cartesian control, the conversion to the joint space is necessary at some point for the calculation of the joint torques.

The procedure starts from the reading of the position sensors in order to gain information on the $\theta(t)$ values. With the aid of simple direct kinematics, the angles are converted to the actual position of the end effector. By knowing the goal coordinates $X_d$, we can obtain cartesian errors $\delta X$:

$$\delta X = X_d - X$$

From these errors, with the aid of a coordinate conversion and some appropriate gains, the torques are then computed and provided to the joints.

The most important part of the block diagram in Fig. 116 is the *Coordinate conversion and gains* block. In the literature, there are several ways of

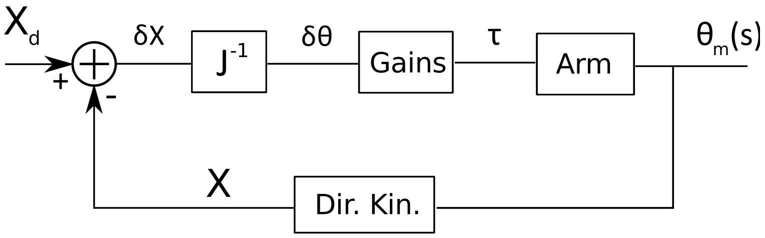


Figure 117: The *inverse-Jacobian* cartesian control block.

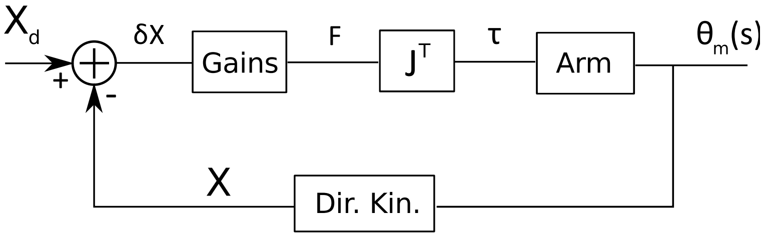


Figure 118: The *transpose-Jacobian* cartesian control block.

practically implementing this block [13, 16, 38, 14].

One the most common strategies is to use a Jacobian-type algorithm. If the time step is sufficiently small, in fact, we can map the cartesian error $\delta X$ into the corresponding displacement $\delta\theta$ in the joint space. The $\delta\theta$ errors are then multiplied by the appropriate gains to compute the torques that will presumably reduce the errors. This approach takes the name of **inverse-Jacobian controller** and is presented in Fig. 117 [14].

Another viable solution, that is strictly related to the previous one, is presented in Fig. 118. In this case, we compute the cartesian error $\delta X$ and we multiply it by a gain block to obtain a force vector $F$ in the cartesian space. We can think of this force as the vector that, if applied to the end effector, would reduce the error $\delta X$. From this vector, we can again use the differential kinematics theory to obtain the solution: $F$ gets multiplied by the transpose of Jacobian, $J^T$, and the torques are obtained. This approach

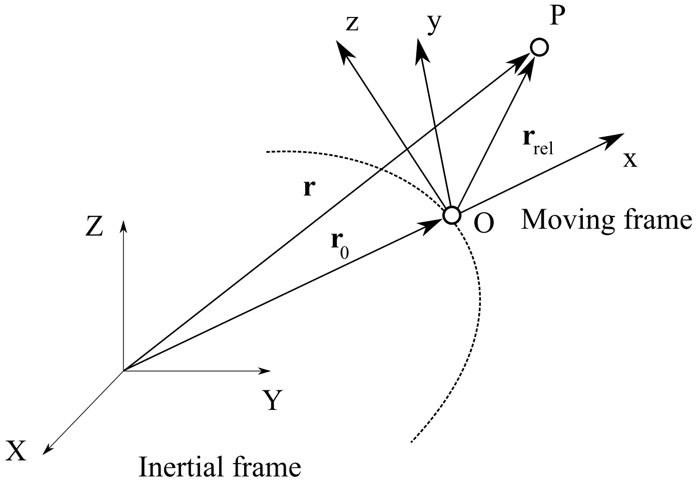


Figure 119: Absolute and relative position vectors

is referred to as **transpose-Jacobian controller**.

Although the block diagrams look neat and simple, the exact dynamics of these systems is cumbersome. It has been shown that both schemes will work, meaning that it is possible, with the appropriate gains, to make the loop stable. This partial success, however, is obscured by the need for adaptive laws: it is not possible to choose some fixed gains and have fixed closed loop poles in all the points of the workspace. The dynamics of these controllers, in fact, is influenced by the arm configuration.

## 6.4 impact modeling

### 6.4.1 Trajectory framework: relative motion in orbit

In close approach maneuvers, generally, one object (the *target*) is passive and non-maneuvering, whereas the other (the *chaser*), is active and trying to approach the target.

Referring to Fig. 119, the position of the target in the geocentric frame is given by $r_0$. The target represents also the origin of the moving frame, whose $x$ axis is along the $r_0$ direction, $y$ points in the local horizon of the target's orbit and $z$ is chosen to complete a right handed frame.

In order to analyze the motion, we recall the formulas for relative velocity and acceleration [39]:

$$\mathbf{v} = \mathbf{v}_0 + \mathbf{v}_{rel} + \boldsymbol{\Omega} \times \mathbf{r}_{rel} \tag{157}$$

$$\mathbf{a} = \mathbf{a}_0 + \mathbf{a}_{rel} + \dot{\boldsymbol{\Omega}} \times \mathbf{r}_{rel} + \boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{r}_{rel}) + 2\,\boldsymbol{\Omega} \times \mathbf{v}_{rel} \tag{158}$$

In these equations, the terms $\boldsymbol{\Omega}$ and $\dot{\boldsymbol{\Omega}}$ need to be computed. The angular moment of the orbit can be calculated as:

$$\mathbf{h} = \mathbf{r}_0 \times \mathbf{v}_0 = (\mathbf{r}_0\,\Omega)\,\hat{\mathbf{z}} = r_0^2\,\boldsymbol{\Omega} \tag{159}$$

From which, the angular velocity of the moving frame is:

$$\boldsymbol{\Omega} = \frac{\mathbf{r}_0 \times \mathbf{v}_0}{r_0^2} \tag{160}$$

As far as concerns the acceleration $\dot{\boldsymbol{\Omega}}$, we derive the previous equation:

$$\dot{\boldsymbol{\Omega}} = \frac{1}{r_0^2}(\dot{\mathbf{r}}_0 \times \mathbf{v}_0 + \mathbf{r}_0 \times \dot{\mathbf{v}}_0) - \frac{2}{r_0^3}\dot{r}_0(\mathbf{r}_0 \times \mathbf{v}_0) \tag{161}$$

Which yields (recalling that $\dot{\mathbf{r}}_0 \times \mathbf{v}_0 = 0$ and $\mathbf{r}_0 \times \dot{\mathbf{v}}_0 = 0$[2]):

$$\dot{\boldsymbol{\Omega}} = -\frac{2}{r_0}\dot{r}_0\boldsymbol{\Omega} \tag{162}$$

Finally, since $\dot{r}_0 = \mathbf{v}_0 \cdot \mathbf{r}_0 / r_0$:

$$\dot{\boldsymbol{\Omega}} = -\frac{2(\mathbf{r}_0 \times \mathbf{v}_0)}{r_0^2}\boldsymbol{\Omega} \tag{163}$$

By substituting equations Eq. 160 and 163 in Eq. 157 and 158, one can calculate the relative velocity and accelerations of an object measured along the frame centered in the target.

[2] Note that $\dot{\mathbf{r}}_0 \times \mathbf{v}_0 = \mathbf{v}_0 \times \mathbf{v}_0 = 0$. As far as concerns the second equation: $\dot{\mathbf{v}}_0 = -\frac{\mu}{r_0^3}\mathbf{r}_0$. Hence: $\mathbf{r}_0 \times \dot{\mathbf{v}}_0 = \mathbf{r}_0 \times \left(-\frac{\mu}{r_0^3}\mathbf{r}_0\right) = -\frac{\mu}{r_0^3}(\mathbf{r}_0 \times \mathbf{r}_0) = \mathbf{0}$.

*Linearization*

We start by recalling that the inertial acceleration of the chaser is given by:

$$\ddot{\mathbf{r}} = -\mu\frac{\mathbf{r}}{r^3} \tag{164}$$

From this, since $\mathbf{r} = \mathbf{r}_0 + \mathbf{r}_{rel}$, we can write that:

$$\ddot{\mathbf{r}}_{rel} = -\ddot{\mathbf{r}}_0 - \mu\frac{\mathbf{r}_0 + \mathbf{r}_{rel}}{r^3}. \tag{165}$$

Referring to Fig. 119, if $\mathbf{r}_{rel}$ is much smaller than $\mathbf{r}_0$ (which is the case of a close approach maneuver), then Eq. 165 can be linearized as follows:

$$\ddot{\mathbf{r}}_{rel} = -\frac{\mu}{r_0^3}\left[\mathbf{r}_{rel} - \frac{3}{r_0^2}(\mathbf{r}_0 \cdot \mathbf{r}_{rel})\mathbf{r}_0\right] \tag{166}$$

Since $\mathbf{R} = R\,\hat{\mathbf{k}}$ and $\mathbf{r}_{rel} = \delta x\hat{\mathbf{i}} + \delta y\hat{\mathbf{j}} + \delta z\hat{\mathbf{k}}$, we can further simplify Eq. 166:

$$\ddot{\mathbf{r}}_{rel} = -\frac{\mu}{r_0^3}(-2\delta x\hat{\mathbf{i}} + \delta y\hat{\mathbf{j}} + \delta z\hat{\mathbf{k}}) \tag{167}$$

To avoid confusion, note that this is the linearized acceleration of the chaser with respect to the geocentric frame; our goal, on the other hand, is to obtain the motion equations with reference to the target centered frame. This means plugging Eq. 167 into Eq. 158.

Omitting the tedious algebraic calculations, we can write the final expression for the relative acceleration:

$$\delta\mathbf{a}_{rel} = -\frac{\mu}{r_0^3}(-2\delta x\hat{\mathbf{i}} + \delta y\hat{\mathbf{j}} + \delta z\hat{\mathbf{k}}) - \frac{2(\mathbf{V}\cdot\mathbf{r}_0)h}{r_0^4}(\delta y\hat{\mathbf{i}} - \delta x\hat{\mathbf{j}}) + \frac{h^2}{r_0^4}(\delta x\hat{\mathbf{i}} + \delta y\hat{\mathbf{j}}) - 2\frac{h}{r_0^2}(\delta\dot{x}\hat{\mathbf{j}} - \delta\dot{y}\hat{\mathbf{i}}) \tag{168}$$

Its components are then:

$$\begin{cases} \delta\ddot{x} - \left(\dfrac{2\mu}{r_0^3} + \dfrac{h^2}{r_0^4}\right)\delta x + \dfrac{2(\mathbf{V}\cdot\mathbf{r}_0)h}{r_0^4}\delta y - 2\dfrac{h}{r_0^2}\delta\dot{y} = 0 \\ \delta\ddot{y} - \left(\dfrac{\mu}{r_0^3} - \dfrac{h^2}{r_0^4}\right)\delta y - \dfrac{2(\mathbf{V}\cdot\mathbf{r}_0)h}{r_0^4}\delta x + 2\dfrac{h}{r_0^2}\delta\dot{x} = 0 \\ \delta\ddot{z} + \dfrac{\mu}{r_0^3}\delta z = 0 \end{cases} \tag{169}$$

*Clohessy-Wiltshire equations*

Equations 169 describe the relative motion of the chaser in the target frame, which has a generic elliptical orbit around the center body. If this orbit is circular, then:

$$\mathbf{V} \cdot \mathbf{r}_0 = 0 \qquad h = \sqrt{\mu\, r_0} \tag{170}$$

And Eq. 169 become:

$$\begin{cases} \delta\ddot{x} - 3\dfrac{\mu}{r_0^3}\delta x - 2\sqrt{\dfrac{\mu}{r_0^3}}\delta\dot{y} = 0 \\ \delta\ddot{y} + 2\sqrt{\dfrac{\mu}{r_0^3}}\delta\dot{x} = 0 \\ \delta\ddot{z} + \dfrac{\mu}{r_0^3}\delta z = 0 \end{cases} \tag{171}$$

These are called Clohessy-Wiltshire (CW) equations and they are relatively simple to solve. With a simple analytical integration, we can obtain the velocity and the position equations:

$$\begin{cases} \delta\dot{x} = 3n\sin(nt)\delta x_0 + \cos(nt)\delta\dot{x}_0 + 2\sin(nt)\delta\dot{y}_0 \\ \delta\dot{y} = 6n[\cos(nt) - 1]\delta x_0 - 2\sin(nt)\delta\dot{x}_0 + [4\cos(nt) - 3]\delta\dot{y}_0 \\ \delta\dot{z} = -n\sin(nt)\delta z_0 + \cos(nt)\delta\dot{z}_0 \end{cases} \tag{172}$$

$$\begin{cases} \delta x = [4 - 3\cos(nt)]\delta x_0 + \dfrac{\sin(nt)}{n}\delta\dot{x}_0 + \dfrac{2}{n}[1 - \cos(nt)]\delta\dot{y}_0 \\ \delta y = 6[\sin(nt) - nt]\delta x_0 + \delta y_0 + \dfrac{2}{n}\delta[\cos(nt) - 1]\delta\dot{x}_0 + \dfrac{1}{n}[4\sin(nt) - 3nt]\delta\dot{y}_0 \\ \delta z = \cos(nt)\delta z_0 + \dfrac{1}{n}\sin(nt)\delta\dot{z}_0 \end{cases} \tag{173}$$

In order to improve the relative motion analysis, the handling of CW equations is made easier with the a matrix notation. First of all, we define:

$$\delta\mathbf{r}(t) = \begin{Bmatrix} \delta x(t) \\ \delta y(t) \\ \delta z(t) \end{Bmatrix} \qquad \delta\mathbf{v}(t) = \begin{Bmatrix} \delta\dot{x}(t) \\ \delta\dot{y}(t) \\ \delta\dot{z}(t) \end{Bmatrix} \tag{174}$$

Whose corresponding initial values, for $t = 0$ are:

$$\delta\mathbf{r}_0 = \begin{Bmatrix} \delta x_0 \\ \delta y_0 \\ \delta z_0 \end{Bmatrix} \qquad \delta\mathbf{v}_0 = \begin{Bmatrix} \delta\dot{x}_0 \\ \delta\dot{y}_0 \\ \delta\dot{z}_0 \end{Bmatrix} \tag{175}$$

Then, the position and velocity of the chaser at instant $t$ is given by:

$$\begin{Bmatrix} \delta\mathbf{r}(t) \\ \delta\mathbf{v}(t) \end{Bmatrix} = \begin{bmatrix} \Psi_{rr}(t) & \Psi_{rv}(t) \\ \Psi_{vr}(t) & \Psi_{vv}(t) \end{bmatrix} \cdot \begin{Bmatrix} \delta\mathbf{r}_0 \\ \delta\mathbf{v}_0 \end{Bmatrix} \tag{176}$$

Or:

$$\{\delta\mathbf{r}(t)\} = [\Psi_{rr}(t)]\{\delta\mathbf{r}_0\} + [\Psi_{rv}(t)]\{\delta\mathbf{v}_0\} \tag{177}$$

$$\{\delta\mathbf{v}(t)\} = [\Psi_{vr}(t)]\{\delta\mathbf{r}_0\} + [\Psi_{vv}(t)]\{\delta\mathbf{v}_0\} \tag{178}$$

Where:

$$\mathbf{\Psi}_{rr}(t) = \begin{bmatrix} 4 - 3cos(nt) & 0 & 0 \\ 6[\sin(nt) - 1] & 1 & 0 \\ 0 & 0 & \cos(nt) \end{bmatrix} \tag{179}$$

$$\mathbf{\Psi}_{rv}(t) = \frac{1}{n} \begin{bmatrix} \sin(nt) & 2[1 - \cos(nt)] & 0 \\ [\cos(nt) - 1] & [4sin(nt) - 3nt] & 0 \\ 0 & 0 & \sin(nt) \end{bmatrix} \tag{180}$$

$$\mathbf{\Psi}_{vr}(t) = \begin{bmatrix} 3n\sin(nt) & 0 & 0 \\ 6n[\cos(nt)-1] & 0 & 0 \\ 0 & 0 & -n\sin(nt) \end{bmatrix} \tag{181}$$

$$\mathbf{\Psi}_{vv}(t) = \begin{bmatrix} \cos(nt) & 2sin(nt) & 0 \\ -2sin(nt) & 4cos(nt)-3 & 0 \\ 0 & 0 & \cos(nt) \end{bmatrix} \tag{182}$$

When using a robotic facility for the simulation of orbital maneuvers, it is fundamental to reproduce the contact dynamics. Given that the relative motion is simulated correctly with the aid of the CW expressions, the dynamic response of the satellites is strictly dependent on the inertial properties of the bodies: the simulated system, in general, will have different inertial properties and will consequently behave with its own, characteristic dynamics; since the robotic system cannot be subjected to drastic inertial modifications, a software strategy has to be implemented.

### 6.4.2 Impact definition

First of all, it is mandatory to model the dynamics of the contact[3]. Along the years, several techniques have been proposed: one of the most used is certainly the spring-dashpot model [40], that models the contact between satellites as a parallel spring-damper system, as pictured in Fig. 120, where $m_T$ and $m_C$ are the target and chaser mass respectively. The differential equations describing the system are:

$$\begin{bmatrix} m_T & 0 \\ 0 & m_C \end{bmatrix} \cdot \begin{Bmatrix} \ddot{x_T} \\ \ddot{x_C} \end{Bmatrix} + \begin{Bmatrix} 1 \\ -1 \end{Bmatrix} \cdot f(t) = \mathbf{0} \tag{183}$$

[3] For this preliminary analysis, we will focus on a 1D model, which can then be extended to a more general 3D case.

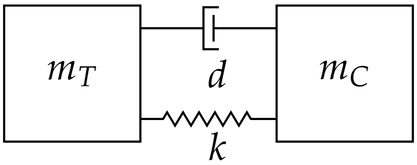


Figure 120: Spring-dashpot model.

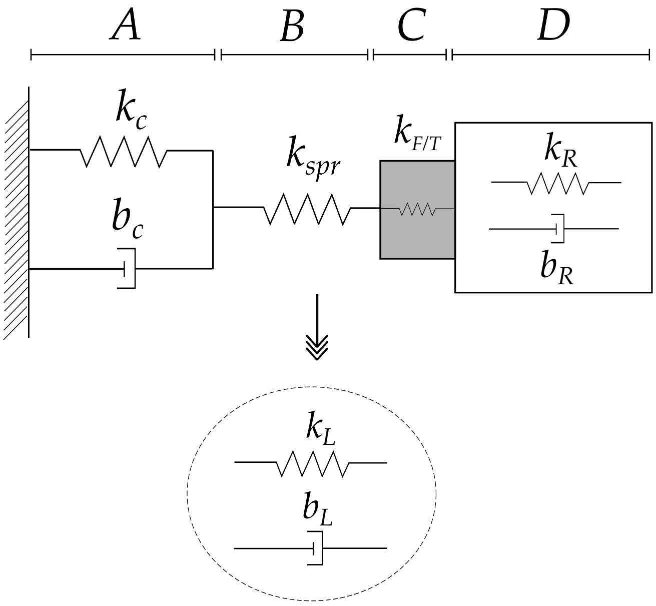


Figure 121: Lumped parameters laboratory model and its approximation.

where, if we define the relative position $x = x_C - x_T$, the force is expressed by:

$$f(t) = -kx - b\dot{x} \tag{184}$$

By using the equivalent mass $m$, the system becomes[4]:

$$m\ddot{x}(t) = kx + b\dot{x} \tag{185}$$

This fully defines the 1D approximated orbital behavior of the impact. In the laboratory case, however, the impact force will be characterized by

[4] The equivalent mass is defined as: $m = \frac{m_C m_T}{m_C + m_T}$.

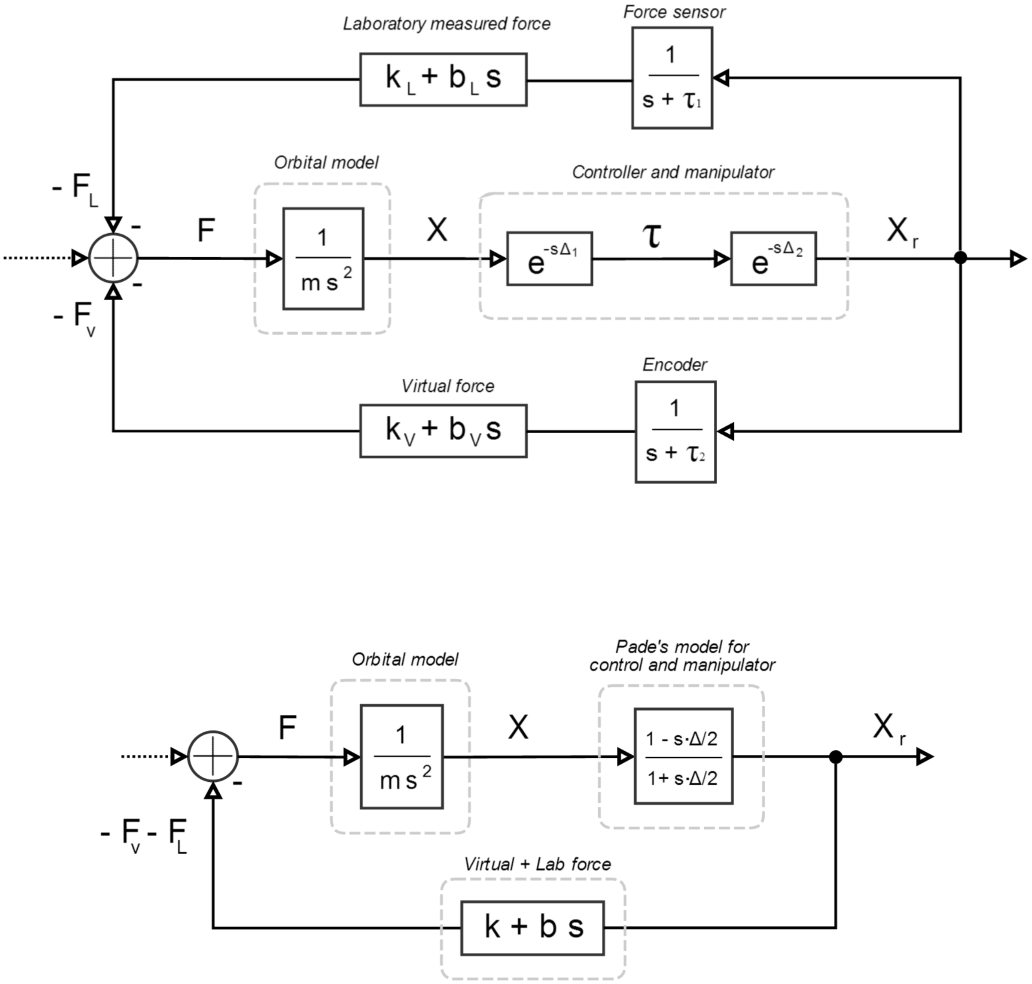


Figure 122: Virtual-force based control loop for contact dynamics simulation.

different parameters, $k_L$ and $b_L$: these parameters are hardware dependent, and represent the facility's equivalent stiffness and damping (in the hypothesis of a $2^{nd}$ order approximation of the system). Referring to Fig. 121, the laboratory installation has been modeled with 4 main blocks: $A$ represents the spring-dashpot impact model (which depends on the mockup of the docking system mounted on the end effector), $B$ is a compliance system that will be further discussed, $C$ is the stiffness of the force/torque (F/T) sensor and $D$ is the manipulator (whose $2^{nd}$ order lumped parameters are a function of the both mechanics and of the control architecture). The compliance stiffness $B$ is introduced in order to simplify the analysis: in a mechanical system, the dominant frequency of the contact is governed by the most compliant part. Thus, the insertion of a spring with a stiffness $k_{spr} << min\{k_C, k_{T/F}, k_R\}$ allows the system's overall stiffness to be approximated with that of the spring. This hypothesis is certainly true for $k_{T/F}$ (typical values for F/T sensors are on the $10^6 \div 10^7 N/m$ range); also, in this preliminary analysis, we suppose that the robot is infinitely rigid and presents no damping. As far as concerns $k_C$, its value depends on the docking interface mockup and no valid approximation can be made upon it. The simplified system is represented in the circled area of Fig. 121. The force, in this case, has the following expression:

$$f_L(t) = -k_L x - b_L \dot{x} \tag{186}$$

where:

$$k_L = k_C + k_{spr} \tag{187}$$

$$b_L = b_C \tag{188}$$

In order to simulate the actual orbital impact force (Eq. 184) in a laboratory environment (which is clearly subjected to a different dynamics, Eq. 186) it is mandatory to introduce a software artifice. With reference to [41], it is possible to implement a virtual force $f_V(t)$ in order to satisfy the following:

$$f(t) = f_L(t) + f_V(t) \tag{189}$$

Expliciting the lumped parameters, we have:

$$f(t) = -(k_L + k_V)x - (b_L + b_V)\dot{x} \tag{190}$$

Hence, the value of the virtual parameters can be computed as:

$$k_V = k - k_L \tag{191}$$

$$b_V = b - b_L + \epsilon \tag{192}$$

These parameters are computed upfront[5] and can be finely tuned in order for the laboratory dynamics to be a truthful representation of the orbital scenario.

The concept of the contact simulation technique is represented in Fig. 122 (a): note that both the inverse dynamics loop and the actuation phase are simulated as delays ($\Delta_1$ and $\Delta_2$ respectively). The total delay block, with $\Delta = \Delta_1 + \Delta_2$, can be approximated with a rational Pade's function in order to perform a frequency response analysis. In this case, we chose Pade's first order approximation [42]:

$$e^{-s\Delta} \approx \frac{1 - \frac{\Delta}{2}s}{1 + \frac{\Delta}{2}s} \tag{193}$$

Moreover, the sensors transfer functions were inserted in the diagram: for a preliminary analysis, however, since the dynamics of the system is largely within their cut-off frequency, these blocks can be ignored. Hence, the

[5] Since the damping coefficient is influenced by the delay $\Delta$ of the simulation system, the parameter $\epsilon(\Delta)$ is introduced in Eq. 192.

overall system can be simplified as the one pictured in Fig. 122 (b), where the Eq. 190 allowed for further compactness.

We then performed a frequency analysis of the system: its transfer function, for a generic input $U(t)$, is:

$$T(s) = \frac{X_r(s)}{U(s)} = \frac{1}{\frac{2+s\Delta}{2-s\Delta}ms^2 + bs + k} \tag{194}$$

The characteristic equation is:

$$\Delta \cdot ms^3 + (2m - b \cdot \Delta)s^2 + (2b - k \cdot \Delta)s + 2k = 0 \tag{195}$$

According to Routh-Hurwitz's criterion [37], the system is stable if:

$$\left\{ m - b\frac{\Delta}{2}, b - k\frac{\Delta}{2}, \left(m - b\frac{\Delta}{2}\right) \cdot \left(b - k\frac{\Delta}{2}\right) - km\Delta \right\} > 0 \tag{196}$$

Which yields the following two conditions:

$$\Delta < min\left\{\frac{2m}{b}, \frac{2b}{k}\right\} \tag{197}$$

$$4mb - 4mk\Delta - 2b^2\Delta + kb\Delta^2 > 0 \tag{198}$$

From these equations, by fixing one of the 4 parameters ($m$, $b$, $k$, $\Delta$), 3D plots can be extracted for the system design; for example, by fixing $\Delta$ (which is known once the robot control architecture has been tested), we can get the minimum mass $m$ required for simulation stability (Fig. 123), which is analytically defined as:

$$m > max\left\{\frac{b\Delta}{2}, \frac{2b^2\Delta - kb\Delta^2}{4b - 4k\Delta}\right\} \tag{199}$$

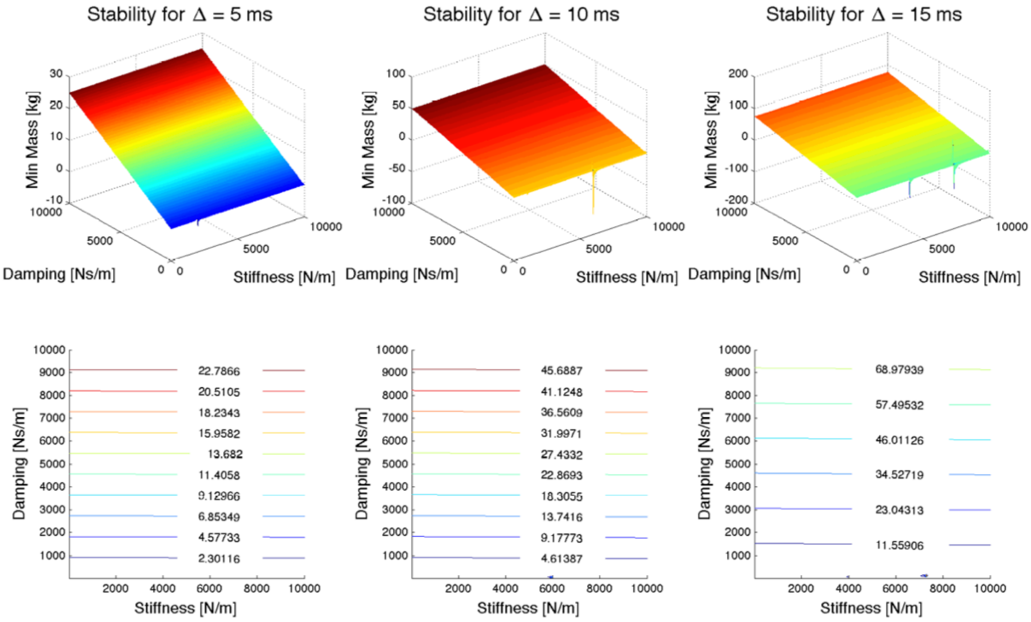


Figure 123: Minimum mass required for simulation stability, for $\Delta = \{5, 10, 15\}$ *ms*

## 6.5 COMPLETE TRAJECTORY ANALYSIS

In order to simulate the complete trajectory, three different phases must be simulated: the initial trajectory, the impact and the consequent trajectory (in the hypothesis of a non-zero coefficient of restitution[6], $\gamma \neq 0$). The first part has been discussed in the trajectory section, and allows for the simulation up to the impact point. Then, with the technique presented in section regarding the impact, if the stability conditions (Eq. 196) are met, the impact can then be simulated; this translates into the knowledge of

[6] The coefficient of restitution $\gamma$ of two colliding objects is a positive real number between 0 and 1 representing the ratio of speeds after and before an impact, taken along the line of the impact. Pairs of objects with $\gamma = 1$ collide elastically, while objects with $\gamma < 1$ collide inelastically. For $\gamma = 0$, the objects effectively "stop" at the collision, not bouncing at all.

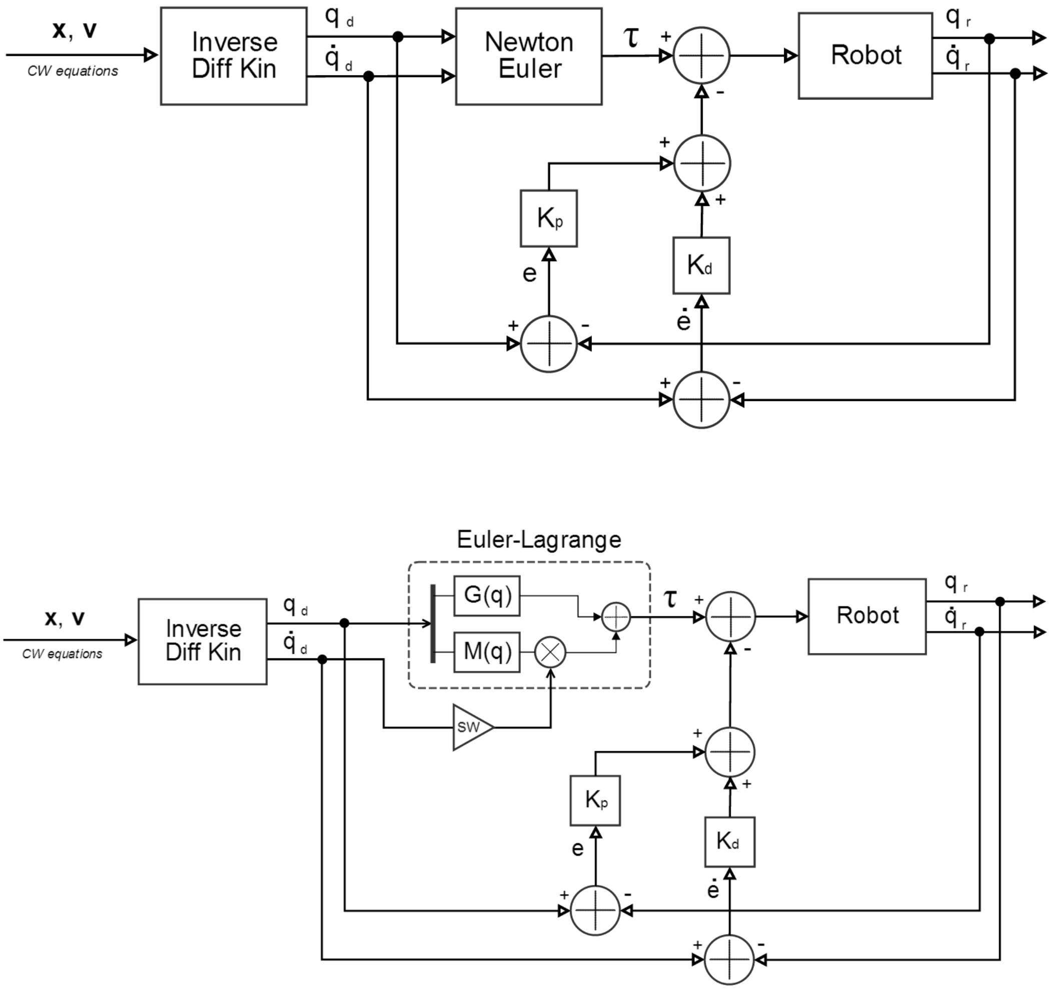


Figure 124: Control systems based on Euler-Newton and Lagrange-Euler dynamic models respectively. Note that it is possible to switch from the two Lagrange-Euler based systems by setting the switch block *SW* of Fig. 122 to 0 for gravity only, and to 1 to gravity and inertia.

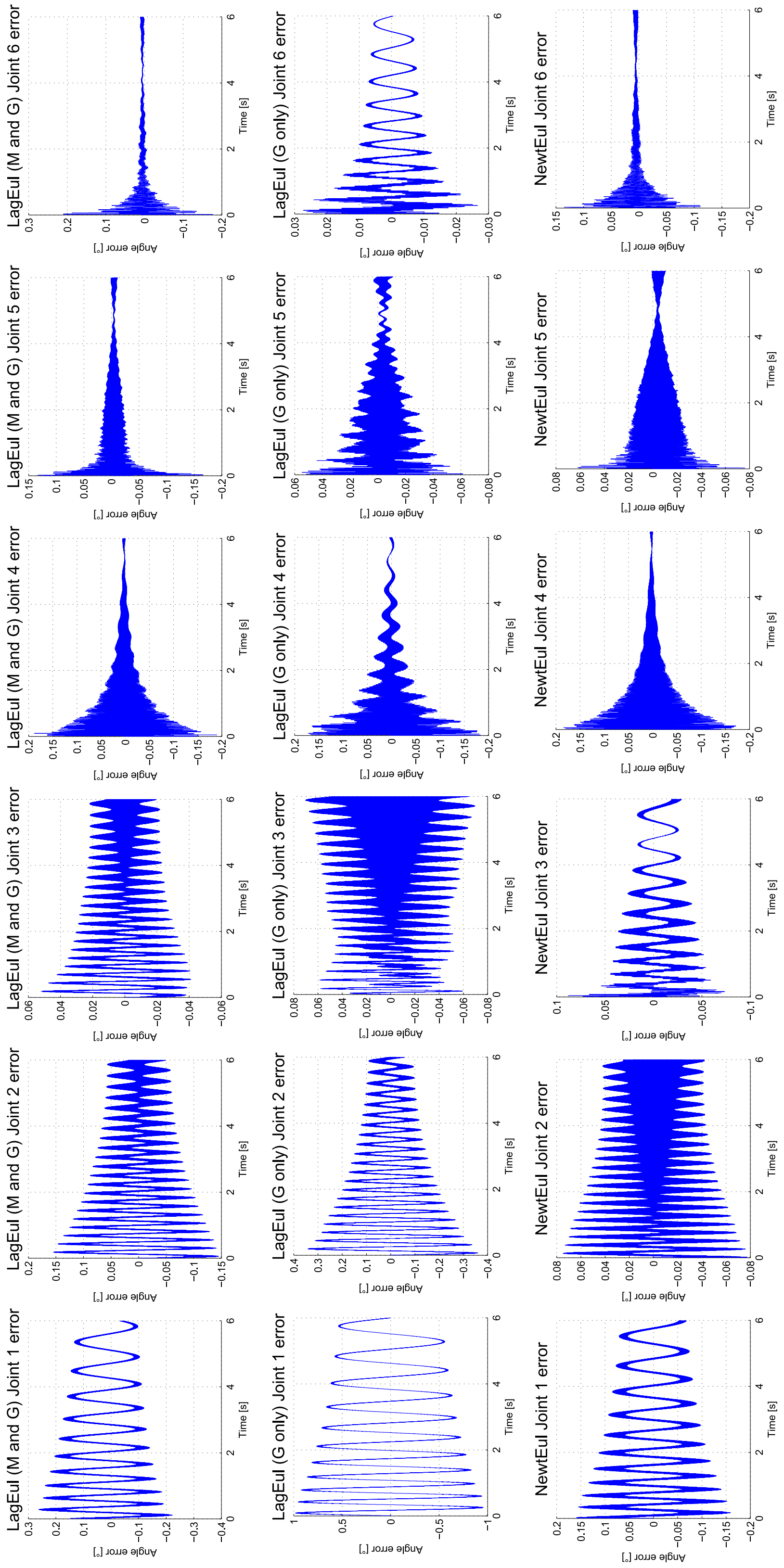


Figure 125: Joint errors for the simulated trajectory presented in the previous section. The control loops are: Lagrange-Euler with M and G for the $1^{st}$ row, Lagrange-Euler with G only for the $2^{nd}$ row, Newton-Euler for the $3^{rd}$ row.

the impact force, or, in other words, of the coefficient of restitution $\gamma$ of the contact. Hence, the computation of the $\Delta v$ caused by the impact is straightforward.

$$\Delta v = \int_{t_{imp}} \frac{|F|}{m} \cdot dt \tag{200}$$

By plugging this into Eq. 196 and supposing that within the duration of the impact the relative position of the spacecrafts has not changed, it is possible to compute the third part of the trajectory, which is trivial. The procedure, in fact, is the same of the one used for the computation of the initial trajectory. In the future, we are planning to implement an autonomous control system that computes, if $\gamma \neq 0$, the new approaching trajectory after the impact, optimizing time schedule and fuel consumption.

## 6.6 CONTROL TECHNIQUES FOR IMPACT ANALYSIS

In this section several control techniques tailored to impact simulations are analyzed: the first technique, based on Newton-Euler dynamics, is the most correct in the sense that no approximations are taking place[7]. However, this approach requires a relatively long computation time, leading to delays that could eventually give rise to instability to the discrete digital loop. In order to find a trade-off between the correctness of the model and the computation time, we analyzed approximate controls based on Lagrange's equation. While Newton-Euler's formulation, due to its iterative, non intuitive form, is not easy to manipulate, Lagrange's expression, on the other hand, thanks to an immediate physical meaning of its components, is definitely more prone to tailoring and approximations. A SimMechanics® model was used to simulate the trajectory and to compute the joint errors, which are displayed in Fig. 125. The system block diagrams are presented in Fig. 124. The presence of computation delays is taken into account by

[7] In the hypothesis of a perfect knowledge of the geometrical and inertial parameters.

feeding the simulated system with a quantized torque, whose step-size is equal to the self iteration time. The gains used for these simulation are constant and equal to $K_p = 50$ and $K_d = 0.01$.

### 6.6.1 Newton-Euler feedforward control

This control technique (Fig. 124 (a)), provides the actuators with the exact torque vector $\tau$ computed with Newton-Euler's approach. Though this system could theoretically work in an open chain fashion, there is, nonetheless, a feedback compensator that rejects external disturbances.

The reference trajectory to be followed is calculated from CW equations and is instantaneously expressed as a Cartesian vector of position and velocity that is further converted into general coordinates $\mathbf{q}$ and $\dot{\mathbf{q}}$ by using the differential kinematics technique. The average iteration time[8] is ∼2.3 ms. Although this technique is certainly not the fastest (it is almost two times longer than approach B), the controller allows for an optimal rejection of noise and system uncertainties, with a maximum error of $0.2°$ at *Joint* 4 (Fig. 125).

### 6.6.2 Lagrange-Euler (gravity compensation) feedforward control

This approach consists in the calculation of Lagrange equation's gravity term only, $\mathbf{G}(\mathbf{q})$. This means ignoring the effect of the inertia and of the centrifugal and Coriolis acceleration. For relatively slow dynamics (like a docking approach maneuver), the gravity term accounts for most of the torque that needs to be produced by the motors for a correct trajectory

[8] This has been calculated using an Intel® Core i7-3770, CPU @ 3,40 GHz, 8 GB RAM computer.

tracking. Since the gravitational term is the most straightforward and less time-consuming part to be calculated in Lagrange's equation, a feedforward control that uses only $\mathbf{G}(\mathbf{q})$ has been designed, and can be seen in Fig. 124 (a) (with the switch block value SW=0). The average iteration time is ∼1.1 ms. Though this is the fastest technique, the neglectance of the inertia and centrifugal terms gives rise to important errors, especially at *Joint* 1, where a drift of 1° was observed (Fig. 125).

### 6.6.3 Lagrange-Euler (inertia and gravity compensation) feedforward control

This technique can be seen as the natural evolution of the gravity compensation feedforward control. In addition, in fact, the inertia contribution $\mathbf{M}(\mathbf{q})$ is considered. At a price of an additional term to be computed, this system is able to track more faithfully trajectories in which the kinematics gives rise to consistent inertial forces. The system can be seen in Fig. 124 (b) (with the switch block value SW=1). The average iteration time is ∼2.1 ms. With respect to the previous system, errors are clearly mitigated, but the performances are still worse than Newton-Euler's approach (Fig. 125).

## 6.7 CONTACT ANALYSIS SUMMARY

In the previous sections, we discussed a step-to-step approach to the simulation of rendezvous and docking maneuvers, from the kinematics analysis to the modeling of the contact.

By adopting a spring-dashpot model, it was possible to analyze the dynamics of the impact and to implement, using the virtual force approach [41], a control loop for the simulation of the orbital scenario. By inspecting the system stability with Routh-Hurwitz's criterion, analytical conditions relating the facility performances and the scenarios that can be possibly simulated have been extracted. These expressions take into account the facility delay $\Delta$ and the satellites parameters $m_T, m_C, k, b$.

As far as the control system is concerned, results showed that the most efficient technique is based on Newton-Euler feedforward control, allowing to obtain an angular joint error $< 0.1^\circ$ with a computation time of ~2.3 ms. By implementing the code in a $C^{++}$ environment, we expect to drastically improve this MATLAB® based result.

For the simulation of impacts, the parameters of Fig. 121 will be fully characterized and the approximations will be verified once the manipulator has been built and the control software implemented.

## 6.8 HARDWARE IMPLEMENTATION

Hardware implementation will be described in this section using the selected parts presented in the Mechanical Design chapter. The choice of the control architecture is strictly dependent from the hardware and communication standards available. In our case, we selected EPOS drivers, which have the capability of being controlled in a wide variety of operating modes: this permits flexible configuration of the drive systems by using positioning, speed or current regulation. The communication standards used in automation for the control of multiple actuators can be divided in two main categories: ETHERCAT® and CANOpen. For this application, the CANOPEN standard will be used.

CANOPEN, for the selected MAXON® drivers, allows PID position control but also feedforward compensation. The feedforward compensation provides faster setpoint following in applications with higher load inertia and accelerations and in applications with considerable speed-dependent load (as with friction-afflicted drives). In this facility, high load inertia and friction affected drives will be present, hence the usage of the CANOPEN

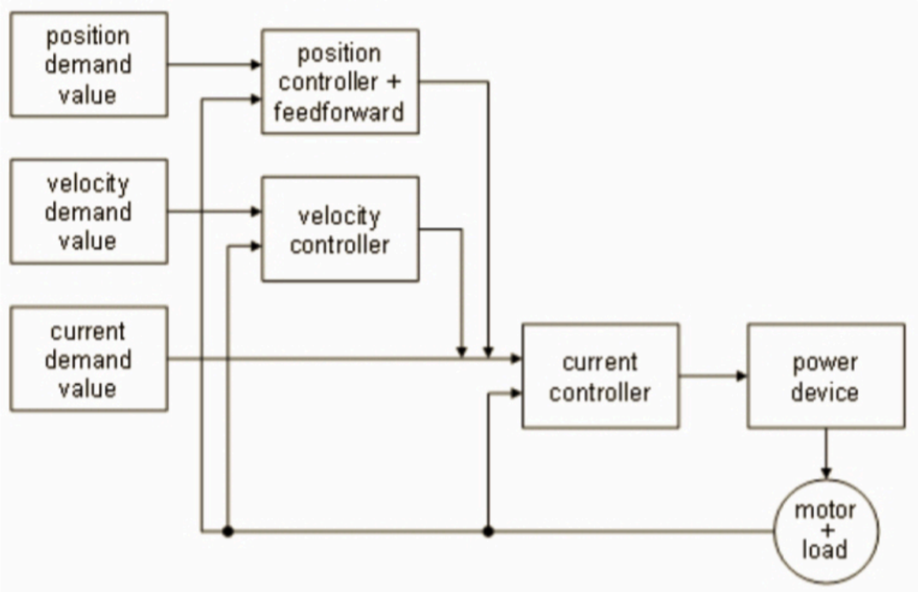


Figure 126: Controller architecture for Maxon® EPOS drivers [Maxon®, 2016]

standard seems to be appropriate.

The CANOpen controller architecture contains three builtin loops, represented in Fig. 126: 1) current regulation is used in all modes; 2) position and velocity controllers are only used in position-based, respectively velocity-based modes; 3) current control loop receives as input the position and the output of the velocity controller. The different regulation methods are depicted in Fig. 127.

These schemes can be particularized in the case of a motor+gear coupling, in which the associated elasticity and backlash of the gear create an effect of compliance as well as a delay in the drive chain. This delay influences the stability and may have a dramatic impact on the performances (dynamic behavior and precision of the system). To overcome these limitations and to combine a motor/gear system with a precise and high dynamic regulation, it will be necessary to control the motor movement as well as the load movement. This results in a new control structure called

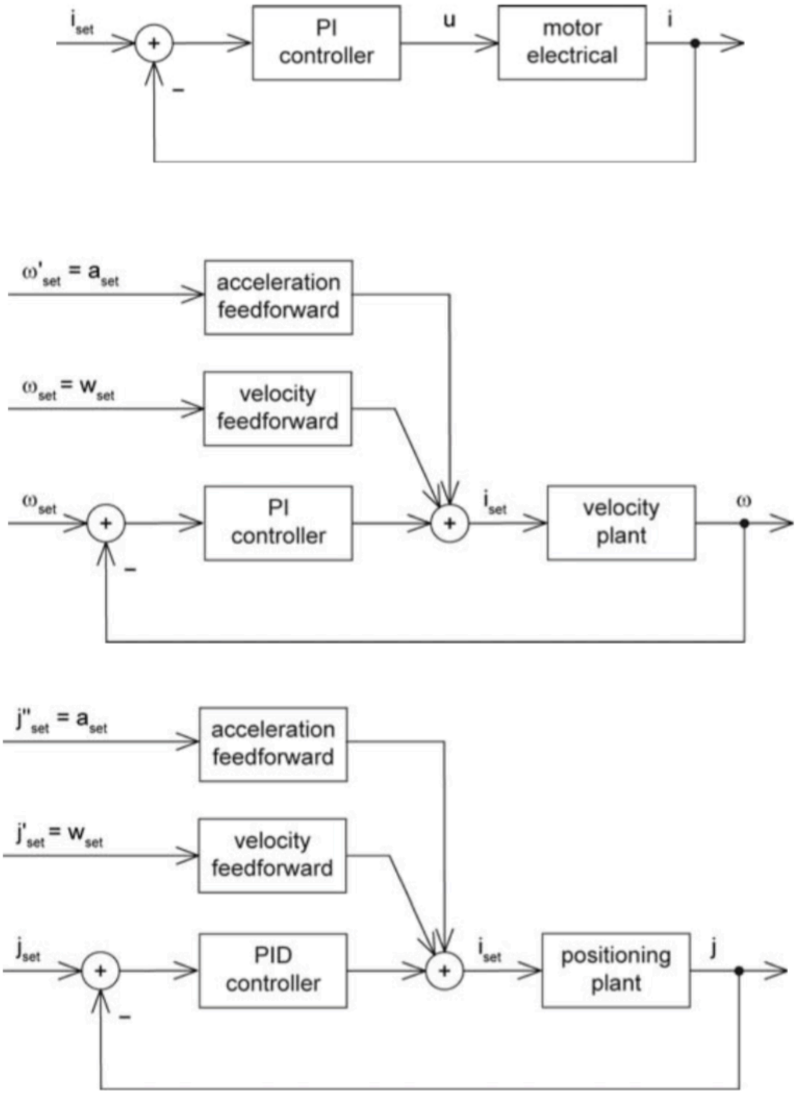


Figure 127: Different regulation methods for the selected Maxon® hardware [Maxon®, 2016].

"dual loop", featuring two individual encoders, one directly mounted to the motor, the another mounted at the gear or linear slide or directly on/near to the load [Maxon®, 2016]. An update of the previous schematics with the addition of the extra loop is presented in Fig. 128

In this application, the selected strategy is position control with feedfor-

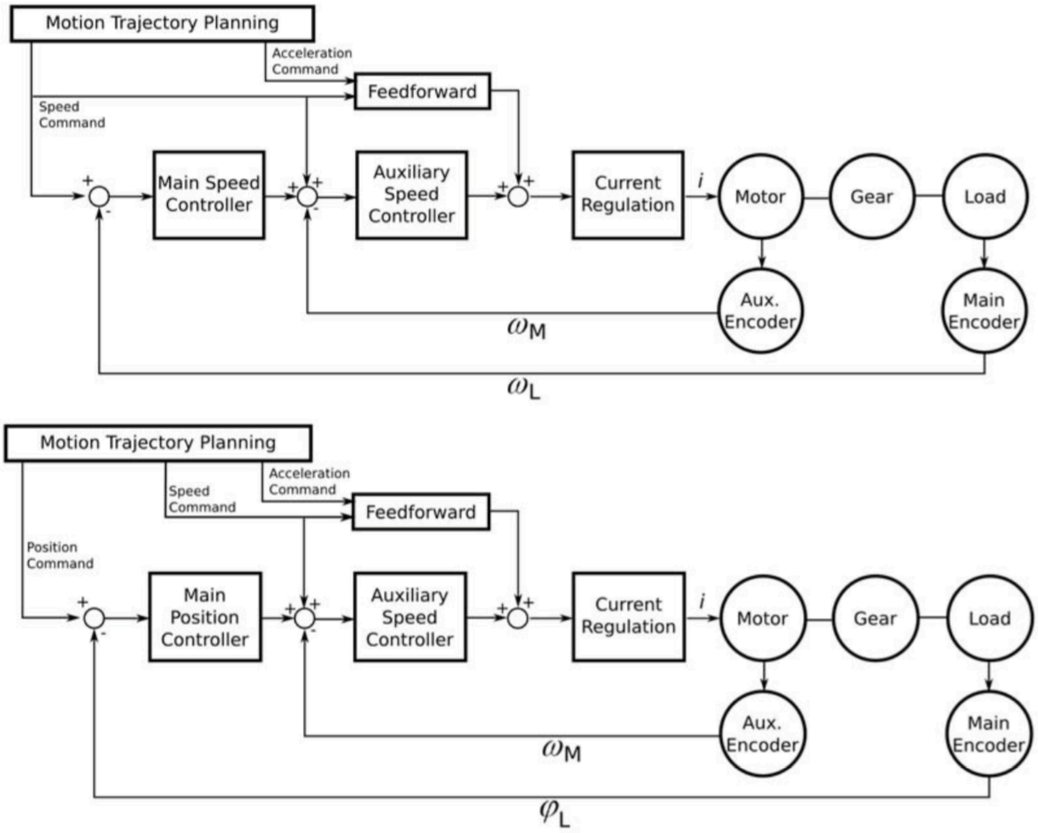


Figure 128: Dual loop control architectures: velocity and position regulations with feedforward [Maxon®, 2016]

ward. The choice of using a feedforward compensator arises for two reasons: 1) velocity feedforward provides additional current in cases where the load increases with speed, such as speed-dependent friction and 2) acceleration feedforward provides additional current in cases of high acceleration and/or high load inertias.

In terms of hardware configuration and connections, each motor is connected to a controller; the controllers are wired as the nodes of the CANBus and are connected to the CANOpen master (Fig. 129). The single controller-motor block is pictured in Fig. 130 in which all the harness is shown.

A Simulink® schematic of the control system using position control is presented in Fig. 131. The performances of the controller and the PID

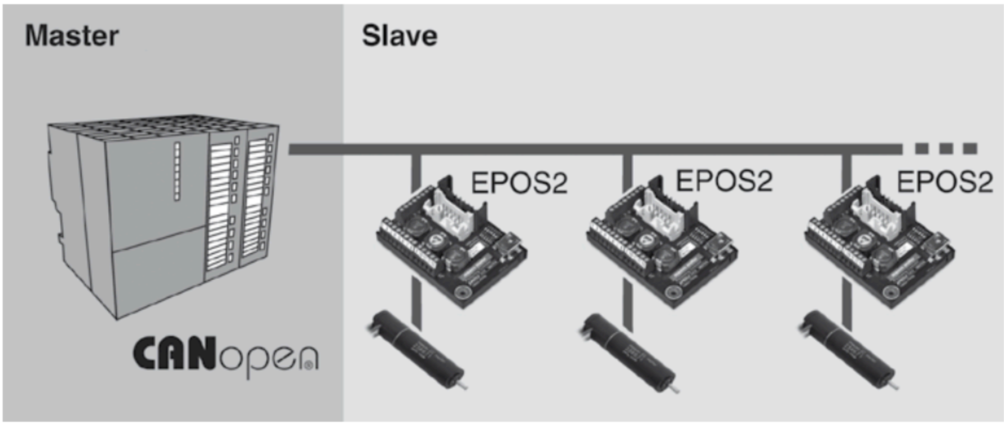


Figure 129: CANOpen multi-motor configuration [Maxon®, 2016]

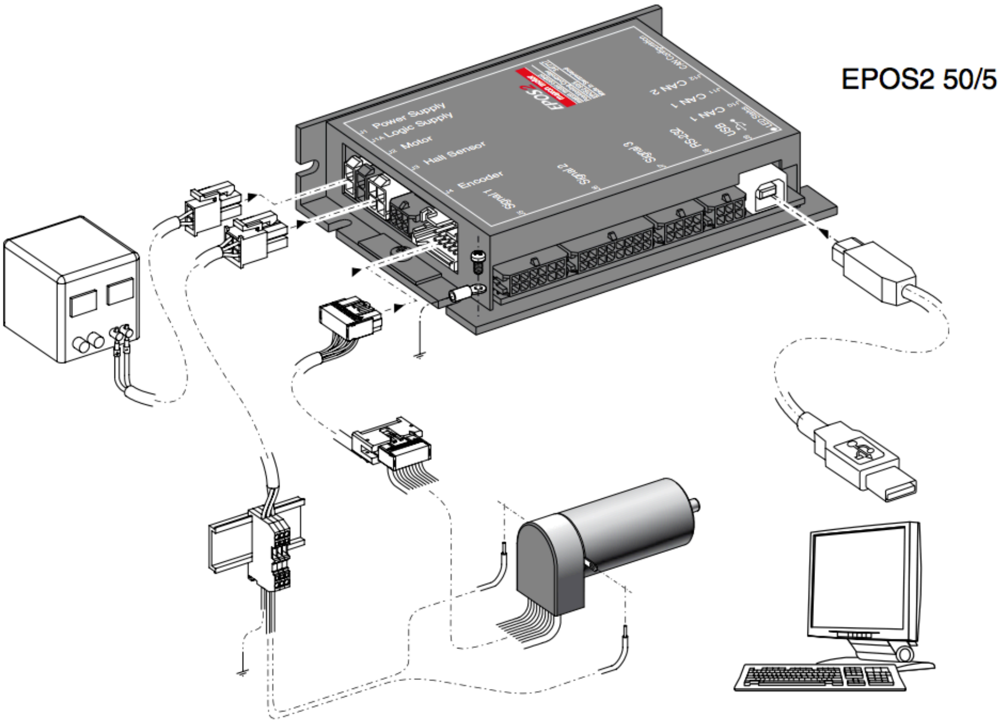


Figure 130: Single controller-motor connections

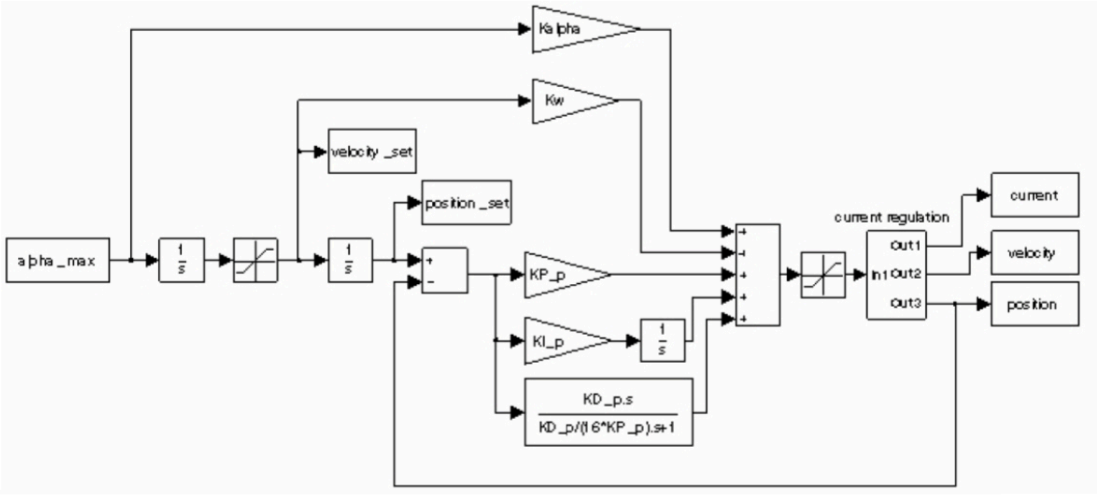


Figure 131: SIMULINK® model of the positioning controller [MAXON®, 2016]

tuning procedure for the single motors is detailed in the hardware chapter.

## 6.9 ARCHITECTURE SOLUTIONS

If the CANOPEN communication standard is chosen, it is necessary to have a master to control the slaves, in this case constituted by the MAXON® native controllers. The CANOPEN master can be either implemented on a computer running a real-time operating system in order to assure real-time performances or can be constituted by a physical unit (a PLC) which is wired to a computer through an Ethernet connection. For this facility, the master unity chosen is a PLC by B&R Automation®. The controllers are powered by individual switching power supply units; in order to monitor the instantaneous power consumption of the system, six dedicated ammeters provided the current absorbed by the motors.

The final electric schematic of the facility is pictured in Fig. 132.

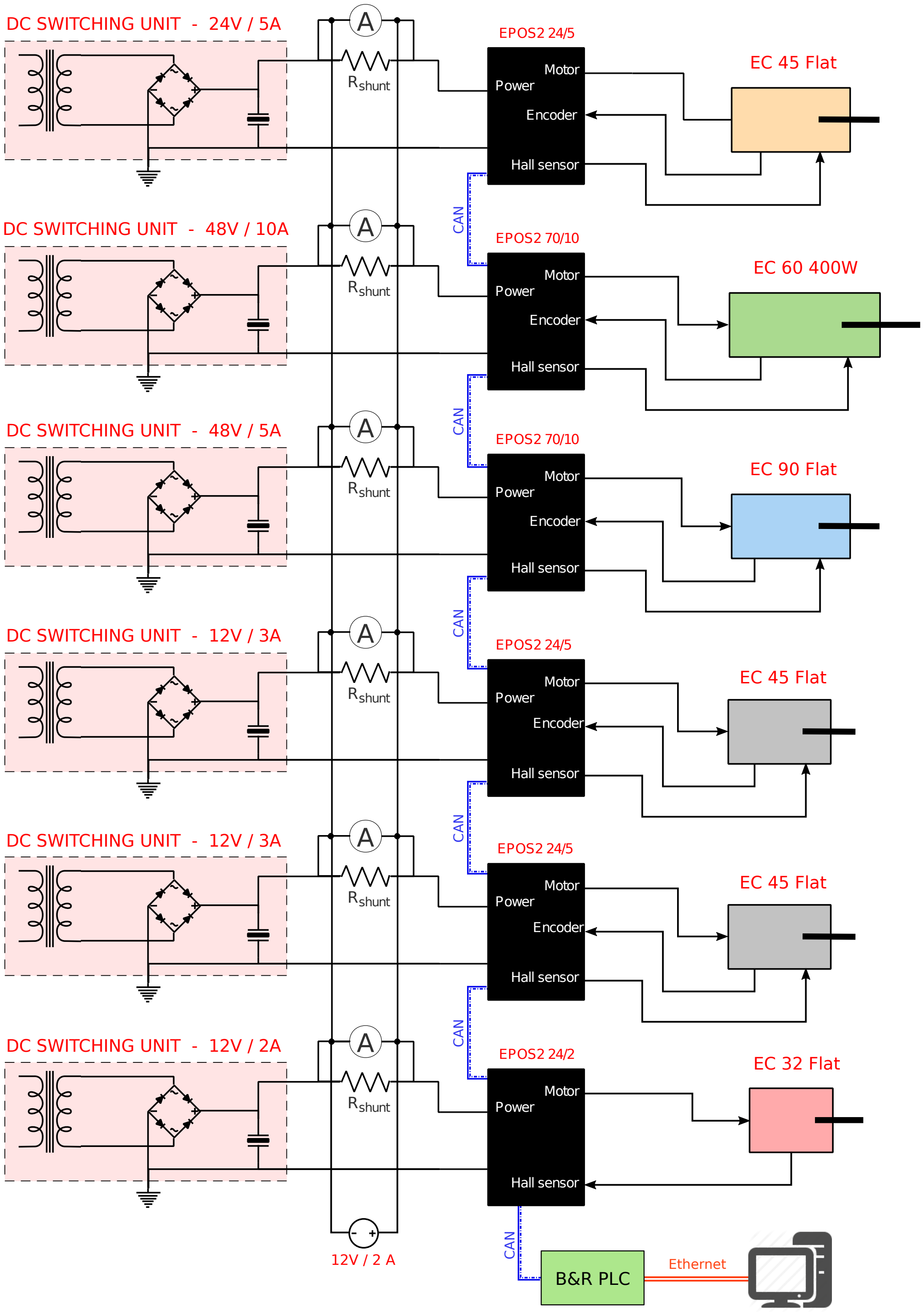


Figure 132: Electric and wiring schematic of the robotic facility.

## 6.10 MOTOR TUNING

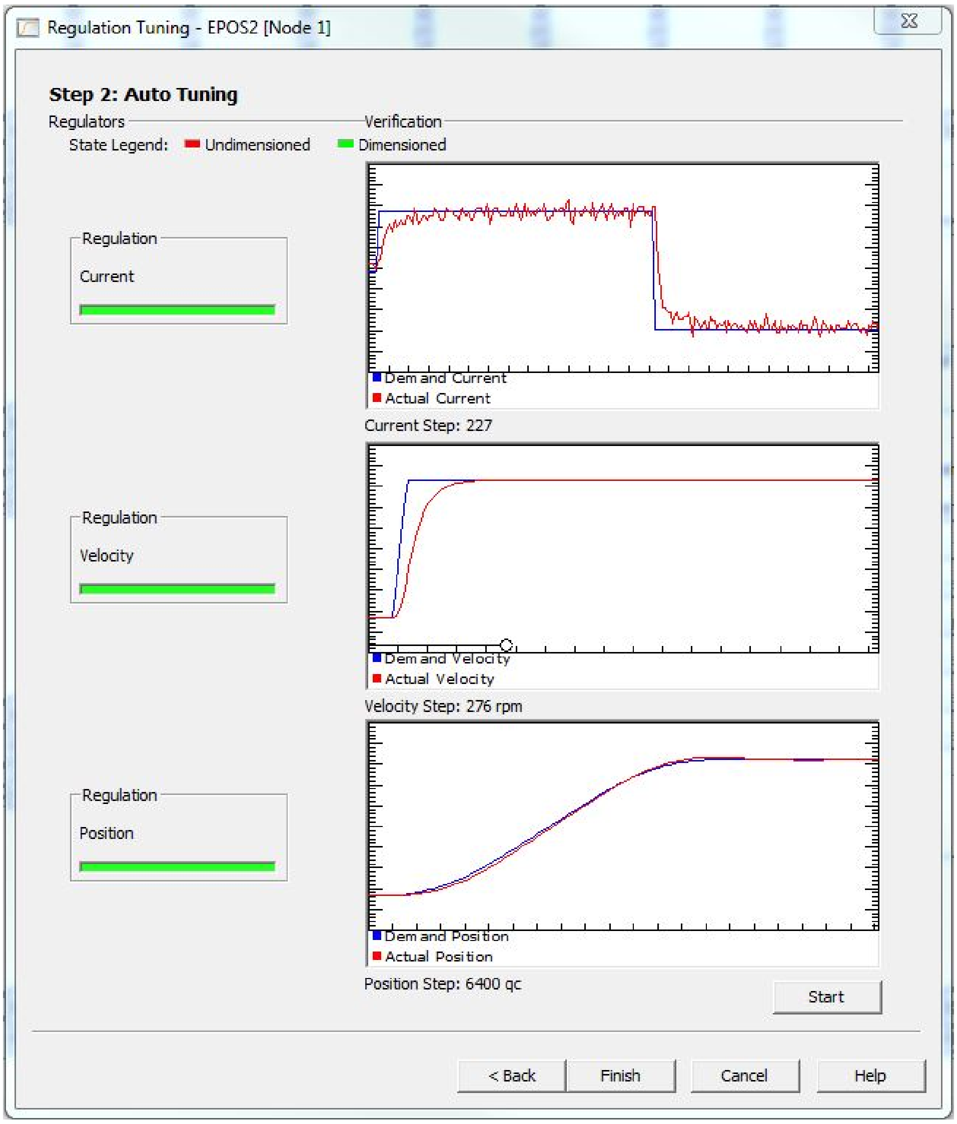
In this section, the tuning of the brushless motors is presented and discussed. The parameters obtained from the process are further implemented in the PLC controller as discusses in the dedicated chapter regarding the facility's control.

The MAXON® EPOS2 controller are able to tune the actuators according to three different controller strategies: current controllers, speed controllers, and position controllers.

The tuning parameters obtained are summarised in Tab. 11. In Fig. 133-137 the responses to the autotuning step inputs for the 6 motors are presented.

| *Joint* | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| *Motor model* | **EC 45** | **EC 60** | **EC 90** | **EC 45** | **EC 45** |
| Current gain | 227 | 200 | 227 | 200 | 200 |
| Velocity gain | 276 | 389 | 275 | 413 | 390 |
| Position gain | 6400 | 2048 | 6400 | 2048 | 2048 |

Table 11: Tuning parameters obtained with MAXON® EPOS2 autotuner.

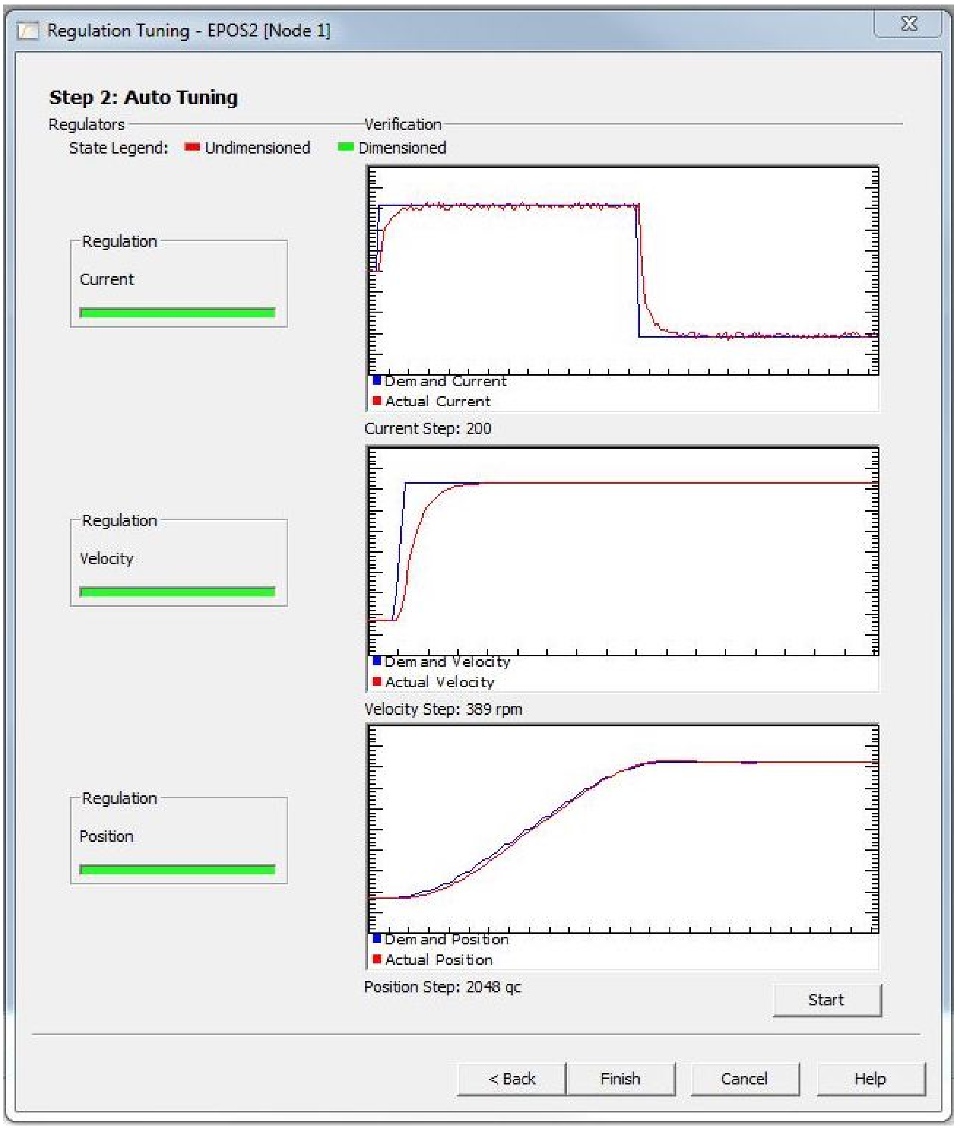


Current Step: 227

Velocity Step: 276 rpm

Position Step: 6400 qc

Figure 133: Motor 1 parameter tuning.



Regulation Tuning - EPOS2 [Node 1]

Step 2: Auto Tuning

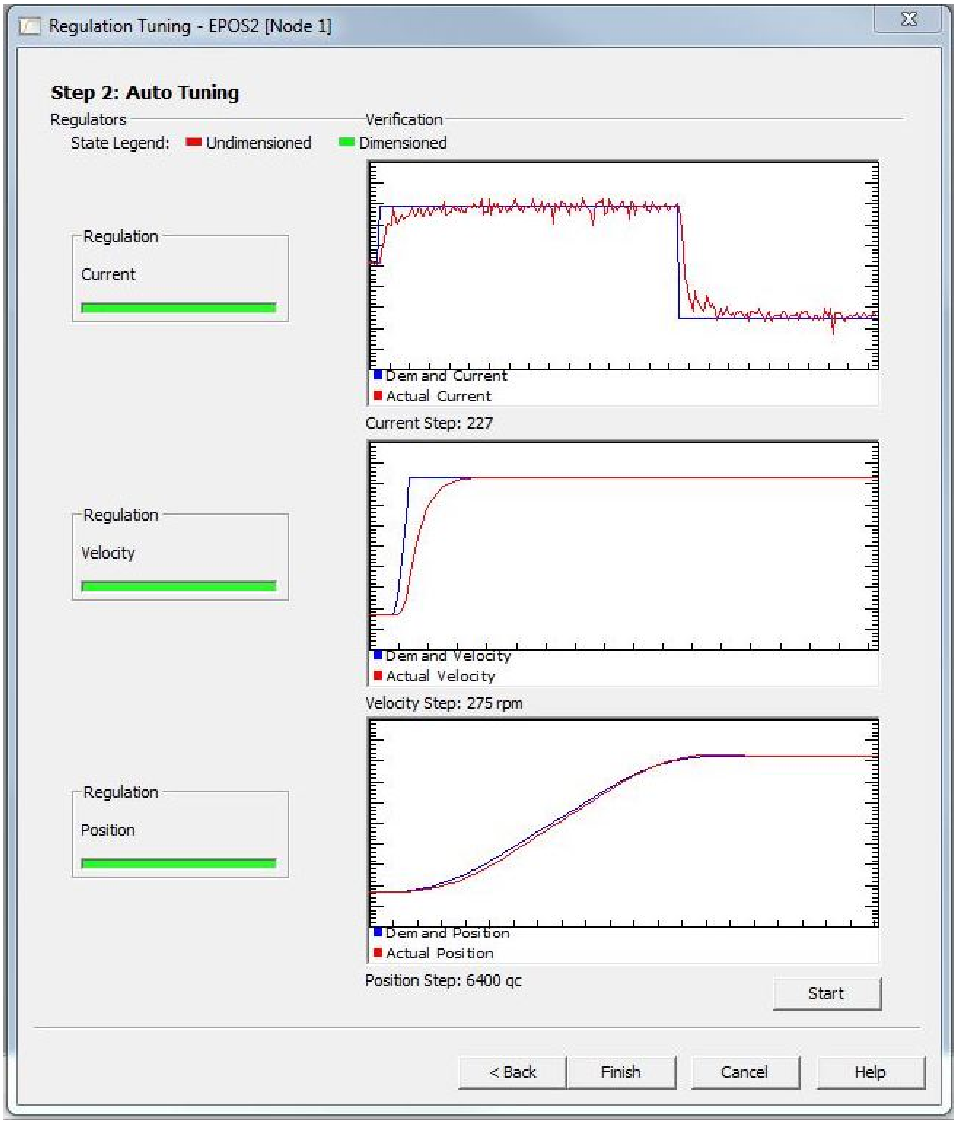
State Legend: 🟥 Undimensioned 🟩 Dimensioned

Current Step: 200

Velocity Step: 389 rpm

Position Step: 2048 qc

Figure 134: Motor 2 parameter tuning.



Regulation Tuning - EPOS2 [Node 1]

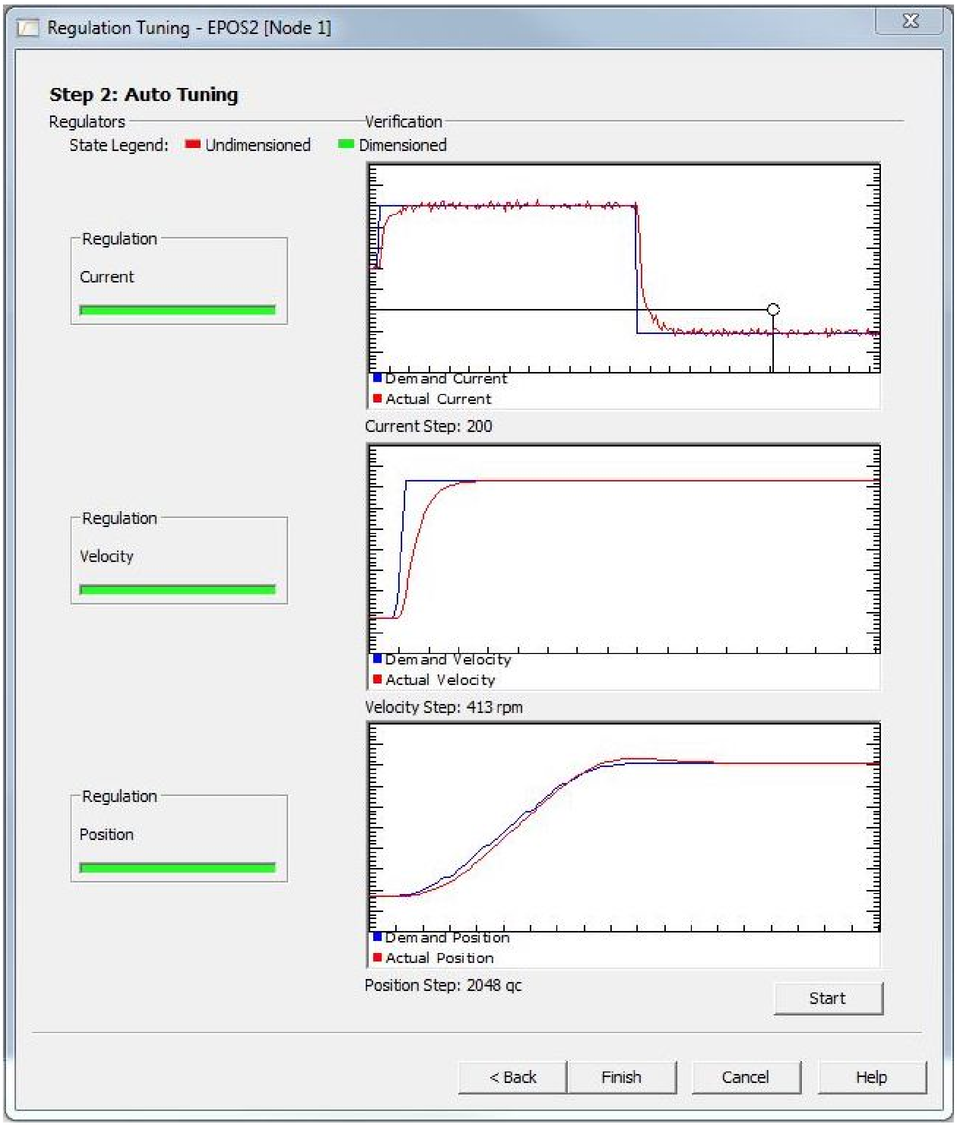
**Step 2: Auto Tuning**

Current Step: 227

Velocity Step: 275 rpm

Position Step: 6400 qc

Figure 135: Motor 3 parameter tuning.



Regulation Tuning - EPOS2 [Node 1]

**Step 2: Auto Tuning**

Regulators

Verification

State Legend: 🟥 Undimensioned 🟩 Dimensioned

Regulation: Current

Regulation: Velocity

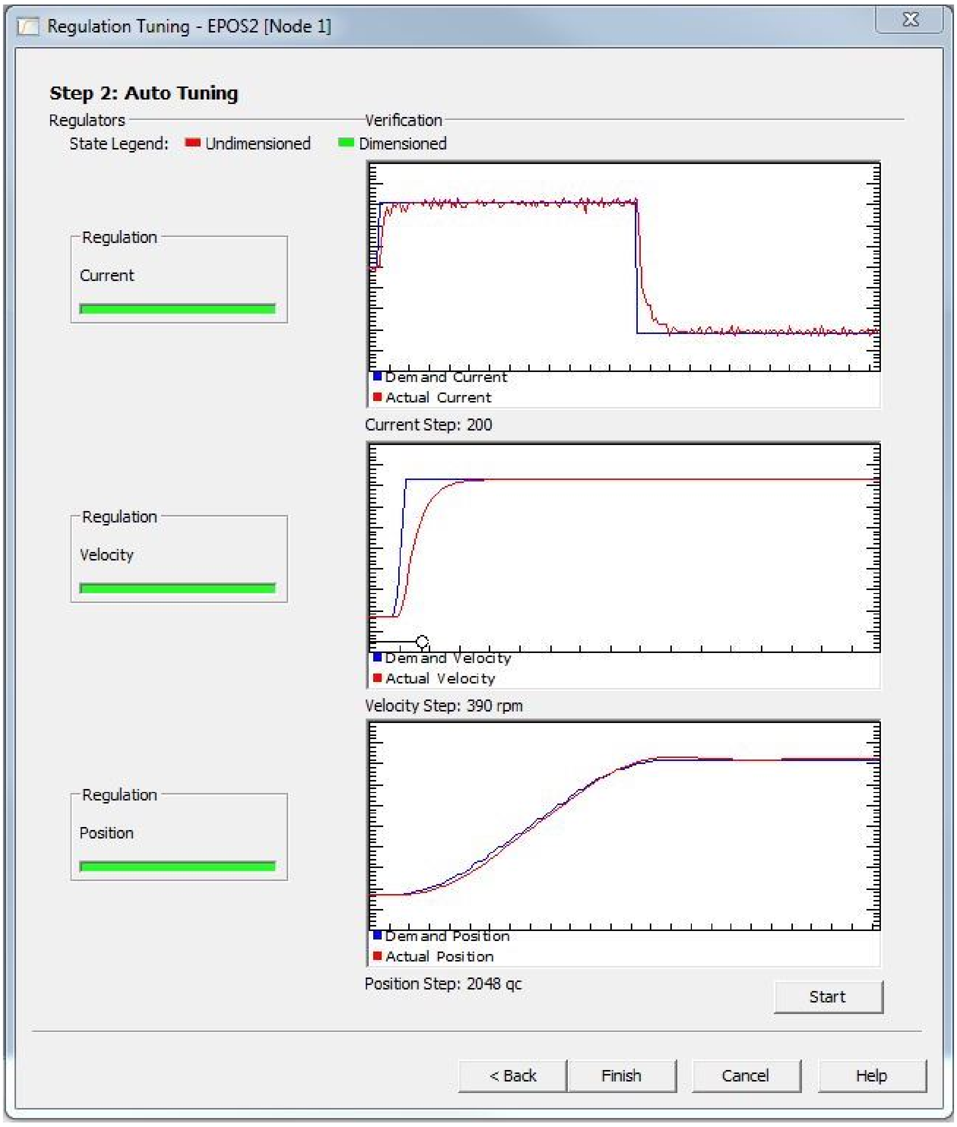
Regulation: Position

Current Step: 200

Velocity Step: 413 rpm

Position Step: 2048 qc

Figure 136: Motor 4 parameter tuning.



Regulation Tuning - EPOS2 [Node 1]

Step 2: Auto Tuning

Current Step: 200

Velocity Step: 390 rpm

Position Step: 2048 qc

Figure 137: Motor 5 parameter tuning.

# 7 SIMULATION SCENARIO

5.1 Following the chapter on the control design of the platform, we introduce in this section a framework of simulations that can be performed with this facility.

The main goal of the robotic arm, due to the availability of 6 degrees of freedom, is to perform OOS with the correct orbital dynamics in a laboratory environment. In the previous pages, care was given to the modeling of the contact in docking scenarios. The idea behind the work presented in this chapter (which resulted from a 10 months period spent between MIT and the Georgia Institute of Technology), is to provide the reader with a broader picture of the possibilities of the facility: this thesis presents not only a detailed design for specific operations (analysis of the contact forces, docking procedures, impacts, etc.) but yearns to provide a much wider framework in which the facility could be used as the main investigation tool.

## 7.1 STATE OF THE ART FACILITIES FOR ON ORBIT SERVICING SIMULATIONS

As discussed in the introduction, only a limited class of facilities are available for the simulation of OOS. Among these, 5 DOF simulators on low friction tables (in which the $z$ coordinate is constrained) have been developed by different research institutions in recent years [43, 44, 45].

One of the largest facilities is certainly the ASTROS platform devel-

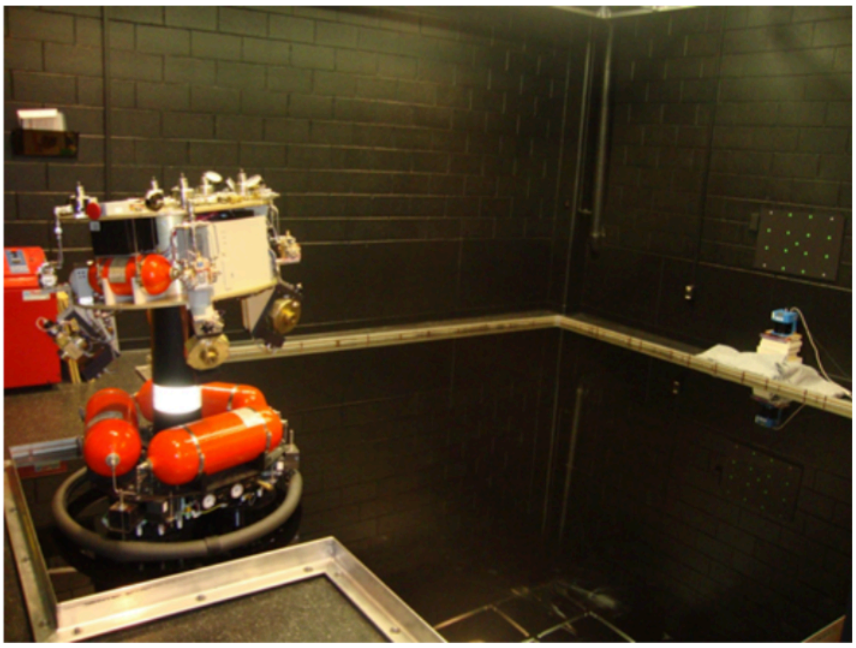


Figure 138: ASTROS facility at the Georgia Institute of Technology

oped at the Dynamics and Control Systems Laboratory under the direction of Prof. P. Tsiotras at Georgia Institute of Technology, which can be seen in Fig. 138 and 139 [46, 47]. The platform is divided in an upper and a lower stage: the lower stage consists of four high-pressure air storage vessels, three linear air-bearing pads, a hemispherical air-bearing and dedicated electronics that drive the solenoid valves for each air-bearing. The total volume of the external containers 3000 in$^3$, while the volume of the internal one is 360 in$^3$. The vessels are connected in series and are filled with compressed air at 3295 psi to provide air to both the linear and hemispherical air-bearings [46]. The upper stage simulates a typical spacecraft "bus" and has 12 thrusters in clusters of three, installed on the platform in a 3x4 configuration, each thruster providing a maximum of 5 N of force; in addition, four Variable-Speed CMGs (VSCMGs), arranged in a conventional inverted pyramid configuration, are used to provide fine attitude control.

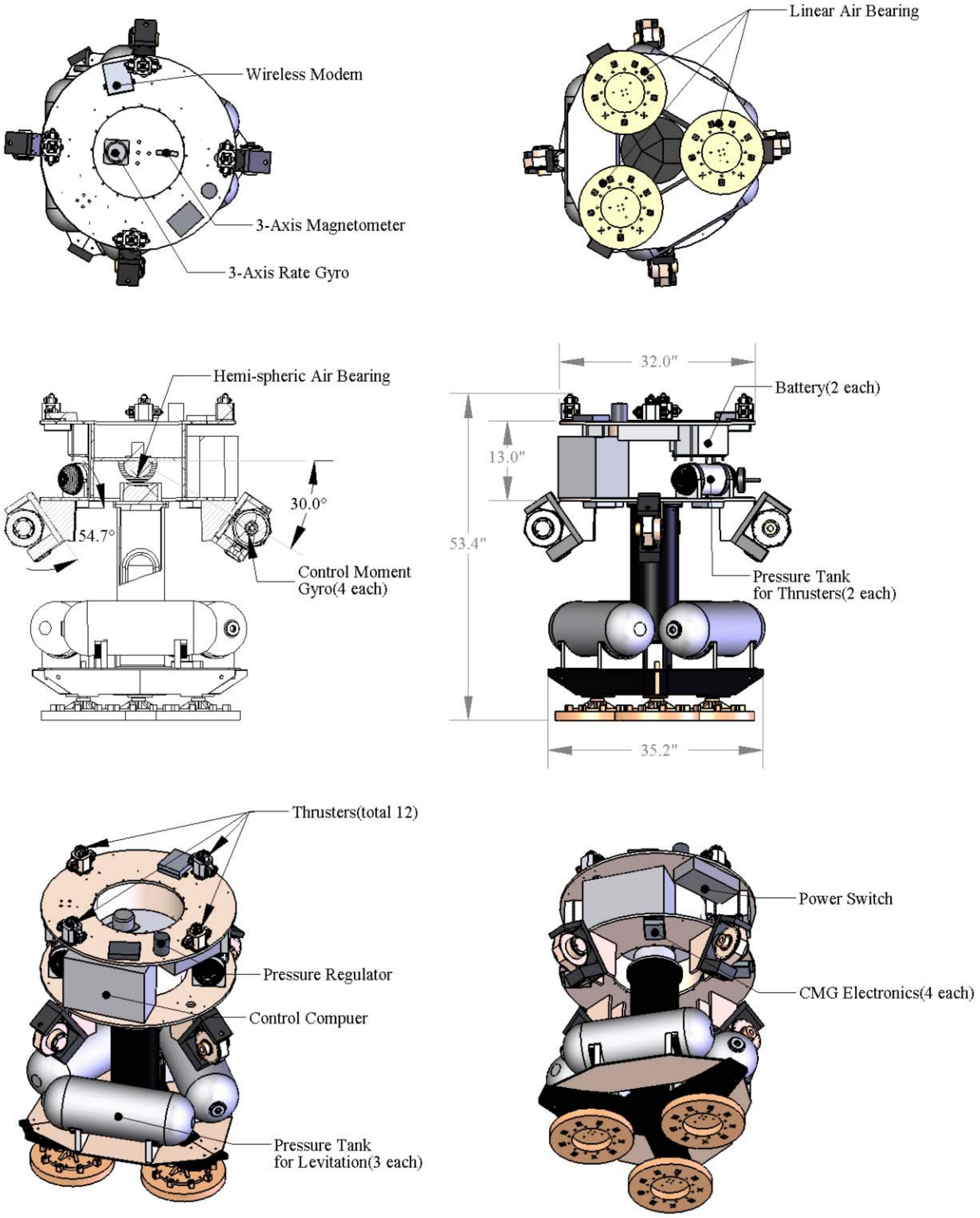


Figure 139: The ASTROS facility at the Georgia Institute of Technology, drawing.

The main objective of ASTROS is to test vision-based pose estimation, localization and relative navigation algorithms for proximity operations in space. In this framework, the author had the opportunity to work on a joint project with the Georgia Institute of Technology which resulted in the development of a novel proximity operation algorithm.

This algorithm is currently under implementation on the ASTROS platform and will serve as a the main GNC tool in OOS performed by the robotic arm facility.

## 7.2 SIMULATION: BASELINE ANALYSIS

Once the mechanics of the facility has been finalized, it is possible to simulate some trial trajectories [10]. These will constitute an initial campaign package to be performed by the arm; in addition, the results obtained from this first maneuvers have been used in an iterative fashion for the choice of the robot. That is, the initial mechanical designs were used to compute the required torque for the maneuvers that follow this introduction; hence, knowing the effort required in terms of torques and accelerations, the structure and actuators were corrected until an optimum (compatibly with the commercially available products) was found.

### 7.2.1 Rectilinear trajectory

The first trajectory to be simulated is a line in 3D space. According to the thorough derivation presented by this author in another work [10], this path needs only the starting and ending points ($\mathbf{p}_i$, $\mathbf{p}_f$) for its complete definition. As far as concerns the motion law, we suppose a $5^{th}$ degree polynomial with zero acceleration at the extremities.

Moreover, we need to define how the end effector orientation changes during the trajectory; since we have no particular requirements at this point, we impose the orientation to be coherent with a random attitude frame described using Euler angles $\phi_{att}$, $\theta_{att}$, $\psi_{att}$.

For sake of simplicity, we simulate a line parallel to the $x$-axis. The parameters used to initialize the code are:

$$\begin{aligned} x_{in} &= [0.4 \;\; 1.1 \;\; 0.2] \quad [m] \\ v_{in} &= [0 \;\; 0 \;\; 0] \quad [m] \\ \psi_{att} &= -20 \\ \theta_{att} &= 90 \\ \phi_{att} &= 45 \end{aligned}$$

As far as concerns the trajectory, we have:

$$x(t) = \begin{Bmatrix} a_5 t^5 + a_4 t^4 + a_3 t^3 + a_0 \\ 1.1 \\ 0.2 \end{Bmatrix} \tag{202}$$

$$\dot{x}(t) = \begin{Bmatrix} 5a_5 t^4 + 4a_4 t^3 + 3a_3 t^2 \\ 0 \\ 0 \end{Bmatrix} \tag{203}$$

Where:

$$\begin{aligned} a_0 &= -0.4 \\ a_1 &= 0 \\ a_2 &= 0 \\ a_3 &= -0.008 \\ a_4 &= 0.0012 \\ a_5 &= -4.8 \cdot 10^{-5} \end{aligned}$$

$$t \in [0;\, 10]\, s$$

$$dt = 0.01\, s$$

With the following boundary conditions on velocity and acceleration:

$$\dot{x}(0) = \dot{x}(T) = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} \tag{205}$$

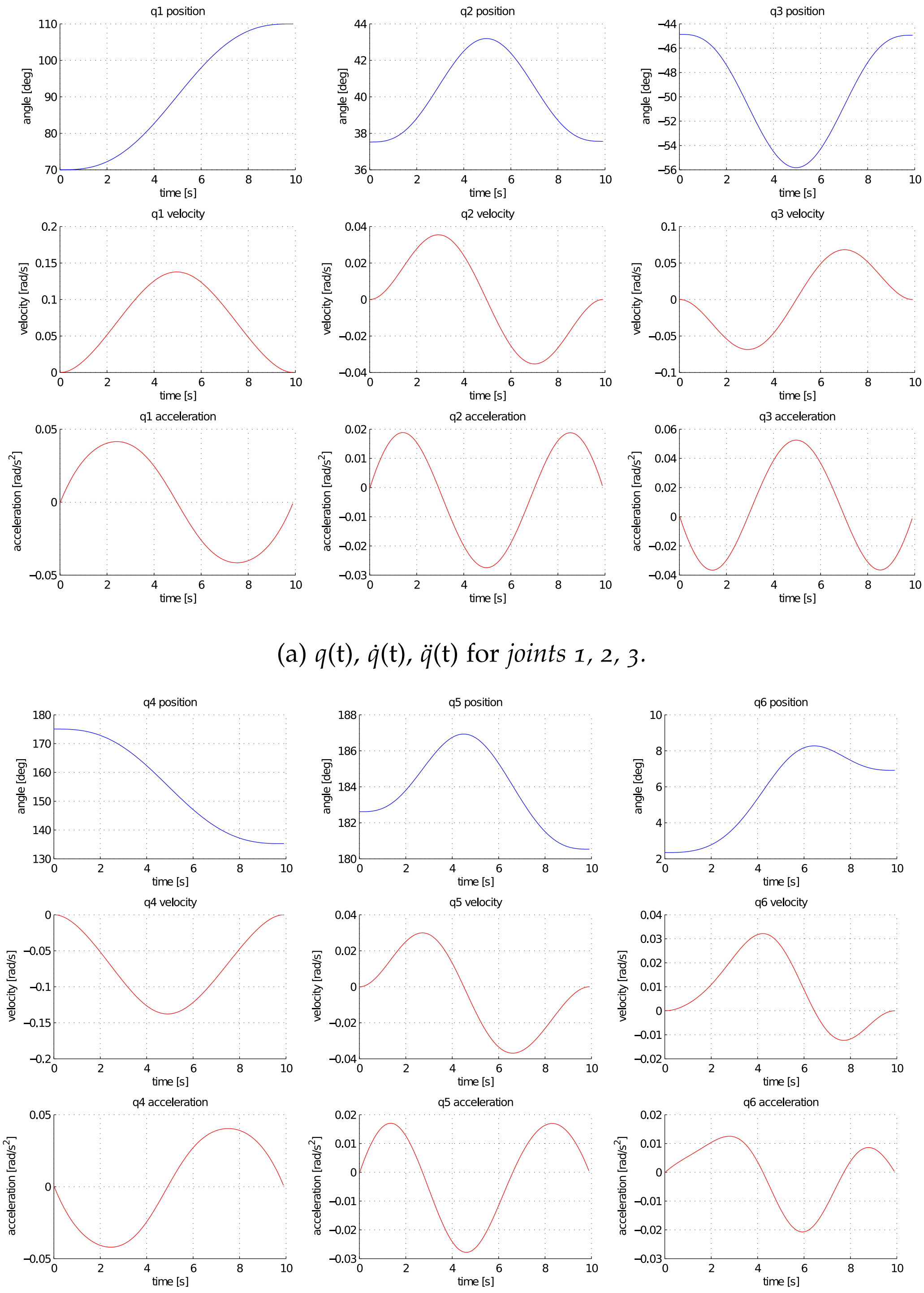
$$\ddot{x}(0) = \ddot{x}(T) = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} \tag{206}$$

Notice that, in the kinematics simulations, no information about the mass and inertial properties of the links are needed. In Fig. 140 the position, velocity and acceleration of the six joints are plotted.

The last frame of the MATLAB® animation is proposed in Fig. 141 *(a)*. The end effector is represented as a concentrated mass, and its orientation is expressed by the frame attached to it. To verify to correctness of the end effector orientation, the goal frame is plotted in the system's origin: it can be seen that they have the same attitude as predicted. The red line in the figure represents the actual trajectory, which follows very well the ideal path (represented with a blue segment, here hidden by the red line).

The importance of the time step choice can be seen in Fig. 141 *(b)*: in this case, the only parameter that was changed in the simulation is the time-step, which was increased to $dt = 0.1$ s. In this picture, the divergence between the red and the blue line is visibly increasing with time.

Following the dynamic analysis, it is straightforward to derive the required torques for these trajectories 142. The geometric parameters (mass, center of mass, inertia tensor) of each link are obtained from SOLIDWORKS® and are listed at the end of the mechanical design chapter. Two simulations are presented for the same trajectory, executed in T=10 s and T=1 s.



(a) $q(t)$, $\dot{q}(t)$, $\ddot{q}(t)$ for *joints* 1, 2, 3.

(b) $q(t)$, $\dot{q}(t)$, $\ddot{q}(t)$ for *joints* 4, 5, 6.

Figure 140: Kinematics analysis for linear trajectory, T=10 s

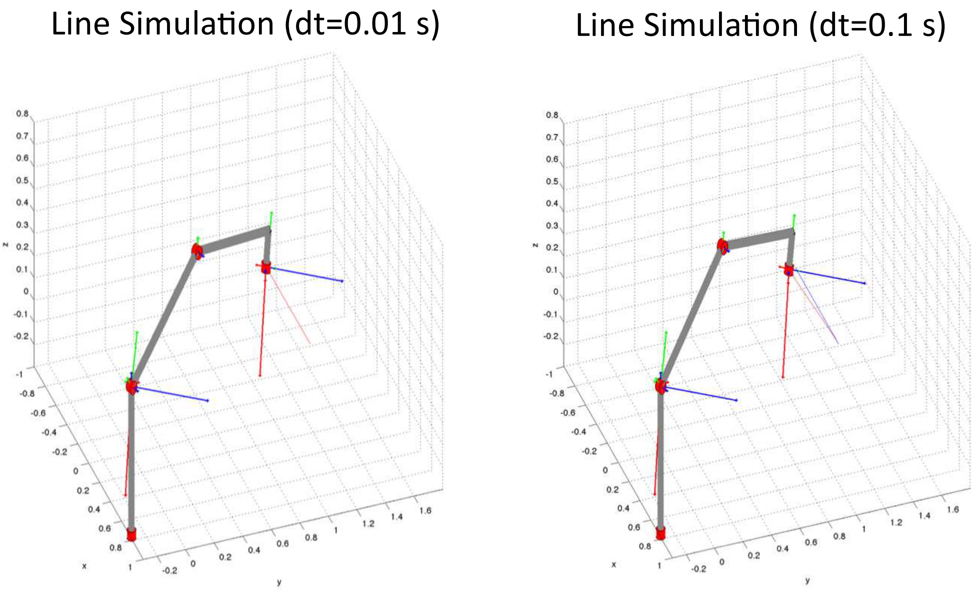
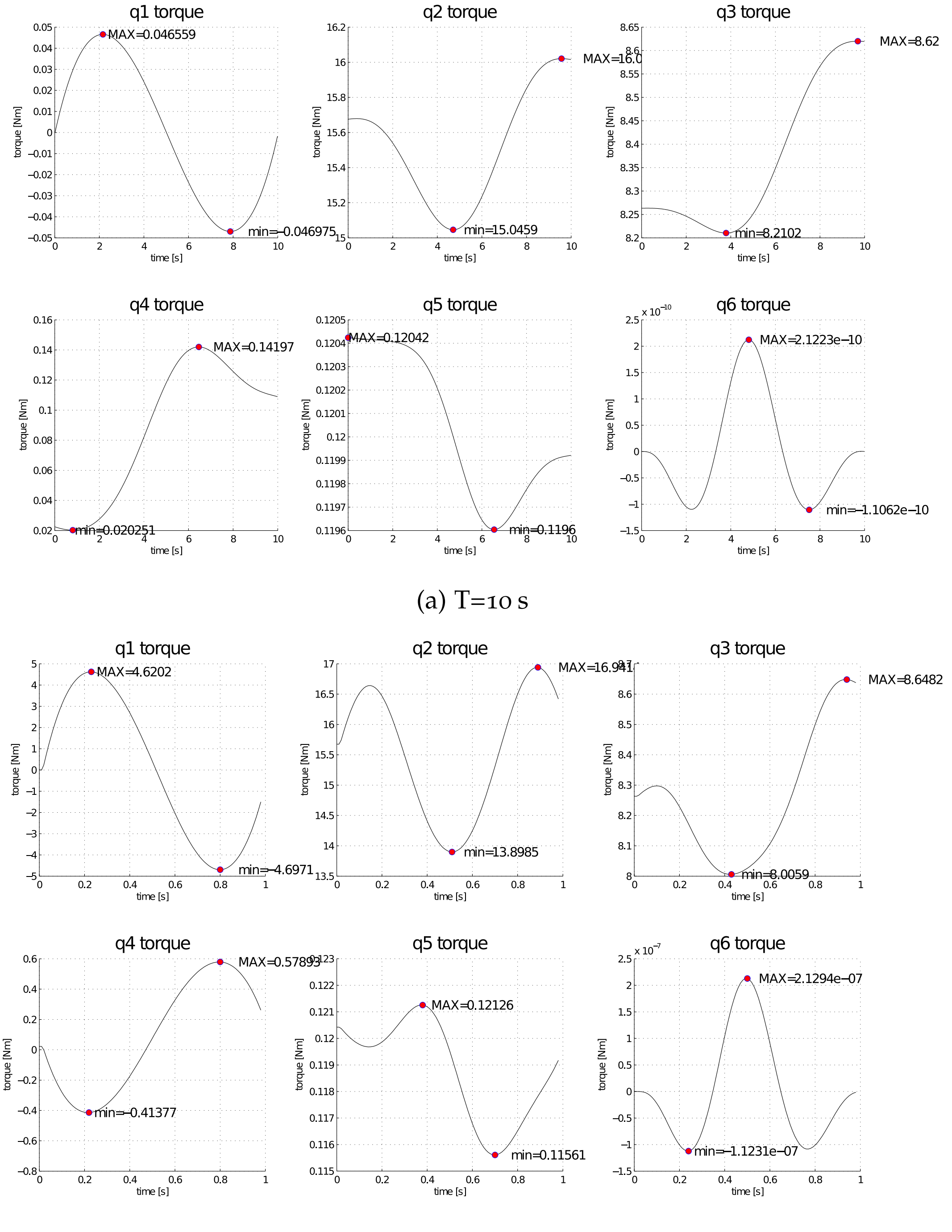


Figure 141: Linear trajectory: 3D simulation in Matlab® native environment. Time steps of $dt = 0.01$ s (left) and $dt = 0.1$ s (right)

### 7.2.2 Circular trajectory

For simulating a circular trajectory, we can follow the afore mentioned procedure. According to the analytical description in [10], a circular path is characterized by the radius, the center vector, the normal vector (i.e. the perpendicular to the circumference plane) and a starting point. In this example, the analysis was carried out with the following parameters:

$$
\begin{aligned}
x_{in} = x_{fin} &= [0.9 \;\; 0.7 \;\; 0.3] \quad [m] \\
x_c &= [0.9 - R \;\; 0.7 \;\; 0.3] \quad [m] \\
\bar{\mathbf{n}} &= [0 \;\; 0 \;\; 1] \\
R &= 0.4\, m \\
\psi_{att} &= 0 \\
\theta_{att} &= 45
\end{aligned}
$$



(a) T=10 s

(b) T=1 s

Figure 142: Torques required in the case of a circular trajectory, T=10 s and T=1 s

$$\phi_{att} = 0$$

As far as the trajectory is concerned, we utilize a 5th order polynomial for the angular law, that is:

$$\begin{aligned}\Theta(t) =& a_5 t^5 + a_4 t^4 + a_3 t^3 \\ \dot{\Theta}(t) =& 5a_5 t^4 + 4a_4 t^3 + 3a_3 t^2 \\ \ddot{\Theta}(t) =& 20a_5 t^3 + 12a_4 t^2 6 + a_3 t\end{aligned}$$

Where:

$$\begin{aligned}a_0 =& 0 \\ a_1 =& 0 \\ a_2 =& 0 \\ a_3 =& 62.83 \\ a_4 =& -94.25 \\ a_5 =& 37.69\end{aligned}$$

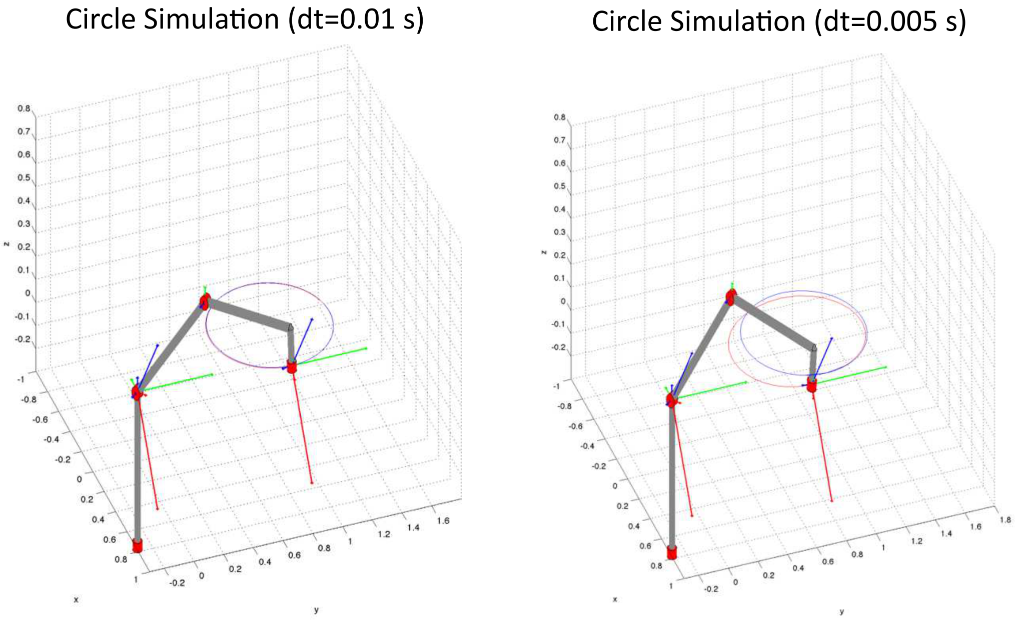
$$t \in [0;\, 1]\, s$$

$$dt = 0.001\, s$$

With the following boundary conditions on velocity and acceleration:

$$\dot{x}(0) = \dot{x}(T) = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix}$$

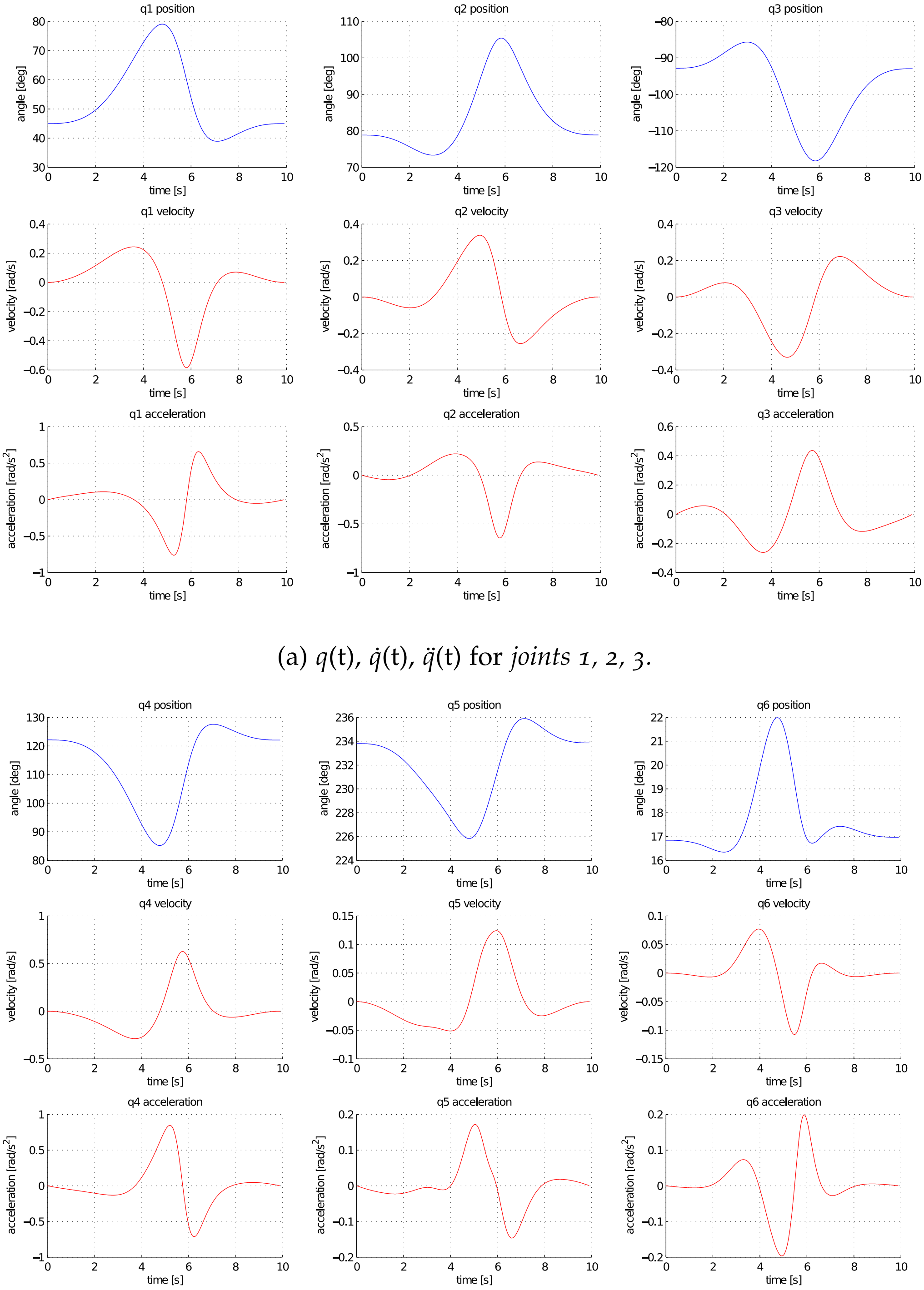
$$\ddot{x}(0) = \ddot{x}(T) = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix}$$



Circle Simulation (dt=0.01 s)

Circle Simulation (dt=0.005 s)

Figure 143: Circular trajectory: 3D simulation in MATLAB®'s native environment. Time steps of $dt = 0.001$ s (left) and $dt = 0.005$ s (right)



(a) $q(t)$, $\dot{q}(t)$, $\ddot{q}(t)$ for *joints 1, 2, 3*.

(b) $q(t)$, $\dot{q}(t)$, $\ddot{q}(t)$ for *joints 4, 5, 6*.

Figure 144: Kinematics analysis for circular trajectory, T=10 s

The cartesian position and velocity, then, are:

$$x(t) = \begin{Bmatrix} R\ \cos\Theta(t) \\ R\ \sin\Theta(t) \\ 0.3 \end{Bmatrix} = \begin{Bmatrix} 0.4\ \cos(a_5t^5 + a_4t^4 + a_3t^3) \\ 0.4\ \sin(a_5t^5 + a_4t^4 + a_3t^3) \\ 0.3 \end{Bmatrix} \tag{210}$$

$$\dot{x}(t) = \begin{Bmatrix} -R\ \sin\Theta(t) \cdot \dot{\Theta}(t) \\ R\ \cos\Theta(t) \cdot \dot{\Theta}(t) \\ 0 \end{Bmatrix} = \begin{Bmatrix} -0.4\ \sin(a_5t^5 + a_4t^4 + a_3t^3)(5a_5t^4 + 4a_4t^3 + 3a_3t^2) \\ 0.4\ \cos(a_5t^5 + a_4t^4 + a_3t^3)(5a_5t^4 + 4a_4t^3 + 3a_3t^2) \\ 0 \end{Bmatrix} \tag{211}$$

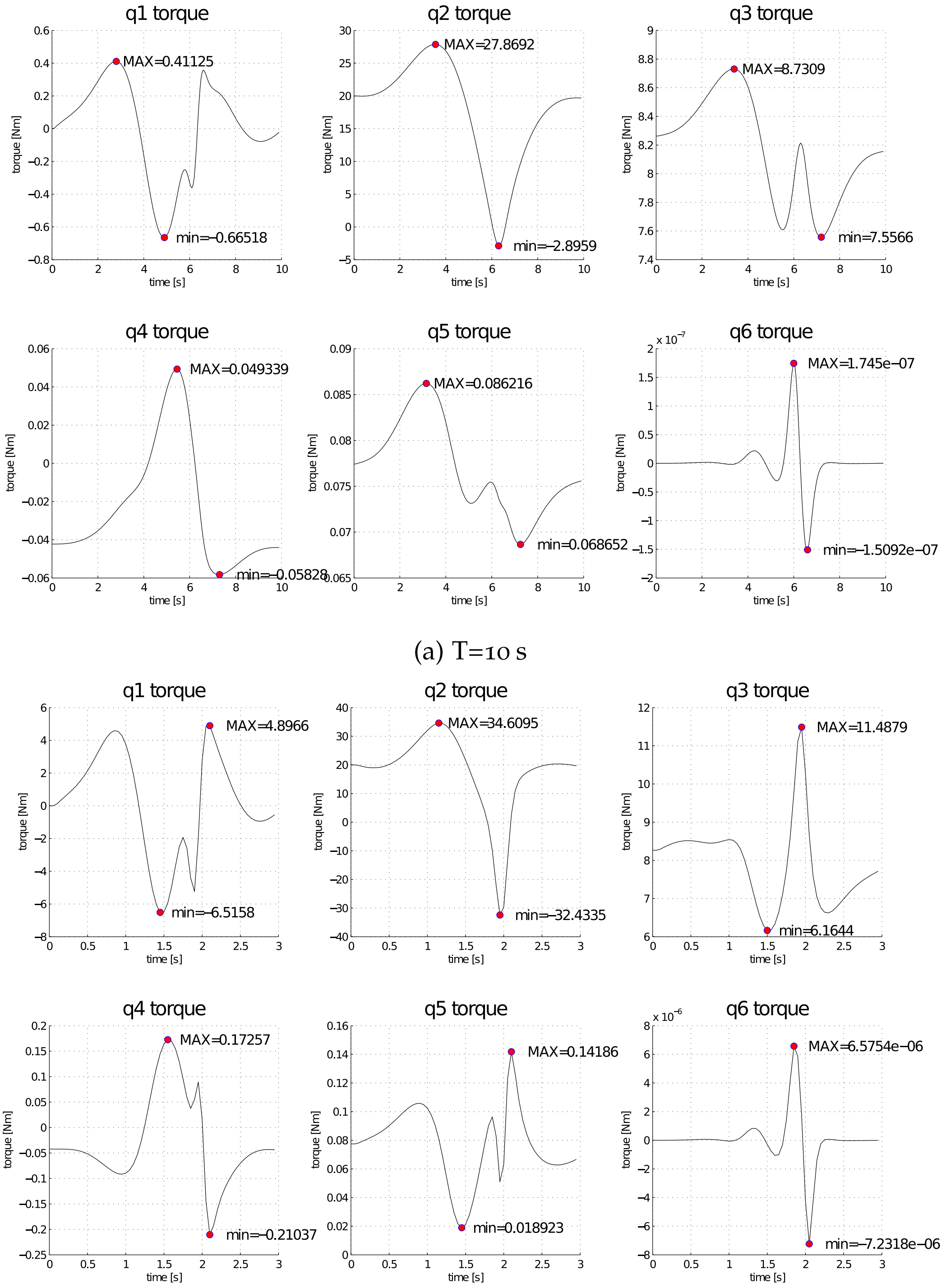
In Fig 144 the position, velocity and acceleration of the six joints are plotted.

Finally, we propose a screenshot of the animation carried out in this case. As usual, the red trajectory represents the actual position of the end effector, whereas the blue line is the goal trajectory. To stress the importance of the step size choice, two simulations are pictured. In the first one, Fig. 143 *(a)*, step size is $dt = 0.001$; in the second one Fig. 143 *(b)*, the value was increased to $dt = 0.005$. Note that also in this case the end effector attitude frame was plotted. In both picture, they are clearly coherent with the predefined attitude frame.

Again, it is straightforward to derive the required torques for this circular path 145. Two simulations are presented for the same trajectory, executed in T=10 s and T=3 s. All these results have been used directly in the aforementioned sizing process.

## 7.3 PROXIMITY OPERATIONS AND THE NEED FOR SELF-LOCALIZATION

Following from the previous section, we now focus on the application of the facility's capabilities to the problem of realistic simulations of OOS. Recent advantages in the field of computing hardware, coupled with the



(a) T=10 s

(b) T=3 s

Figure 145: Torques required in the case of a circular trajectory, T=10 s and T=3 s

enhancement of sensor performance, have paved the way for autonomous navigation to become a reality. In this framework, map generation and localization are the key for successful autonomous operation and navigation of robots. This is particularly true in the case of orbiting vehicles, in which autonomous formation flying and docking could enable new designs of space systems and enable OOS [48].

Autonomy depends on the capability for a satellite to accurately estimate its position with respect to another object. State-of-the-art proximity-navigation policies solve the problem of control and estimation separately. That is, the mutual effects the controller induces on the estimator (and vice versa) are not considered [49, 50, 51, 52, 53].

In the following pages, we depart from the separation principle of stochastic control, and integrate planning and stochastic optimization with localization in order to perform control of an autonomous spacecraft under uncertainty conditions.

Firstly, we approach the problem of a chaser satellite circumnavigating a target satellite in a simplified two-dimensional orbit. The chaser has a vision sensor and observes a set of landmarks on the target: its goal is to obtain a detailed map of these features. The control acts on the yaw-rotation of the sensor in order to maximize the time allocated to landmark observation.

A certain cost function (e.g., the estimation accuracy of the detected landmarks) drives which feature to be selected next, and hence also drives the next control action. An Extended Kalman Filter (EKF) provides the state uncertainty, which can then be used to design the cost function. Since the optimization problem is difficult to solve, we resort to cross-entropy (CE) minimization, which iteratively searches for the near-optimal control action. The final result is a trajectory that achieves the predefined goal in the state space and reduces total localization uncertainty, while limiting actuation cost.

Three different cost functions are proposed and simulated, and their

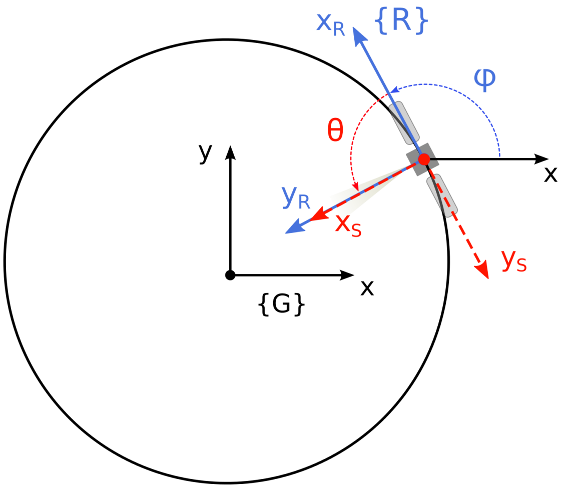


Figure 146: Problem set up and frame of reference definition

performances compared with the case of a fixed, nadir-pointing camera.

Eventually, we extended the analysis to more realistic case of a 3D scenario. In this part, the orbit will no longer be circular, but will be described by Clohessy Wiltshires equations (CW). For the sake of simulations, we consider the landmarks as single 3D points randomly located in a bounded box inside the relative orbit. In addition, the sensor will have a more realistic tridimensional field of view.

## 7.4 PROBLEM FORMULATION

### 7.4.1 Relative Navigation in Orbit

We consider an observer and a chaser satellite circumnavigating along a circular trajectory of radius $R_{\text{orb}}$ having linear velocity $V$ and orbital velocity $\omega_\phi = V/R_{\text{orb}}$. Typical relative orbits of two satellites flying in formation would result in an elliptical orbit [54].

The objective of the chaser satellite is to obtain a map of a certain set of landmarks that are present on the target satellite. These are features such as edges, patches, arrays of LEDs, etc. In this part, we are considering the landmarks as single points distributed in the $xy$ plane. The process of gathering information on the landmark positions is achieved through the application of a Simultaneous Localization and Mapping Algorithm (SLAM), which also allows for the simultaneous improvement of the chaser localization.

The satellite has an onboard sensor which is free to rotate around the axis that is normal to the $xy$ plane going through its center of mass. According to the notation in Fig. 146, frame {G} denotes the Global Frame[1], {R} the Local non Rotating Frame attached to the chaser and {S} the Local Rotating Frame attached to the satellite sensor. In addition, we define the angles $\phi$ and $\theta$, which respectively represent the heading direction of the satellite and the sensor bearing. Note that in this notation, frame {S} has a positive $\pi/2$ angular offset with respect to {R}: that is, when the sensor points to Nadir, the bearing is set to zero.

Detection of the landmarks — whose number and locations are to be determined — can be obtained with the aid of various sensors, i.e., sonars, lasers, LIDARs, cameras, ecc. Here, we use a range and bearing sensor, which outputs the distance and angular displacement of the detected feature in the {S} reference frame.

The sensor is a faithful representation of a real device, having a limited field of view, a fixed angular span and maximum angular acceleration.

[1] For example, {G} could represent the base frame of a Clohessy-Wilthshire transformation for a relative navigation problem.

### 7.4.2 State Model

The state model of the orbiting satellite, augmented with the position of the landmarks and expressed in differential form, is the following:

$$\begin{bmatrix} \mathrm{d}x(t) \\ \mathrm{d}y(t) \\ \mathrm{d}\phi(t) \\ \mathrm{d}\theta(t) \\ \mathrm{d}\mathbf{p}_1(t) \\ \vdots \\ \mathrm{d}\mathbf{p}_N(t) \end{bmatrix} = \begin{bmatrix} V\cos\phi(t)\mathrm{d}t \\ V\sin\phi(t)\mathrm{d}t \\ \omega_\phi(t)\mathrm{d}t \\ \omega_\theta(t)\mathrm{d}t \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{I}_4 \\ \mathbf{0}_{2N\times 4} \end{bmatrix} \mathrm{d}\mathbf{w}(t) \tag{212}$$

where $x$ and $y$ indicate the position of the chaser satellite in the {G} frame, whereas angles $\phi$ and $\theta$ are the rotation of the chaser and the sensor expressed in frames {G} and {R} respectively. The landmark positions are expressed in the global frame, and yield an augmented state $\mathbf{x} \in \mathbb{R}^{2N+4}$.

In the model, $\mathrm{d}\mathbf{w} \in \mathbb{R}^4$ represents Wiener process noise, with covariance matrix $\mathbf{\Sigma}_w = \mathrm{diag}(\sigma_1^2, \sigma_2^2, \sigma_3^2, \sigma_4^2)$.

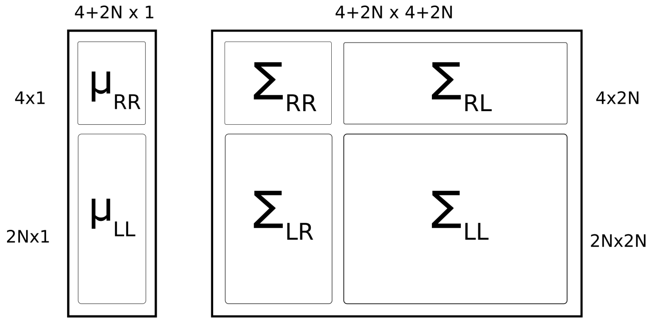
In the simulations, a discrete-time state model derived from Eq. 212 using Euler discretization will be used:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \phi_{k+1} \\ \theta_{k+1} \\ \mathbf{p}_{1k+1} \\ \vdots \\ \mathbf{p}_{Nk+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \phi_k \\ \theta_k \\ \mathbf{p}_{1k} \\ \vdots \\ \mathbf{p}_{Nk} \end{bmatrix} + \begin{bmatrix} V\cos(\phi_k)\,\delta t \\ V\sin(\phi_k)\,\delta t \\ \omega_{\phi_k}\,\delta t \\ \omega_{\theta_k}\,\delta t \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{I}_4 \\ \mathbf{0}_{2N\times 4} \end{bmatrix} \mathbf{w}(t_k) \tag{213}$$

where $\delta t$ is the discretization step and $\mathbf{w}(t_k)$ denotes white Gaussian noise. In this model, the only control parameter is the angular velocity of the sensor $\omega_{\theta_k}$ at time $t_k$. Since the sensor has a limited field of view, the capability of controlling $\omega_{\theta_k}$ may have a significant influence on the uncertainty reduction of the state estimate. We now introduce the Extend Kalman Filter that will be used in the algorithm [55, 56, 57, 58].

## 7.5 DESIGN OF THE KALMAN FILTER

We break up the analysis of the filter into the prediction and update parts. Let's first introduce the structure of the state vector and covariance matrix, where N is the number of landmarks:



As far as the prediction is concerned, we start by rewriting the predicted state of the system. We use just the first 4 rows of the state:

$$\hat{\mathbf{x}}_{t_{k+1}} = \mathbf{f}(\hat{\mathbf{x}}_{\mathbf{t_k}}) \tag{214}$$

this equation contains just the robot pose. The explicit state prediction equation is:

$$\hat{\mathbf{x}}_{t_{k+1}} = \hat{\mathbf{x}}_{t_k} + \mathbf{g}(x, y, \phi, \theta) \cdot \delta t \tag{215}$$

$$\begin{bmatrix} \hat{x}_{t_{k+1}} \\ \hat{y}_{t_{k+1}} \\ \hat{\phi}_{t_{k+1}} \\ \hat{\theta}_{t_{k+1}} \end{bmatrix} = \begin{bmatrix} \hat{x}_{t_k} \\ \hat{y}_{t_k} \\ \hat{\phi}_{t_k} \\ \hat{\theta}_{t_k} \end{bmatrix} + \begin{bmatrix} V \ \cos(\hat{\phi}_{t_k}) \\ V \ \sin(\hat{\phi}_{t_k}) \\ \omega_{\phi} \\ \omega_{\theta} \end{bmatrix} \cdot \delta t \tag{216}$$

The real state is simulated with the following expression:

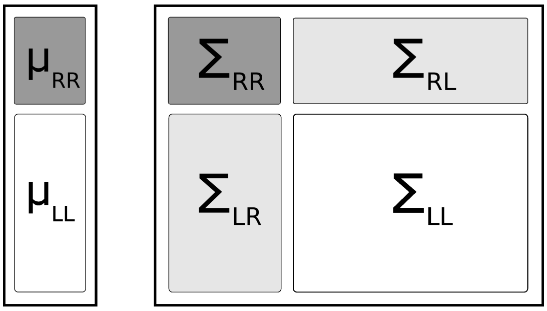
$$\mathbf{x}_{t_{k+1}} = \mathbf{x}_{t_k} + \mathbf{g}(x, y, \phi, \theta) \cdot \delta t + \mathbf{w}(t) \tag{217}$$

$$\begin{bmatrix} x_{t_{k+1}} \\ y_{t_{k+1}} \\ \phi_{t_{k+1}} \\ \theta_{t_{k+1}} \end{bmatrix} = \begin{bmatrix} x_{t_k} \\ y_{t_k} \\ \phi_{t_k} \\ \theta_{t_k} \end{bmatrix} + \begin{bmatrix} V \cos(\hat{\phi}_{t_k}) \\ V \sin(\hat{\phi}_{t_k}) \\ \omega_\phi \\ \omega_\theta \end{bmatrix} \cdot \delta t + \sqrt{\Sigma_w} \cdot rnd(4, 1) \tag{218}$$

Where $\mathbf{\Sigma}_w$ =diag$(\sigma_1^2, \sigma_2^2, \sigma_3^2, \sigma_4^2)$ is the covariance matrix of the process.

### 7.5.1 Covariance computation

To complete the prediction part of the EKF, the covariance matrix needs to be computed. The following diagram shows which parts of the covariance matrix are being updated:



We update the covariance matrix first by starting with the 4×4 robot block (dark grey).

$$\mathbf{\Sigma}_{RR}^{4\times4} = \mathbf{G}_{mot}\mathbf{\Sigma}_{RR}^{4\times4}\mathbf{G}_{mot}^{T} + \mathbb{I}^{4\times4} \cdot \mathbf{\Sigma}_w \cdot \mathbb{I}^{4\times4} \tag{219}$$

The $G_{mot}$ block is the Jacobian of the $\mathbf{f}(x, y, \phi, \theta)$ function.

$$\mathbf{G}_{mot} = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial \phi} & \frac{\partial f_1}{\partial \theta} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_4}{\partial x} & \frac{\partial f_4}{\partial y} & \frac{\partial f_4}{\partial \phi} & \frac{\partial f_4}{\partial \theta} \end{bmatrix} \tag{220}$$

$$\mathbf{G}_{mot} = \frac{\partial}{\partial(x, y, \phi, \theta)} \left\{ \begin{bmatrix} x \\ y \\ \phi \\ \theta \end{bmatrix} + \begin{bmatrix} V \ \cos(\phi) \cdot \delta t \\ V \ \sin(\phi) \cdot \delta t \\ \omega_\phi \cdot \delta t \\ \omega_\theta \cdot \delta t \end{bmatrix} \right\} \tag{221}$$

The explitic Jacobian is then:

$$\mathbf{G}_{mot} = \begin{bmatrix} 1 & 0 & -V\sin(\phi) \cdot \delta t & 0 \\ 0 & 1 & V\cos(\phi) \cdot \delta t & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{222}$$

The light-grey blocks of the covariance matrix concern the cross covariances between the robot state and the landmark locations. They are updated as follows:

$$\mathbf{\Sigma}_{RL}^{4\times 2N} = \mathbf{G}_{mot} \cdot \mathbf{\Sigma}_{RL}^{4\times 2N} \tag{223}$$

$$\mathbf{\Sigma}_{LR}^{2N\times 4} = (\mathbf{\Sigma}_{RL}^{4\times 2N})^T \tag{224}$$

### 7.5.2 Measurement model

Detection of the landmarks occurs only if they are within the field of view and range of the sensor. When a feature is detected, the sensor outputs $\mathbf{z} = (r, \alpha)$, where $r$ is the range and $\alpha$ is the bearing of the observed landmark. The measurement model, expressed in continuous form, is given by:

$$\mathbf{z}(t) = {}^{S}_{R}\mathbf{R}(\theta(t))^{R}_{G}\mathbf{R}(\phi(t))(\mathbf{p}_i(t) - \mathbf{p}_R(t)) + \mathbf{v}(t) \tag{225}$$

where $\mathbf{p}_i = (p_{x_i}, p_{y_i})$ and $\mathbf{p}_R = (x, y)$ are the position of the landmarks and the observer satellite, respectively. The term $\mathbf{v}(t)$ corresponds to the observation noise of the sensor which is considered zero-mean Gaussian with covariance matrix $\mathbf{\Sigma}_v = \text{diag}(\sigma^2_{\text{I}}, \sigma^2_{\text{II}})$. The matrices ${}^{S}_{R}\mathbf{R}(\theta(t))$ and ${}^{R}_{G}\mathbf{R}(\phi(t))$ express rotational transformations from the global {G} to the observer frame of reference {R} and from {R} to the sensor reference frame {S}, respectively. In compact form the observation model is written as:

$$\mathbf{z}(t) = \mathbf{h}(\mathbf{x}(t)) + \mathbf{v}(t). \tag{226}$$

However, in a real scenario, measurements will be taken discretely, according to the sampling strategy adopted. The measurement model, in discrete time form, can therefore be expressed as:

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k, \tag{227}$$

where $\mathbf{h}(\mathbf{x}_k)$ is given by:

$$\mathbf{h}(\mathbf{x}_k) = \begin{bmatrix} \sqrt{(x_L - x_k)^2 + (y_L - y_k)^2} \\ \tan^{-1}\left(\dfrac{y_L - y_k}{x_L - x_k}\right) - \phi_k - \theta_k \end{bmatrix} \tag{228}$$

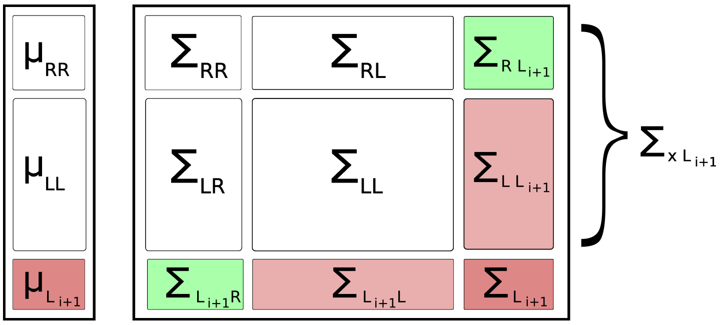
A matrix contains the location and a reference numbering of the N landmarks. As the simulation goes by, an extra row is added to this vector, containing a new progressive numbering which augments whenever a new landmark is detected. For example, if at step $k$ the landmarks that have been observed are, in order of appearance, 4, 10, and 2, the matrix will look like the following:

$$map = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} \\ y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 & y_8 & y_9 & y_{10} \\ \cdot & 3 & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & 2 \end{bmatrix} \tag{229}$$

And the state will be:

$$\hat{\mathbf{x}} = \begin{bmatrix} x \\ y \\ \phi \\ \theta \\ \hat{x}_{L_4} \\ \hat{y}_{L_4} \\ \hat{x}_{L_{10}} \\ \hat{y}_{L_{10}} \\ \hat{x}_{L_2} \\ \hat{y}_{L_2} \end{bmatrix} \tag{230}$$

Every time a landmark that has never been seen before is detected by the sensor, the state and the covariance matrices are augmented:



The state is augmented by plugging in the inverse measurement model $\chi$ the new measurement $z = [r, \alpha]$ along with the predicted state $\hat{\mathbf{x}}$:

$$\mu_{L_{i+1}} = \begin{bmatrix} x_{L_{i+1}} \\ y_{L_{i+1}} \end{bmatrix} = \chi(\hat{\mathbf{x}}, [r, \alpha]) = \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} + \begin{bmatrix} r\ \cos(\alpha + \hat{\phi} + \hat{\theta}) \\ r\ \sin(\alpha + \hat{\phi} + \hat{\theta}) \end{bmatrix} \tag{231}$$

The Jacobians of this model are:

$$\mathbf{G}_r = \frac{\partial \chi}{\partial(x, y, \phi, \theta)} \tag{232}$$

$$\mathbf{G}_r^{2\times 4} = \begin{bmatrix} 1 & 0 & -r\ \sin(\alpha + \phi + \theta) & -r\ \sin(\alpha + \phi + \theta) \\ 0 & 1 & r\ \cos(\alpha + \phi + \theta) & r\ \cos(\alpha + \phi + \theta) \end{bmatrix} \tag{233}$$

$$\mathbf{G}_y = \frac{\partial \chi}{\partial(r, \alpha)} \tag{234}$$

$$\mathbf{G}_y^{2\times 2} = \begin{bmatrix} \cos(\alpha + \phi + \theta) & -r \sin(\alpha + \phi + \theta) \\ \sin(\alpha + \phi + \theta) & r \cos(\alpha + \phi + \theta) \end{bmatrix} \tag{235}$$

The updated matrix is then:

$$\mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma}_{old} & \mathbf{\Sigma}'_{x,L_{i+1}} \\ \mathbf{\Sigma}_{x,L_{i+1}} & \mathbf{\Sigma}_{L_{i+1}} \end{bmatrix} \tag{236}$$

If the landmark has previously been detected and added to the state, an algorithm compares the new measurement to the previous one stored in the state. Let's suppose we are measuring landmark $j$. The update code is as follows:

$$\begin{bmatrix} \delta x \\ \delta y \end{bmatrix} = \begin{bmatrix} \hat{x}_{L_j} - \hat{x} \\ \hat{y}_{L_j} - \hat{y} \end{bmatrix} \tag{237}$$

$$r = \sqrt{\delta x^2 + \delta y^2} \tag{238}$$

$$\hat{\mathbf{z}}_{t_k} = \begin{bmatrix} \sqrt{\delta x^2 + \delta y^2} \\ atan2\left(\frac{\delta y}{\delta x}\right) - \phi - \theta \end{bmatrix} \tag{239}$$

Let's compute the Jacobian of the observation model with respect to the robot pose and the observed landmark coordinates. For observation $i$, the Jacobian becomes[2]:

$$\mathbf{H}_i^{2\times size(\hat{X})} = \frac{\partial\,\mathbf{z}}{\partial(x, y, \phi, \theta, x_{L,i}, y_{L,i})} \tag{240}$$

$$\mathbf{H}_i = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial \phi} & \frac{\partial f_1}{\partial \theta} & \dots & \frac{\partial f_1}{\partial x_{L,i}} & \frac{\partial f_1}{\partial y_{L,i}} & \dots \\ \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial \phi} & \frac{\partial f_2}{\partial \theta} & \dots & \frac{\partial f_2}{\partial x_{L,i}} & \frac{\partial f_2}{\partial y_{L,i}} & \dots \end{bmatrix} \tag{241}$$

Using the previously defined $\delta x$, $\delta y$ and $r$, the Jacobian becomes:

$$\mathbf{H}_i = \begin{bmatrix} -\frac{\delta x}{r} & \frac{\delta y}{r} & 0 & 0 & \dots & -\frac{\delta x}{r} & \frac{\delta y}{r} & \dots \\ \\ \frac{\delta y}{r^2} & -\frac{\delta x}{r^2} & -1 & 1 & \dots & -\frac{\delta y}{r^2} & \frac{\delta x}{r^2} & \dots \end{bmatrix} \tag{242}$$

In brief:

$$\mathbf{H}_i = \begin{bmatrix} \mathbf{H}_{r,i} & \dots & -\mathbf{H}_{r,i} & \dots & \end{bmatrix} \tag{243}$$

Let's suppose we observe at time $t$ landmarks 4 and 2 (or 1 and 3 in the new numbering of the example in the previous pages). Hence, since in this case $size(z) = 2$, the stacked Jacobian will be of the form[3]:

$$\mathbf{H}^{2\cdot|z|\times|\hat{x}|} = \left[\begin{array}{c|ccc} \mathbf{H}_{r,1} & -\mathbf{H}_{r,1} & 0 & 0 \\ \mathbf{H}_{r,3} & 0 & 0 & -\mathbf{H}_{r,3} \end{array}\right] \tag{244}$$

For each measure, we build the vector $\mathbf{v}$, which stacks together the innovation vectors $\mathbf{v}_i = \mathbf{z}_i - \hat{\mathbf{z}}_i$ for all the current available landmarks:

$$\mathbf{v}_{2\cdot|z|} = \begin{bmatrix} \{\mathbf{z}_{i,1} - \hat{\mathbf{z}}_{i,1}\} \\ \{\mathbf{z}_{i,3} - \hat{\mathbf{z}}_{i,3}\} \end{bmatrix}$$

[2] Index $i$ represents the progressive numbering of the discovered landmarks: in Eq. 230, landmark 4 will have i=1, landmark 10 will have 2 and so on. Index $i$ is then equal to the size of the state minus the robot state, in this case 4

[3] We simplify the notation of $size(z)$ with $|z|$

The Kalman gain is defined as:

$$\mathbf{Z} = \mathbf{H}_{2\cdot|z|\times|\hat{x}|} \cdot \Sigma_{|\hat{x}|\times|\hat{x}|} \cdot \mathbf{H}^{T}_{|\hat{x}|\times 2\cdot|z|} + \mathbf{R}_{2\cdot\{|z|\times|z|\}} \tag{245}$$

$$\mathbf{K}_{|\hat{x}|\times 2\cdot|z|} = \Sigma_{|\hat{x}|\times|\hat{x}|} \cdot \mathbf{H}^{T}_{|\hat{x}|\times 2\cdot|z|} \cdot \mathbf{Z}^{-1}_{2\cdot\{|z|\times|z|\}} \tag{246}$$

In which:

$$\mathbf{R}_{2\cdot\{|z|\times|z|\}} = \begin{bmatrix} \mathbf{\Sigma}_{v}^{2\times 2} & 0 \\ 0 & \mathbf{\Sigma}_{v}^{2\times 2} \end{bmatrix}$$

The update step of the EKF is then:

$$\hat{\mathbf{x}}_{new} = \hat{\mathbf{x}} + \mathbf{K}_{|\hat{x}|\times 2\cdot|z|} \cdot \mathbf{v}_{2\cdot|z|} \tag{247}$$

$$\Sigma_{new} = \Sigma - \mathbf{K}_{|\hat{x}|\times 2\cdot|z|} \cdot \mathbf{Z} \cdot \mathbf{K}^{T}_{2\cdot|z|\times|\hat{x}|} \tag{248}$$

## 7.6 MAIN PROBLEM

We want to estimate the position of the landmarks by evaluating the measurements taken by the sensor. To do this, we control the rotation of the sensor in the $x$-$y$ plane to maximize the performance over a finite time horizon. The objective is to minimize a cost function that encloses both the final uncertainty of the estimate and the actuation cost. The cost function can be written as:

$$\mathcal{L}(\mathbf{x}, \mathbf{u}) = \|e^{2}(t_N)\| + \int_{0}^{t_N} \left( Q(\mathbf{x}) + \frac{1}{2}\mathbf{u}^{T} R \mathbf{u} \right) \mathrm{d}t \tag{249}$$

where $\|e^{2}(t_N)\|$ is the terminal cost at a certain time horizon setpoint $T = t_N$. Since we do not know this error, a strategy for its approximation needs to be designed.

This strategy is obtained by approximating the error with a measure of the estimation uncertainty. We introduce a strategy based on the covariance matrix trace, along with alternative strategies based deriving from the observation of the landmarks. All these strategies evaluate both the terminal performances of the piecewise control trajectory and the actuation cost needed to achieve it.

### 7.6.1 Trace of the covariance matrix (TCM)

The first strategy uses the trace of the covariance matrix as a measure of the uncertainty for the state estimate given by an Extended Kalman Filter (EKF) [55, 56, 57, 58]:

$$\mathcal{L}(\mathbf{x},\mathbf{u}) = \psi^1(\mathbf{x}_{t_N}) + \sum_{k=0}^{N}\left(Q(\mathbf{x}_{t_N}) + \frac{1}{2}\mathbf{u}(\mathbf{t_k})^T R\mathbf{u}(\mathbf{t_k})\right) \tag{250}$$

in which the terminal cost is:

$$\psi^1(\mathbf{x}_{t_N}) = \text{trace}(\Sigma(t_N)). \tag{251}$$

For simplicity, let in the following $Q(x) = 0$ to obtain:

$$\mathcal{L}_{\text{tcm}} = \text{trace}(\Sigma(t_N)) + \sum_{k=0}^{N}\left(\frac{1}{2}\mathbf{u}(\mathbf{t_k})^T R\mathbf{u}(\mathbf{t_k})\right) \tag{252}$$

### 7.6.2 Time under observation (TUO)

In this second strategy, the cost is defined as the time under observation of the landmarks by the sensor. For each sampled trajectory, we count the number of landmarks seen by the sensor at each iteration ($T_i$). The total number of observed landmarks is then summed:

$$\psi^2(\mathbf{x}_{t_N}) = \sum_{i=1}^{N} T_i \tag{253}$$

The complete function, taking into account the actuation cost, is then:

$$\mathcal{L}_{\text{tuo}} = \sum_i T_i + \sum_{k=0}^{N}\left(\frac{1}{2}\mathbf{u}(\mathbf{t_k})^T R\mathbf{u}(\mathbf{t_k})\right) \tag{254}$$

### 7.6.3 Modified Time under observation (MTUO)

The previous strategy maximizes the time under observation of the target, but may lead to overlooking features in relatively remote areas of the working space. In order to avoid this situation, we modify Eq. 256 by defining the trajectory cost based not only on the TUO, but also on the number of different landmarks observed. That is, we count the TUO and the number of different landmarks observed for each iteration ($N_{\text{lnd}_i}$) and we define MTUO as the normalized sum:

$$\psi^3(\mathbf{x}_{t_N}) = \sum_i (\hat{T}_i + \hat{N}_{\text{lnd}_i}) \tag{255}$$

Normalization is mandatory in order to correctly compare and sum the two partial costs. We normalize the two terms as follows:

$$\hat{T}_i = \frac{T_i - T_{i,\min}}{T_{i,\max} - T_{i,\min}}$$

$$\hat{N}_{\text{lnd}_i} = \frac{N_{\text{lnd}_i} - N_{\text{lnd}_{i,\min}}}{N_{\text{lnd}_{i,\max}} - N_{\text{lnd}_{i,\min}}}$$

So that $\{\hat{T}_i, \hat{N}_{\text{lnd}_i}\} \in [0,1]$ and the new cost $\mathcal{L}_{\text{mtuo}} \in [0,2]$.

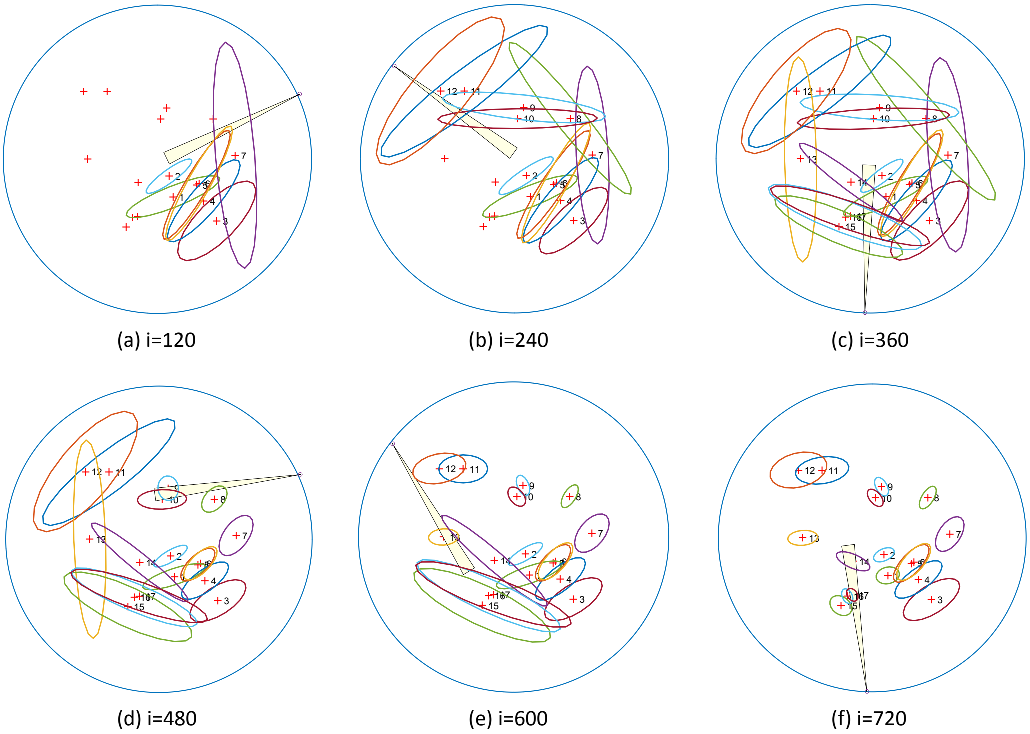
The complete function, taking into account the actuation cost, is then:

$$\mathcal{L}_{\text{mtuo}} = \sum_i [\hat{T}_i + \hat{N}_{\text{lnd}_i}] + \sum_{k=0}^{N} \left( \frac{1}{2} \mathbf{u}(\mathbf{t_k})^T R \mathbf{u}(\mathbf{t_k}) \right) \tag{256}$$

### 7.6.4 Cross Entropy Minimization

We now present the Cross Entropy minimization algorithm, and show how it can be used to solve a certain class of stochastic optimal control problems. Assume that the following stochastic dynamics system is given:

$$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{u})\delta t + g(x)\mathrm{d}\mathbf{w} \tag{257}$$


(a) i=120

(b) i=240

(c) i=360

(d) i=480

(e) i=600

(f) i=720

Figure 147: Recognition phase (a-c) and Cross Entropy minimization (d-f) at different time-steps: note the progressive numbering assigned to the landmarks by the algorithm. The uncertainty is displayed as a covariance ellipse.

where $\mathbf{x} \in \mathbb{R}^n$ is the state of the system, $\mathbf{u} \in \mathbb{R}^p$ is the control input, and $\mathbf{w} \in \mathbb{R}^l$ is a zero-mean Gaussian Wiener process with covariance $\boldsymbol{\Sigma}_w$. Our objective is to minimize a cost function of the form:

$$\min \mathbb{E}_p[\mathcal{L}(\mathbf{x}, \mathbf{u})], \tag{258}$$

where the expectation in Eq. 258 is with respect to the trajectories of Eq. 257. Assuming that $\mathbf{u}(t)$ depends on a parameter vector $\lambda \in \mathbb{R}^m$, we can rewrite the control input as $\mathbf{u}(t; \lambda)$. The result of this parametrization is that we will minimize the cost function with respect to the finite dimensional parameters vector $\lambda$. According to the CE minimization method [59], we rewrite the cost function as follows:

$$J(\lambda) = \mathbb{E}_p\,[\mathcal{L}(\lambda)] = \int p(\lambda)\,\mathcal{L}(\lambda)\mathrm{d}\lambda \tag{259}$$

where $p(\lambda)$ is the probability density function corresponding to sampling trajectories based on Eq. 257. Under a parameterization of the baseline probability density, we have that $p(\lambda) = p(\lambda;\mu)$. We perform importance sampling from a proposal probability density $q(\lambda)$ and evaluate the expectation in Eq. 259 as follows:

$$\mathcal{J}(\lambda) = \int \frac{p(\lambda;\mu)}{q(\lambda)}\,\mathcal{L}(\lambda)\,q(\lambda)d\lambda = \mathbb{E}_p\left[\frac{p(\lambda;\mu)}{q(\lambda)}\mathcal{L}(\lambda)\right]$$

The expression above can be approximated numerically from

$$\hat{\mathcal{J}}(\lambda) \approx \frac{1}{N_s}\sum\left[\frac{p(\lambda;\mu)}{q(\lambda)}\mathcal{L}(\lambda)\right] \tag{260}$$

with $\hat{\mathcal{J}}$ being an unbiased estimator and $N_s$ the number of samples drawn. The probability density that minimizes the variance of the estimator $\hat{\mathcal{J}}$ is:

$$q^*(\lambda) = \underset{q}{\operatorname{argmin}}\ \mathrm{Var}\left[\frac{p(\lambda;\mu)}{q(\lambda)}\mathcal{L}(\lambda)\right] = \left[\frac{p(\lambda;\mu)\mathcal{L}(\lambda)}{\mathcal{J}(\lambda)}\right] \tag{261}$$

and it is the optimal (with respect to variance) importance sampling density. The goal of CE is to find the parameters $\psi \in \Psi$ in the parametric class of pdfs $p(\lambda;\psi)$, such that the probability distribution $p(\lambda;\psi)$ approaches the optimal distribution $q^*(\lambda)$ given in Eq. 261.

The optimal parameters can be approximated numerically using the Kullback-Leibler divergence as a distance metric between $q^*(\lambda)$ and $p(\lambda;\psi)$ yields

$$\mathbb{D}(q^*(\lambda)||p(\lambda;\psi)) = \int q^*(\lambda)\,ln[q^*(\lambda)]\,d\lambda - \int q^*(\lambda)\,ln[p(\lambda;\psi)]\,d\lambda \tag{262}$$

The minimization problem can now be formulated as follows:

$$\begin{aligned}
\psi^* &= \underset{\psi}{\operatorname{argmin}}\, \mathbb{D}(q^*(\lambda)||p(\lambda;\psi)) \\
&= \underset{\psi}{\operatorname{argmax}} \int q^*(\lambda)\, ln[p(\lambda;\psi)]\, d\lambda \\
&= \underset{\psi}{\operatorname{argmax}} \int \frac{p(\lambda;\mu)\,\mathcal{L}(\lambda)}{\mathcal{J}(\lambda)}\, ln[p(\lambda;\psi)]\, d\lambda \\
&= \underset{\psi}{\operatorname{argmax}} \int p(\lambda;\mu)\,\mathcal{L}(\lambda)\, ln[p(\lambda;\psi)]\, d\lambda \\
&= \underset{\psi}{\operatorname{argmax}}\, \mathbb{E}_{p(\lambda;\mu)}\, [\mathcal{L}(x)\, ln[p(\lambda;\psi)]
\end{aligned}$$

Based on the previous equation, the optimal parameters can be approximated numerically as:

$$\psi^* \approx \underset{\psi}{\operatorname{argmax}} \frac{1}{N_s} \sum \mathcal{L}(\lambda)\, ln[p(\lambda;\psi)] \tag{263}$$

We want to compute the value of $\lambda$ that satisfies the following equation:

$$\mathbb{P}(\mathcal{L} \leq \epsilon) = \mathbb{E}_{p(\lambda;\mu)}[I_{\{\mathcal{L}\leq\epsilon\}}] \tag{264}$$

where $\epsilon$ is a small constant and $I$ is the indicator function. Using Eq. 260, this probability can be numerically approximated:

$$\hat{\mathbb{P}}(\mathcal{L} \leq \epsilon) \approx \frac{1}{N_s} \sum \left[ \frac{p(\lambda_i;\mu)}{p(\lambda_i;\psi)} I_{\{\mathcal{L}(\lambda_i)\leq\epsilon\}} \right]$$

where $\lambda_i$ are i.i.d samples drawn from $p(\lambda;\psi)$. Based on Eq. 263, the goal is to find the optimal $\psi^*$, which is defined as:

$$\psi^* \approx \underset{\psi}{\operatorname{argmax}} \frac{1}{N_s} \sum I_{\{\mathcal{L}(\lambda_i)\leq\epsilon\}}\ \ln[p(\lambda_i;\psi)], \tag{265}$$

where now the samples $\lambda_i$ are generated according to probability density $p(\lambda;\mu)$. In order to estimate the above probability, it is infeasible to use a brute force method, e.g. Monte-Carlo [60]: this is because the event

$\{\mathcal{L} \leq \epsilon\}$ is rare. An alternative is to start with $\epsilon_1 > \epsilon$ for which the probability of the event $\{\mathcal{L} \leq \epsilon_1\}$ is equal to some $\rho > 0$. Thus, the value $\epsilon_1$ is set to the $\rho$-th quantile of $\mathcal{L}(\lambda)$ which means that $\epsilon_1$ is the largest number for which:

$$\mathbb{P}(\mathcal{L}(\lambda) \leq \epsilon_1) = \rho.$$

The parameter $\epsilon_1$ can be found by sorting the samples according to their cost in increasing order and setting $\epsilon_1 = \mathcal{L}_{\rho N}$. The optimal parameter $\psi_1$ for the level $\epsilon_1$ is calculated according to Eq. 265 using the value $\epsilon_1$. This iterative procedure terminates when $\epsilon_k \leq \epsilon$, in which case the corresponding parameter $\psi_k$ is the optimal one and thus $\psi^* = \psi_k$.

To summarize, in order to find the optimal trajectory $\lambda^*$ and the corresponding optimal parameters $\psi_k$, the process of estimating rare event probabilities is iterated until $\epsilon \to \epsilon^*$, where $\epsilon^* = \min \mathcal{L}(\epsilon)$. Since $\epsilon^*$ is not known a priori, we choose as $\epsilon^*$ the value of $\epsilon$ for which no further improvement within a predefined tolerance in the iterative process is observed. The overall problem is summarized in the table below.

The proposed algorithm consists of two phases:

- the *recognition* phase, during which the first orbit the measurements taken by the chaser provide a first estimation of all the landmarks,
- the *incremental estimation* phase, during which the chaser keeps taking measurements in order to improve the overall state estimation.

The recognition phase is necessary, since the chaser does not know the number and the position of the landmarks and, in turn, the dimension of the overall system state. During the recognition phase the chaser runs the aforementioned Extended Kalman filter algorithm whose state is augmented whenever a measurement related to a new landmark is collected [55, 56, 57, 58].

### 7.6.5 Recognition Phase

Let $\tilde{N}$ be the number of landmarks recognized up to the time instant $k$ so the current state of the EKF is given by:

$$\mathbf{x}_k^{(\tilde{N})} = \begin{bmatrix} x_k & y_k & \phi_k & \theta_k & \mathbf{p}_k^{(1)} & \mathbf{p}_k^{(2)} & \dots & \mathbf{p}_k^{(\tilde{N})} \end{bmatrix}^T.$$

We divide the design of the EKF into prediction and update steps.

*Prediction step*

The update equation is

$$\begin{bmatrix} \hat{x}_{k+1|k} \\ \hat{y}_{k+1|k} \\ \hat{\phi}_{k+1|k} \\ \hat{\theta}_{k+1|k} \\ \hat{p}^{(1)}_{k+1|k} \\ \hat{p}^{(2)}_{k+1|k} \\ \vdots \\ \hat{p}^{(\tilde{N})}_{k+1|k} \end{bmatrix} = \begin{bmatrix} \hat{x}_{k|k} \\ \hat{y}_{k|k} \\ \hat{\phi}_{k|k} \\ \hat{\theta}_{k|k} \\ \hat{p}^{(1)}_{k|k} \\ \hat{p}^{(2)}_{k|k} \\ \vdots \\ \hat{p}^{(\tilde{N})}_{k|k} \end{bmatrix} + \begin{bmatrix} V\cos(\hat{\theta}_{k|k})\delta t \\ V\sin(\hat{\theta}_{k|k})\delta t \\ \omega_{\phi_k}\delta t \\ \omega_{\theta_k}\delta t \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

or, in a more compact form,

$$\hat{\mathbf{x}}^{(\tilde{N})}_{k+1|k} = \mathbf{f}(\hat{\mathbf{x}}^{(\tilde{N})}_{k|k}, \omega_{\phi_k}, \omega_{\theta_k}).$$

The update of the covariance matrix is given by

$$P^{(\tilde{N})}_{k+1|k} = F_k P^{(\tilde{N})}_{k|k} F_k^T + Q_k, \tag{266}$$

where

$$F_k = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}. \tag{267}$$

The matrix $F_k$ has the following structure

$$F = \begin{bmatrix} F_k^{\mathrm{mot}} & 0 \\ 0 & \mathbb{I}_{\tilde{N}} \end{bmatrix}, \tag{268}$$

where $F_k^{\mathrm{mot}}$ is given by the following expression

$$\begin{aligned} \mathbf{F}_k^{\mathrm{mot}} &= \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial \phi} & \frac{\partial f_1}{\partial \theta} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_4}{\partial x} & \frac{\partial f_4}{\partial y} & \frac{\partial f_4}{\partial \phi} & \frac{\partial f_4}{\partial \theta} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & -V\sin(\hat{\phi}_{k|k})\delta t & 0 \\ 0 & 1 & V\cos(\hat{\phi}_{k|k})\delta t & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned} \tag{269}$$

Finally the matrix $Q_k$ is of the form

$$Q_k = \begin{bmatrix} \Sigma_w & 0 \\ 0 & 0_N \end{bmatrix},$$

where $0_N$ is a null matrix of dimension $N$.

*Update step*

Assuming we have the information provided by the range and bearing sensor $\mathbf{z} = [r, \alpha]$, we collect multiple measurements at the same time instant $k+1$, e.g. $\bar{\mathbf{z}}_{k+1}$. This vector can be divided in two components, the first component $\mathbf{z}_{k+1}^{(1)}$ which is given by all the measurements collected from already seen landmarks, and the second $\mathbf{z}_{k+1}^{(2)}$ which represents measurements collected by observing new landmarks. The measurement model can be written as

$$\bar{\mathbf{z}}_{k+1} = \begin{bmatrix} \mathbf{z}_{k+1}^{(1)} \\ \mathbf{z}_{k+1}^{(2)} \end{bmatrix} = \begin{bmatrix} \mathbf{h}^{(1)}(\hat{\mathbf{x}}_{\mathbf{k+1}}) + v_{k+1}^{(1)} \\ \mathbf{h}^{(2)}(\hat{\mathbf{x}}_{\mathbf{k+1}}) + v_{k+1}^{(2)} \end{bmatrix}.$$

Let us compute the Jacobian of the observation model with respect to the robot pose and the observed landmark coordinates. At iteration $k+1$ we obtain

$$\mathbf{H}_{k+1} = \left.\frac{\partial \mathbf{h}_{k+1}^{(1)}}{\partial \mathbf{x}^{(\tilde{N})}}\right|_{\hat{x}_{k+1|k}} \tag{270}$$

With the output matrix $\mathbf{H}_{k+1}$ we can update the state related to all chaser attitute and all the already seen landmarks

$$\begin{aligned} K_{k+1} &= P_{k+1|k}^{(\tilde{N})} H_{k+1}^{T} \left( H_{k+1} P_{k+1|k}^{(\tilde{N})} H_{k+1}^{T} + R_{k+1} \right)^{-1} \\ x_{k+1|k+1}^{(\tilde{N})} &= x_{k+1|k}^{(\tilde{N})} + K_{k+1} \mathbf{z}_{k+1}^{(1)}. \\ P_{k+1|k+1}^{(\tilde{N})} &= (\mathbb{I} - K_{k+1} H_{k+1}) P_{k+1|k}^{(\tilde{N})} \end{aligned}$$

Without loss of generality, suppose that $\mathbf{z}_{k+1}^{(2)}$ refers to just one new landmark $p^{(\tilde{N}+1)}$, then we have that

$$\hat{p}_{k+1|k+1}^{(\tilde{N}+1)} = \begin{bmatrix} \hat{x}_{k+1|k} \\ \hat{y}_{k+1|k} \end{bmatrix} + \begin{bmatrix} r\cos(\alpha + \hat{\phi}_{k+1|k} + \hat{\theta}_{k+1|k}) \\ r\sin(\alpha + \hat{\phi}_{k+1|k} + \hat{\theta}_{k+1|k}) \end{bmatrix}.$$

Then we can extend the state

$$x_{k+1|k+1}^{(\tilde{N}+1)} = \begin{bmatrix} x_{k+1|k+1}^{(\tilde{N})} \\ \hat{p}_{k+1|k+1}^{(\tilde{N}+1)} \end{bmatrix},$$

and the covariance matrix

$$P_{k+1|k+1}^{(\tilde{N}+1)} = \begin{bmatrix} P_{k+1|k+1}^{(\tilde{N})} & P^{(\tilde{N},\tilde{N}+1)} \\ P^{(\tilde{N}+1,\tilde{N})} & P^{(\tilde{N}+1)} \end{bmatrix},$$

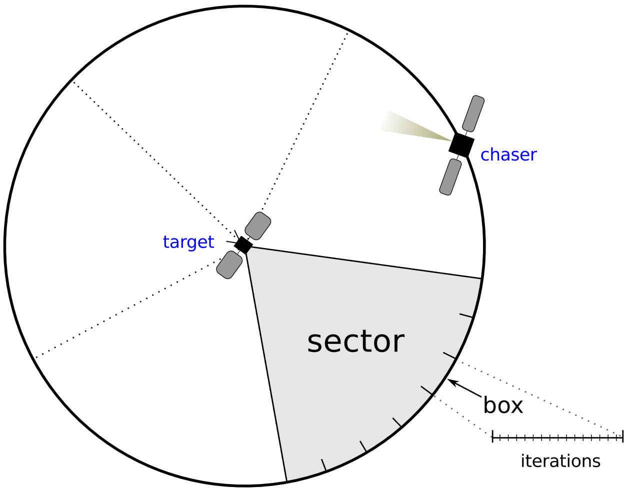


Figure 148: Discretization strategy for cross-entropy algorithm

where

$$P^{(\tilde{N}+1,\tilde{N})} = \left(P^{(\tilde{N},\tilde{N}+1)}\right)^T = \left.\begin{bmatrix} \dfrac{\partial \hat{p}^{(\tilde{N}+1)}_{k+1|k+1}}{\partial x_k} \\ \dfrac{\partial \hat{p}^{(\tilde{N}+1)}_{k+1|k+1}}{\partial y_k} \\ \dfrac{\partial \hat{p}^{(\tilde{N}+1)}_{k+1|k+1}}{\partial \phi_k} \\ \dfrac{\partial \hat{p}^{(\tilde{N}+1)}_{k+1|k+1}}{\partial \theta_k} \end{bmatrix}\right|_{(\hat{x}_{k+1|k},\bar{\mathbf{z}}_{k+1})} .$$

and

$$P^{(\tilde{N}+1)} = \left.\frac{\partial \hat{p}^{(\tilde{N}+1)}_{k+1|k+1}}{\partial \mathbf{z}_{k+1}}\right|_{(\hat{x}_{k+1|k},\bar{\mathbf{z}}_{k+1})} .$$

### 7.6.6 Incremental Estimation Phase

After the recognition phase an initial guess of the landmark's position is stored in the state of the system. At this point the core of the algorithm runs to improve the estimate of the state, and this information is exploited to control the vision sensor.

Specifically, for any orbit the following steps are repeated:

1. We draw $N_{\text{traj}}$ random possible acceleration trajectories for the sensor, $\lambda = \{\lambda_1, \lambda_2, \dots, \lambda_{N_{traj}}\}$, from a Gaussian distribution with parameter $v_i$ .

2. For all $\lambda$ we simulate the behavior of the camera running an Extended Kalman filter.

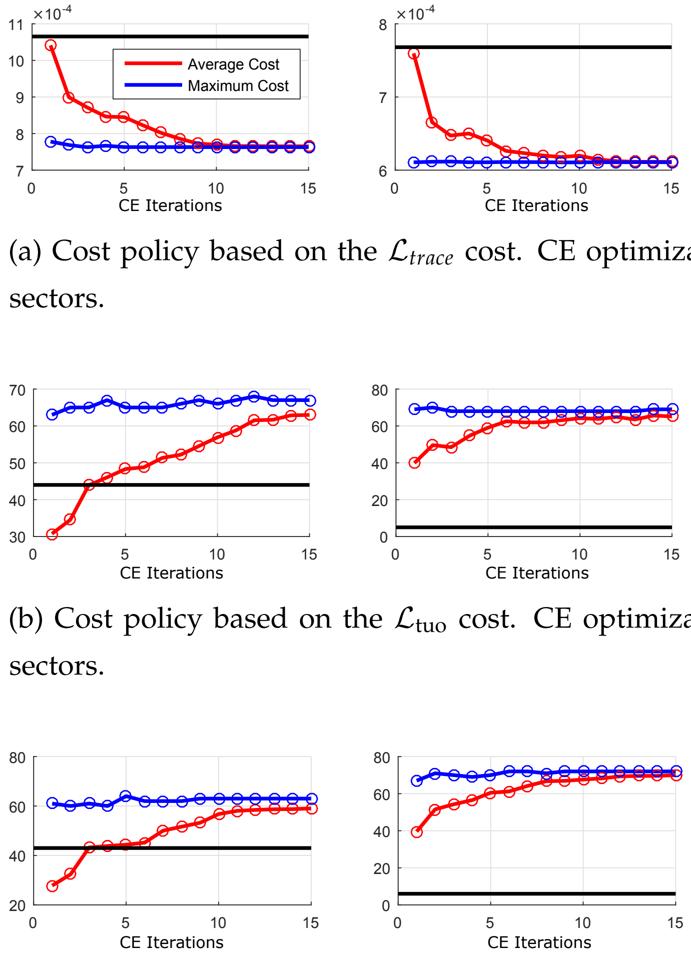
3. Once the state has been estimated at any time instant we can evaluate one of the cost function presented in Section 7.6 and perform the CE algorithm. Basically we have to select the $\rho - th$ best performing percentile, i.e. the trajectories with an associated lower cost.

4. From these reduced subset of samples the new parameters for the distribution are inferred. The aforementioned procedure is repeated up to the convergence of the cross entropy method and then the optimal solution is applied.

### 7.6.7 Controller

The controller acts on the angular velocity of the sensor, $\omega_\theta$. Recalling Eq. 254, we can rewrite the discretized cost as:

$$\hat{\mathcal{L}}(\mathbf{x}, \mathbf{u}) \approx \psi(\mathbf{x}_N) + \frac{1}{2} R \sum_{k=0}^{N} \omega_{\theta_k}^2, \tag{271}$$



(a) Cost policy based on the $\mathcal{L}_{trace}$ cost. CE optimization for three different orbit sectors.

(b) Cost policy based on the $\mathcal{L}_{\text{tuo}}$ cost. CE optimization for three different orbit sectors.

(c) Cost policy based on the $\mathcal{L}_{mtuo}$ cost. CE optimization for three different orbit sectors.

Figure 149: Cross Entropy optimization for the proposed cost policies. The solid black line represents the null cost. In this case, $N_{CE} = 15$.

where in Eq. 250 we let $Q(\mathbf{x}) = 0$ and $\psi(\mathbf{x}_N) = \|e^2(N)\|$. The control law is parametrized as follows:

$$\begin{aligned} \omega_{\theta_k} &= \mathbf{u}(\omega_{\theta_{k-1}}, \eta(k-1;\lambda)) \\ &= \omega_{\theta_{k-1}} + \eta(k-1;\lambda)\delta t \end{aligned} \tag{272}$$

where $\eta(k-1;\lambda)$ is the rotational acceleration, which is parameterized as a piecewise trajectory composed by $m$ constant pieces. The choice of parameterizing the acceleration allows to have smooth (at least of class $C^1$) angular trajectories.

Each constant acceleration $\eta_m$ is being applied for a constant duration $\delta t_i$, where $t_{\text{sect}} = \sum_{i=1}^{m} \delta t_i$. The sum of all time intervals is fixed and equal to the time horizon corresponding to the duration of each sector $s$ (refer to the table in Section 7.10). The parameter vector $\lambda$ is defined as $\lambda^T = (t_1, \eta_1, \ldots, t_m, \eta_m) \in \mathbb{R}^{2m}$.

Each parameter vector $\lambda$ corresponds to a unique control vector $\mathbf{u}$, which generates a trajectory $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{t_N}]$.

In the numerical simulations, and without loss of generality, we maintain the controller timestep constant $\delta t_i = \delta t_m = t_{\text{sect}}/m$. The accelerations $\eta_i$ are initially obtained from a uniform distribution $\mathcal{U}([\eta_{min}, \eta_{max}])$, where the bounds are dictated by the specifics of the sensor.

### 7.6.8 Algorithm Set Up

**Problem Algorithm**

```
for $s = 1$ to total sectors per turn do
    Selected initial distribution parameters $\mathbf{v}_0$
    for $i = 1$ to total CE optimization steps do
        Draw $N_{\text{traj}}$ random acceleration vectors $\lambda \in \mathbb{R}^m$ from distribution
        with parameters $v_i$
        for $j = 1$ to $N_{\text{traj}}$ do
            Run a simulated EKF with the input $N^j_{\text{traj}}$
            Evaluate the cost function 250 and store the value
        end for
        Sort all the cost function values in ascending order
        Take the $\rho$-th quantile Run the cross entropy optimization in
        Eq. 265 and extract the new distribution parameters $\mathbf{v}_{i+1}$
    end for
    Apply the obtained near optimal control law $\lambda \in \mathbb{R}^m$ to sector $s$.
end for
```

To evaluate the proposed control policies, we consider the scenario of a satellite circumnavigating another satellite in orbit, while observing a set of feature points (landmarks) on the target satellite. The objective of observing the satellite is to accurately localize the landmarks.

The optimization step follows a first full orbit around the object in which the chaser satellite observes the landmarks in a *recognition* mode: in this first part, no control is applied to the sensor, which, for example, points towards the center of the orbit at all times.

After a first turn has been completed, and the state vector $\mathbf{x}$ has been augmented to dimension $\mathbb{R}^{4+2\times\mathcal{N}}$ through landmark observations[4], the CE

[4] We have that $\mathcal{N} \leq N$

routine is implemented.

Since the trajectory simulated in the EFK prediction routine is dependent on the intrinsic uncertainty of the sensor, a long time horizon will result in a build up of errors and uncertainties. The CE routine is then applied to a finite time horizon, equal to a fraction of the orbital period.

In this simulation, the orbit has been divided in $s$ sectors: each sector is divided in $m$ sampling boxes, where $m$ is the size of the control action vector $\lambda$. Depending on the implemented discretization, each box consists of $l$ iterations. That is, for each box $m_i$, constant control parameters $\lambda_i$ are applied for $l$ number of iterations of the box. Fig. 148 illustrates the discretization strategy.

The cross entropy control strategy for the simulation is listed in the Problem Algorithm routine: $N_{\text{traj}}$ random control laws are drawn by using the starting distribution parameters $\mathbf{v}_0$. An EKF simulation is then run for each of the $N_{\text{traj}}$ control laws, leading to different trajectories; these are ordered according to their respective cost and a quantile $q - th$ is selected. The best $q - th$ quantile provides the new parameters $\mathbf{v}_i$ from which the next $N_{\text{traj}}$ control laws are drawn. The process iterates for the $N_{CE}$ cross-entropy optimization steps. The output of the algorithm is the near-optimal control law $\lambda \in \mathbb{R}^m$ with $m$ being the number of boxes in which the $s_i$ sector is divided.

## 7.7 SIMULATION RESULTS

We present in this section the results for the three proposed cost policies. All the simulations, in order to maintain consistency, comprise the same number and location of landmarks, with the same orbit and sensor characteristics. The simulation parameters are listed in Table 12.

The position of the landmarks is obtained by randomly extracting $N_{\text{lnd}} \times 2$ values in $\mathcal{N}(0, \frac{R}{2})$.

Table 12: Simulation parameters

| | | | |
|---|---|---|---|
| Orbit | Orbit radius | R | 5 |
| | Angular velocity | $\omega$ | 0.2 |
| Set up | Sectors per turn | s | 6 |
| | Boxes per sector | m | 10 |
| | Iterations per box | $l$ | 20 |
| | Number of landmarks | $N_{\text{lnd}}$ | 10 |
| Sensor | Range | r | 4.7 |
| | Bearing | $\alpha$ | 5ą |
| | Max acceleration | $a_{\text{max}}$ | 2 |
| Noise | Model | $\sigma_\omega$ | 0.002 |
| | Measurement | $\sigma_v$ | 0.002 |

The simulation starts with a first orbit in *recognition mode*, during which the sensor is kept Nadir-pointing (that is, $\mathbf{x}(4) = \theta = 0$). Data regarding the landmarks is collected, along with the uncertainty of the EKF estimate. Graphically, the uncertainty can be represented with covariance ellipses.

In Fig. 147 (a-c), the *recognition mode* simulation is illustrated at different steps during the first turn. Initial uncertainty is dictated by the simulation and sensor noise and is influenced by the number of measurements taken.

After this first initial turn, the cross entropy control is applied. The algorithm's ability to drastically reduce the uncertainty and to improve the mutual localization has been demonstrated for all three different control strategies.

In Fig. 147 (d–f), the cross entropy minimization orbit is depicted at three different steps, in which the shrinking in the size of the covariance matrix ellipses can be clearly seen (in this particular case, the cost strategy based on $\mathcal{L}_{\text{tcm}}$ has been implemented). Note that the beam points always towards a group of landmarks, whereas in the recognition mode the sensor is kept pointed at the center of the orbit.

Analysis of the proposed optimization strategies is presented in Fig. 149.

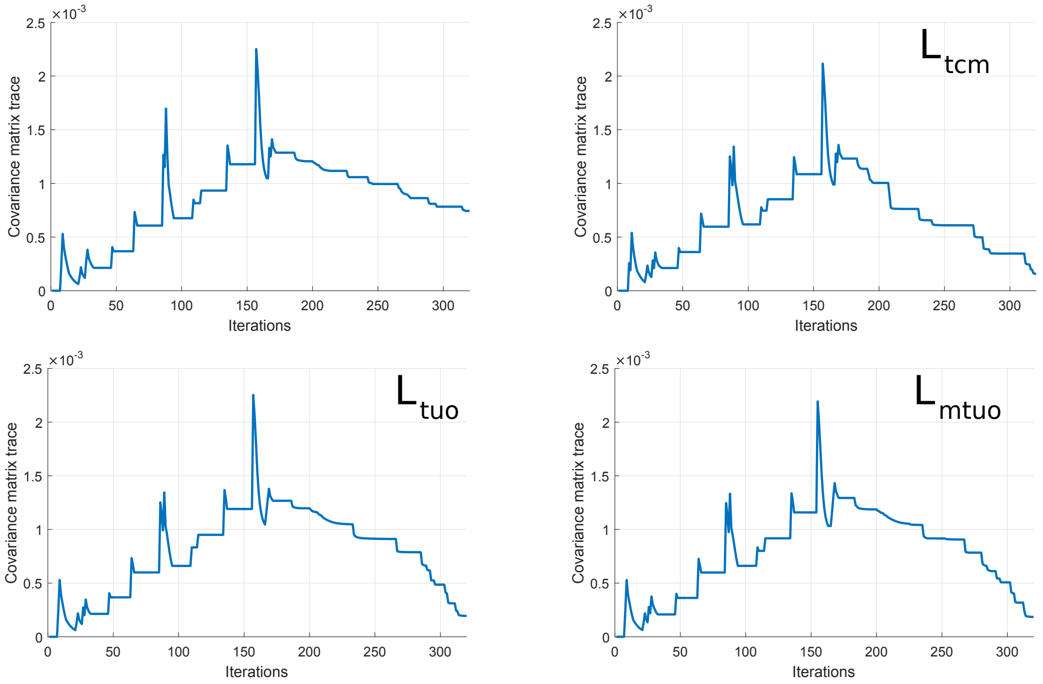


Figure 150: Trace of the covariance matrix in the uncontrolled and controlled cases.

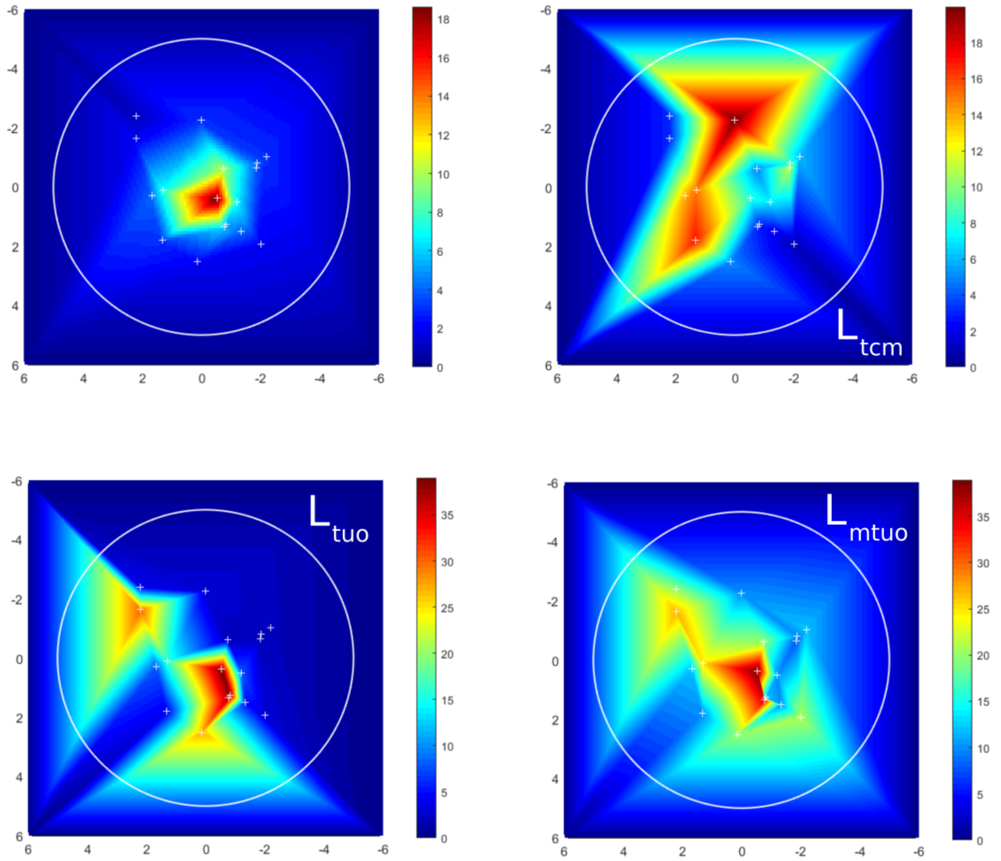


Figure 151: Landmark observations' potential map in the uncontrolled and controlled cases.

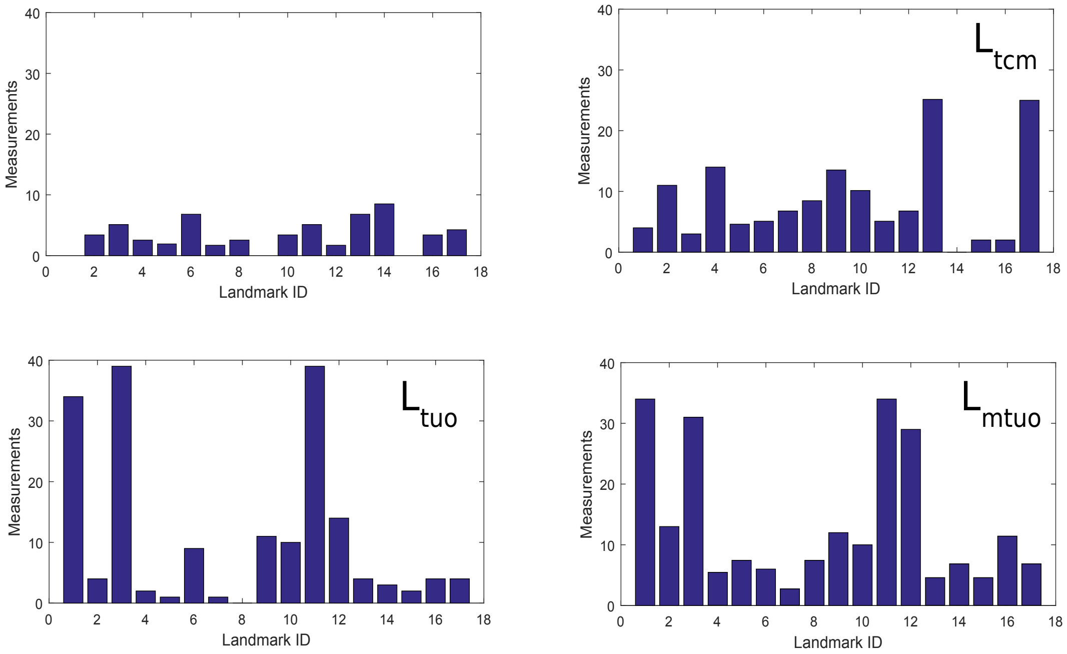


Figure 152: Cumulative landmark detection in he uncontrolled and controlled cases.

These plots represent the average and maximum cost associated with the parameter vector $\lambda_i$ for $i = 1 \ldots N_{CE}$, for different sectors $s$ along the circular orbit. Due to the design of the cost policies, the objective is to find the minimum for $\mathcal{L}_{\text{trace}}$ and the maxima for $\mathcal{L}_{\text{tuo}}, \mathcal{L}_{\text{mtuo}}$.

The cost strategy based on the trace of the covariance matrix enables the top $\lambda_i$ quantile to converge almost immediately to the optimal trajectory (blue line in Fig. 149 (a)). On the contrary, the costs based on the landmark observation inferences ($\mathcal{L}_{\text{tuo}}$ and $\mathcal{L}_{\text{mtuo}}$) show a slower convergence rate towards the optimal value, albeit the latter is always reached without significant oscillations (Fig. 149 (b-c)). In all the performed simulations, the proposed cost strategies outperform the non controlled case (solid horizontal line in Fig. 149).

A further analysis to gain an insight of the performance takes into account the overall uncertainty reduction made possible with the proposed cost strategies. A good estimate of this uncertainty is again the trace of the covariance matrix: in Fig. 150 the trace behavior is plotted for the non controlled and the controlled cases. The trace increase in the first leg of the curve is due to the landmark acquisition and population of the (initially empty) covariance matrix during the first orbit. Since $\omega_\theta = 0$ during the *recognition* phase, this first part is identical (with the obvious differences due to noise) for all cases.

The second part of the curve is influenced by the strategy under analysis. As expected, the decay of trace($\Sigma(t_N)$) is faster in the controlled case ($\mathcal{L}_{\text{tcm}}$, $\mathcal{L}_{\text{tuo}}$ and $\mathcal{L}_{\text{mtuo}}$) compared to the non-controlled scenario (Fig. 150 (a)). In particular, the cost $\mathcal{L}_{\text{tcm}}$ allows for the best performance in terms of uncertainty reduction, with $\mathcal{L}_{\text{tuo}}$ and $\mathcal{L}_{\text{mtuo}}$ performing very similarly (Fig. 150 (c-d)).

Finally, we studied how the different costs lead to differences in landmark detection. To do this, we represented the time spent under observation as the potential map in Fig. 149, in which the color intensity represents the number of times each landmark has been measured in the same orbital portion. The scatter plot has then been interpolated and a 3D surface was computed. As expected, in the non controlled case most observations happen in the proximity of the orbit's center, Fig. 149 (a). The cost based on the trace, Fig. 149 (b), presents fairly good performances in terms of duration and distribution of the observations. Cost strategies based on $\mathcal{L}_{\text{tuo}}$ and $\mathcal{L}_{\text{mtuo}}$ present a very similar shape in terms of observed landmarks: however, the cost $\mathcal{L}_{\text{mtuo}}$ allows for an even distribution of the observations. This is due to the additional term in Eq. 256, which takes also into account the number of different features: in the potential plot, this is confirmed by the more uniform gradient among the landmarks.

Overall, strategies based on $\mathcal{L}_{\text{tuo}}$ and $\mathcal{L}_{\text{mtuo}}$ allow for a higher number of observed features: we present another performance indicator for the algorithm. In Fig. 152, the cumulative landmark observations are shown with the aid of bar charts: each bar represents a landmark and the height of each bar represents the number of detection during a turn.

It is interesting to note how the non-controlled case, Fig. 152 (a), performs poorly, both in terms of number of detected landmarks and distribution (some landmarks, for example, are never detected). The first control policy, based on $\mathcal{L}_{\text{tcm}}$, allows for a significant performance increase, both in terms of number and frequency of the observations (Fig. 152 (b)). Cases $\mathcal{L}_{\text{tuo}}$ and $\mathcal{L}_{\text{mtuo}}$ show again a similar structure in the observation frequency, but with the substantial difference of a much more even distribution in the case of $\mathcal{L}_{\text{mtuo}}$.

In conclusion, based on the potential and cumulative analyses, the modified $\mathcal{L}_{\text{mtuo}}$ allows for more frequent and more even observations if compared to the performance of $\mathcal{L}_{\text{tcm}}$.

## 7.8 CROSS ENTROPY 3D EXTENSION

### 7.8.1 Relative Navigation in Orbit

In this section, we extend the work presented in the previous pages to the general case of a target and a chaser satellite in a 3D orbital scenario. The relative orbit will be described with Clohessy-Wiltshire's equations (CW).

The satellite has again an onboard sensor, which is free to rotate around the axis normal to the orbital plane. In order to describe the relative motion between the chaser and the target, we start by analyzing Hill's equations.

### 7.8.2 Clohessy-Wiltshire Reference Frame

The Clohessy-Wiltshire framework allows for the description of orbital relative motion, in which the target is in a circular orbit, and the chaser is in an elliptical (or circular) orbit. This model is a first-order approximation of the actual chaser's motion in a target-centered coordinate system.

Hill's differential equations in Cartesian coordinates and in the non homogeneous form can be written as follows [61, 62]:

$$\begin{cases} \ddot{x} - 3n^2x - 2n\dot{y} & = f_x \\ \ddot{y} + 2n\dot{x} & = f_y \\ \ddot{z} + n^2z & = f_z \end{cases}$$

The CW equations can be obtained solving Hill's differential unforced equations with the standard Laplace transform, which yields:

$$\begin{cases} x(t) = x_0[4 - 3\cos(nt)] + \dfrac{\sin(nt)}{n}\dot{x}_0 + \dfrac{2\dot{y}_0}{n}[1 - \cos(nt)] \\ y(t) = y_0 - \dfrac{2\dot{x}_0}{n} - 3(2nx_0 + \dot{y}_0)t + 2(3x_0 + \dfrac{2\dot{y}_0}{n})\sin(nt) \\ \qquad + \dfrac{2\dot{x}_0}{n}\cos(nt) \\ z(t) = z_0\cos(nt) + \dfrac{\dot{z}_0}{n}\sin(nt) \end{cases}$$

One interesting property of these equations is that, although the equations describing the in-plane motion are coupled, the out-of-plane motion is decoupled.

Even though the chaser does not actually orbit around the target satellite, the instantaneous motion is elliptical [63]. The term $(2nx_0 + \dot{y}_0)t$ in the $y$-equation represents the secular drift between the chaser and the target due to differences in the orbital periods. If this term is set to zero by choosing the appropriate initial conditions, $\dot{y}_0 + 2nx_0 = 0$, then the linearized relative orbit will have a bounded motion.

Assuming this constraint is satisfied, then the HCW equations can be rewritten as follows [54]:

$$\begin{cases} x(t) = \alpha\sin(nt + \phi_1) \\ y(t) = 2\alpha\cos(nt + \phi_1) + \Delta y \\ z(t) = \beta\sin(nt + \phi_2) \end{cases}$$

where the parameters $\alpha$, $\beta$, $\Delta y$, $\phi_1$ and $\phi_2$ are determined through the relative orbit initial conditions:

$$\alpha = \sqrt{x_0^2 + \frac{\dot{x}_0^2}{n^2}}, \qquad \beta = \sqrt{z_0^2 + \frac{\dot{z}_0^2}{n^2}} \tag{273}$$

$$\Delta y = y_0 - 2\frac{\dot{x}_0}{n} \tag{274}$$

$$\phi_1 = \tan^{-1}\left(\frac{nx_0}{\dot{x}_0}\right), \qquad \phi_2 = \tan^{-1}\left(\frac{nz_0}{\dot{z}_0}\right) \tag{275}$$

In order to simplify the analysis, we may impose a target-centered orbit by driving $\Delta y$ to zero, that is let $y_0 = 2(\dot{x}_0/n)$. The case $\Delta y \neq 0$ is trivial and does not add any significant novelty. One finally obtains:

$$\begin{cases} x(t) = \alpha \sin(nt + \phi_1) \\ y(t) = 2\alpha \cos(nt + \phi_1) \\ z(t) = \beta \sin(nt + \phi_2) \end{cases}$$

$$\begin{cases} \dot{x}(t) = \alpha \cos(nt + \phi_1)n \\ \dot{y}(t) = -2\alpha \sin(nt + \phi_1)n \\ \dot{z}(t) = \beta \cos(nt + \phi_2)n \end{cases}$$

The unit vector normal to the osculating plane can be derived as the unity momentum:

$$\hat{\mathbf{n}} = \frac{\mathbf{x} \times \dot{\mathbf{x}}}{\|\mathbf{x} \times \dot{\mathbf{x}}\|} \tag{276}$$

A new frame of reference attached to the Hill's orbit is defined, {H}, with the $x$-axis and $y$-axis lying on the osculating plane and directed towards the apsis and periapsis respectively. The direction is chosen to form a right-handed frame $\{\hat{\mathbf{i}}_H, \hat{\mathbf{j}}_H, \hat{\mathbf{k}}_H\}$ with the $z$-axis, represented by $\hat{\mathbf{n}}$. Frames {G} and {H} have the same null origin by definition since $y_0 = 2(\dot{x}_0/n)$.

### 7.8.3 State Model

In this case, the state model changes with respect to the one used in the 2D case, and can be expressed as:

$$\begin{bmatrix} \mathrm{d}x(t) \\ \mathrm{d}y(t) \\ \mathrm{d}z(t) \\ \mathrm{d}\phi(t) \\ \mathrm{d}\theta(t) \\ \mathrm{d}\mathbf{p}_1(t) \\ \vdots \\ \mathrm{d}\mathbf{p}_N(t) \end{bmatrix} = \begin{bmatrix} \alpha\cos(\phi + \phi_1)n \\ -2\alpha\sin(\phi + \phi_1)n \\ \beta\cos(\phi + \phi_2)n \\ \omega_\phi(t)\mathrm{d}t \\ \omega_\theta(t)\mathrm{d}t \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{I}_5 \\ \mathbf{0}_{3N\times 5} \end{bmatrix} \mathrm{d}\mathbf{w}(t) \tag{277}$$

where $x, y, z$ indicate the position of the chaser satellite in the $\{G\}$ frame, and the angles $\phi$ and $\theta$ are the rotation of the chaser and the sensor expressed in frames $\{H\}$ and $\{R\}$ respectively; note that $\phi(t) = nt$ is the cumulative angle: that is, we suppose that in the baseline case (with no control applied), the chaser rotates with an angular velocity vector perpendicular to the relative orbital plane and with magnitude equal to the mean motion ($\omega_{\phi_k} = n$).

The landmark positions $\mathbf{p}_1, \ldots \mathbf{p}_N$ are expressed in the global frame, and yield an augmented state $\mathbf{x} \in \mathbb{R}^{3N+5}$.

In the model, $\mathrm{d}\mathbf{w} \in \mathbb{R}^5$ represents a Wiener process with covariance matrix $\mathbf{\Sigma}_w = \mathrm{diag}(\sigma_1^2, \sigma_2^2, \sigma_3^2, \sigma_4^2, \sigma_5^2)$.

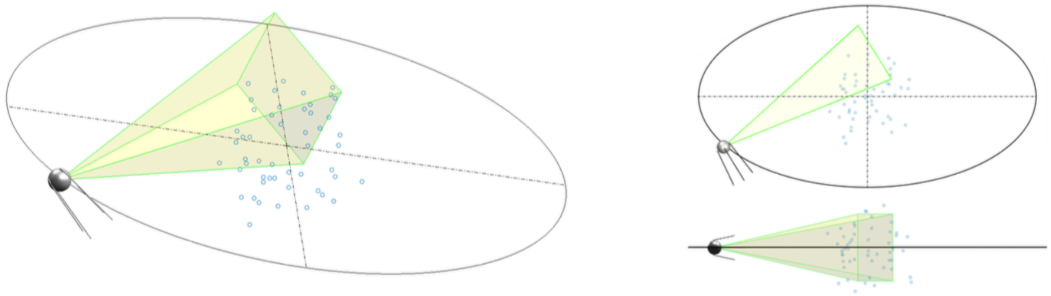


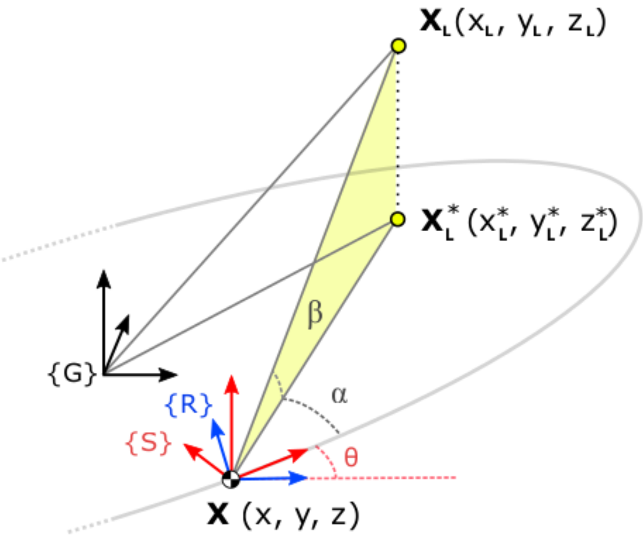
Figure 153: Problem set up and simulated sensing scenario.

In discrete form, the model becomes:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ z_{k+1} \\ \phi_{k+1} \\ \theta_{k+1} \\ \mathbf{p}_{1k+1} \\ \vdots \\ \mathbf{p}_{Nk+1} \end{bmatrix} = \begin{bmatrix} x_k + \alpha\cos(\phi_k + \phi_1)n\delta t \\ y_k - 2\alpha\sin(\phi_k + \phi_1)n\delta t \\ z_k + \beta\cos(\phi_k + \phi_2)n\delta t \\ \phi_k + \omega_{\phi_k}\delta t \\ \theta_k + \omega_{\theta_k}\delta t \\ \mathbf{p}_{1k} \\ \vdots \\ \mathbf{p}_{Nk} \end{bmatrix} + \begin{bmatrix} \mathbf{I}_5 \\ \mathbf{0}_{3N\times 5} \end{bmatrix} \mathbf{w}(t_k) \tag{278}$$

## 7.8.4 Measurement Model

Detection of the landmarks occurs only if they are within the field of view and range of the sensor, depicted as the yellow truncated pyramid in Fig. 153. When a feature is detected, the sensor outputs this time the augmented vector $\mathbf{z} = (r, \alpha, \beta)$, where $r$ is the range and the tuple $(\alpha, \beta)$ are the azimuth and elevation angles (see Fig. 154).



$\mathbf{X}_L(x_L, y_L, z_L)$
$\mathbf{X}^*_L(x^*_L, y^*_L, z^*_L)$
β
{G}
{R}
α
{S}
θ
$\mathbf{X}$ (x, y, z)

Figure 154: Measurement model definition.

The measurement model, expressed in continuous form, is the following:

$$\mathbf{z}(t) = {}^{S}_{R}\mathbf{R}(\theta(t))\,{}^{R}_{H}\mathbf{R}(\phi(t))\,{}^{H}_{G}\mathbf{R}(\mathbf{p}_i(t) - \mathbf{p}_R(t)) + \mathbf{v}(t) \tag{279}$$

where $\mathbf{p}_i = (p_{x_i}, p_{y_i}, p_{z_i})$ and $\mathbf{p}_R = (x, y, z)$ are the position of the landmarks and the observer satellite, respectively, expressed in the base frame. The term $\mathbf{v}(t)$ corresponds to the observation noise of the sensor which is considered zero-mean Gaussian with covariance matrix $\mathbf{\Sigma}_v = \mathrm{diag}(\sigma^2_{\mathrm{I}}, \sigma^2_{\mathrm{II}}, \sigma^2_{\mathrm{III}})$. The matrices ${}^{H}_{G}\mathbf{R}$, ${}^{S}_{R}\mathbf{R}(\theta(t))$ and ${}^{R}_{H}\mathbf{R}(\phi(t))$ express rotational transformations from the base frame {G} to the orbit {H}, from {H} to the observer frame {R} and from {R} to the sensor frame {S}, respectively. In compact form, the observation model is then written as:

$$\mathbf{z}(t) = \mathbf{h}(\mathbf{x}(t)) + \mathbf{v}(t) \tag{280}$$

Or, in discrete time form:

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \tag{281}$$

Referring to Fig. 154, the measurement model mapping function can be described as:

$$\mathbf{h}(\mathbf{x}) = \begin{bmatrix} \sqrt{(x_L - x)^2 + (y_L - y)^2 + (z_L - z)^2} \\ \tan^{-1}\left(\dfrac{y_L^* - y_G}{x_L^* - x_G}\right) - \phi - \theta \\ \tan^{-1}\left(\dfrac{\text{sgn}((\mathbf{x}_L - \mathbf{x}_L^*) \cdot \mathbf{n})\ \|\mathbf{x}_L - \mathbf{x}_L^*\|}{\|\mathbf{x}_L^* - \mathbf{x}\|}\right) \end{bmatrix} \tag{282}$$

The projected vector $\mathbf{x}_L^*$ can be further expressed as a function of the state by knowing the transformation map between frames {G} and {H}, which remains constant throughout the simulation:

$$\mathbf{x}_L^* = \mathbf{x}_L - (\mathbf{x}_L \cdot \mathbf{n})\mathbf{n} \tag{283}$$

The measurement model can then be written as:

$$\mathbf{h}(\mathbf{x}) = \begin{bmatrix} \sqrt{(x_L - x)^2 + (y_L - y)^2 + (z_L - z)^2} \\ \\ \tan^{-1}\left(\dfrac{(\mathbf{x}_L - (\mathbf{x}_L \cdot \mathbf{n})\mathbf{n} - \mathbf{x}) \cdot \mathbf{j}_H}{(\mathbf{x}_L - (\mathbf{x}_L \cdot \mathbf{n})\mathbf{n} - \mathbf{x}) \cdot \mathbf{i}_H}\right) - \phi - \theta \\ \\ \tan^{-1}\left(\dfrac{\text{sgn}(((\mathbf{x}_L \cdot \mathbf{n})\mathbf{n}) \cdot \mathbf{n})\ \|(\mathbf{x}_L \cdot \mathbf{n})\mathbf{n}\|}{\|\mathbf{x}_L - (\mathbf{x}_L \cdot \mathbf{n})\mathbf{n} - \mathbf{x}\|}\right) \end{bmatrix} \tag{284}$$

## 7.9 DESIGN OF THE ALGORITHM

The proposed algorithm is still composed by two phases, described thoroughly in the previous sections:

- the *recognition* phase,
- the *incremental* estimation phase

Although the EKF structure remains the same, the details change due to the added dimensionality of the problem; the following sections explains these changes.

### 7.9.1 Recognition Phase

With $\tilde{N}$ being the number of landmarks recognized up to the time instant $k$, we write:

$$\mathbf{x}_k^{(\tilde{N})} = \begin{bmatrix} x_k & y_k & z_k & \phi_k & \theta_k & \mathbf{p}_k^{(1)} & \mathbf{p}_k^{(2)} & \dots & \mathbf{p}_k^{(\tilde{N})} \end{bmatrix}^T.$$

*Prediction step*

The augmented state is:

$$\begin{bmatrix} \hat{x}_{k+1|k} \\ \hat{y}_{k+1|k} \\ \hat{z}_{k+1|k} \\ \hat{\phi}_{k+1|k} \\ \hat{\theta}_{k+1|k} \\ \hat{p}^{(1)}_{k+1|k} \\ \hat{p}^{(2)}_{k+1|k} \\ \vdots \\ \hat{p}^{(\tilde{N})}_{k+1|k} \end{bmatrix} = \begin{bmatrix} \hat{x}_{k|k} \\ \hat{y}_{k|k} \\ \hat{z}_{k|k} \\ \hat{\phi}_{k|k} \\ \hat{\theta}_{k|k} \\ \hat{p}^{(1)}_{k|k} \\ \hat{p}^{(2)}_{k|k} \\ \vdots \\ \hat{p}^{(\tilde{N})}_{k|k} \end{bmatrix} + \begin{bmatrix} \alpha\cos(\phi_{k|k} + \phi_1) n\delta t \\ -2\alpha\sin(\phi_{k|k} + \phi_1) n\delta t \\ \beta\cos(\phi_{k|k} + \phi_2) n\delta t \\ \omega_{\phi_k}\delta t \\ \omega_{\theta_k}\delta t \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

or, in a more compact form:

$$\hat{\mathbf{x}}^{(\tilde{N})}_{k+1|k} = \mathbf{f}(\hat{\mathbf{x}}^{(\tilde{N})}_{k|k}, \omega_{\phi_k}, \omega_{\theta_k}).$$

The update of the covariance matrix is given by

$$P^{(\tilde{N})}_{k+1|k} = F_k P^{(\tilde{N})}_{k|k} F_k^T + Q_k, \tag{285}$$

where

$$F_k = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}. \tag{286}$$

The matrix $F_k$ has the following structure

$$F = \begin{bmatrix} F_k^{\text{mot}} & 0 \\ 0 & \mathbb{I}_{\tilde{N}} \end{bmatrix}, \tag{287}$$

where $F_k^{\mathrm{mot}}$ is given by the following:

$$\mathbf{F}_k^{\mathrm{mot}} = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial z} & \frac{\partial f_1}{\partial \phi} & \frac{\partial f_1}{\partial \theta} \\ \dots & \dots & \dots & \dots & \dots \\ \frac{\partial f_5}{\partial x} & \frac{\partial f_5}{\partial y} & \frac{\partial f_5}{\partial z} & \frac{\partial f_5}{\partial \phi} & \frac{\partial f_5}{\partial \theta} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & -\alpha \sin(\phi_{k|k} + \phi_1) n\delta t & 0 \\ 0 & 1 & 0 & -2\alpha \cos(\phi_{k|k} + \phi_1) n\delta t & 0 \\ 0 & 0 & 1 & -\beta \sin(\phi_{k|k} + \phi_2) n\delta t & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} . \tag{288}$$

where $0_N$ is a null matrix of dimension $N$.

*Update step*

We assume to have the information provided by the range and bearing sensor $\mathbf{z} = [r, \alpha, \beta]$. Furthermore, we assume that we collect multiple measurements at the same time instant $k+1$, e.g. $\bar{\mathbf{z}}_{k+1}$. This vector can be divided in two components, the first component $\mathbf{z}_{k+1}^{(1)}$ which is given by all the measurements associated to already seen landmarks and the second component $\mathbf{z}_{k+1}^{(2)}$ which represents measurements associated to new landmarks. The measurement model can be written as

$$\bar{\mathbf{z}}_{k+1} = \begin{bmatrix} \mathbf{z}_{k+1}^{(1)} \\ \mathbf{z}_{k+1}^{(2)} \end{bmatrix} = \begin{bmatrix} \mathbf{h}^{(1)}(\hat{\mathbf{x}}_{\mathbf{k+1}}) + v_{k+1}^{(1)} \\ \mathbf{h}^{(2)}(\hat{\mathbf{x}}_{\mathbf{k+1}}) + v_{k+1}^{(2)} \end{bmatrix} .$$

We proceed with the computation of the Jacobian of the observation model with respect to the robot pose and the observed landmark coordinates. At iteration $k+1$ we get

$$\mathbf{H}_{k+1} = \left. \frac{\partial \mathbf{h}_{k+1}^{(1)}}{\partial \mathbf{x}^{(\tilde{N})}} \right|_{\hat{x}_{k+1|k}} \tag{289}$$

By having the output matrix $\mathbf{H}_{k+1}$ we can update the state related to all chaser attitude and all the already seen landmarks

$$\begin{aligned}
K_{k+1} &= P^{(\tilde{N})}_{k+1|k} H^T_{k+1} \left( H_{k+1} P^{(\tilde{N})}_{k+1|k} H^T_{k+1} + R_{k+1} \right)^{-1} \\
x^{(\tilde{N})}_{k+1|k+1} &= x^{(\tilde{N})}_{k+1|k} + K_{k+1} \mathbf{z}^{(1)}_{k+1} \\
P^{(\tilde{N})}_{k+1|k+1} &= (\mathbb{I} - K_{k+1} H_{k+1}) P^{(\tilde{N})}_{k+1|k}
\end{aligned}$$

Without loss of generality, suppose that $\mathbf{z}^{(2)}_{k+1}$ refers to just one new landmark $p^{(\tilde{N}+1)}$, then we have that

$$\hat{p}^{(\tilde{N}+1)}_{k+1|k+1} = \begin{bmatrix} \hat{x}_{k+1|k} \\ \hat{y}_{k+1|k} \\ \hat{z}_{k+1|k} \end{bmatrix} + {}^R_H\mathbf{R} \begin{bmatrix} r\cos(\beta)\cos(\alpha + \hat{\phi}_{k+1|k} + \hat{\theta}_{k+1|k}) \\ r\cos(\beta)\sin(\alpha + \hat{\phi}_{k+1|k} + \hat{\theta}_{k+1|k}) \\ r\sin(\beta) \end{bmatrix} .$$

Then we can extend the state

$$x^{(\tilde{N}+1)}_{k+1|k+1} = \begin{bmatrix} x^{(\tilde{N})}_{k+1|k+1} \\ \hat{p}^{(\tilde{N}+1)}_{k+1|k+1} \end{bmatrix} ,$$

and the covariance matrix

$$P^{(\tilde{N}+1)}_{k+1|k+1} = \begin{bmatrix} P^{(\tilde{N})}_{k+1|k+1} & P^{(\tilde{N},\tilde{N}+1)} \\ P^{(\tilde{N}+1,\tilde{N})} & P^{(\tilde{N}+1)} \end{bmatrix} ,$$

where

$$P^{(\tilde{N}+1,\tilde{N})} = \left( P^{(\tilde{N},\tilde{N}+1)} \right)^T = \left. \begin{bmatrix} \dfrac{\partial \hat{p}^{(\tilde{N}+1)}_{k+1|k+1}}{\partial x_k} \\ \dfrac{\partial \hat{p}^{(\tilde{N}+1)}_{k+1|k+1}}{\partial y_k} \\ \dfrac{\partial \hat{p}^{(\tilde{N}+1)}_{k+1|k+1}}{\partial \phi_k} \\ \dfrac{\partial \hat{p}^{(\tilde{N}+1)}_{k+1|k+1}}{\partial \theta_k} \end{bmatrix} \right|_{(\hat{x}_{k+1|k}, \bar{\mathbf{z}}_{k+1})} .$$

and

$$P^{(\tilde{N}+1)} = \left. \frac{\partial \hat{p}^{(\tilde{N}+1)}_{k+1|k+1}}{\partial \mathbf{z}_{k+1}} \right|_{(\hat{x}_{k+1|k}, \bar{\mathbf{z}}_{k+1})} .$$

### 7.9.2 Incremental Estimation Phase

The incremental estimation phase is derived directly from the 2D case:

1. We draw $N_{\text{traj}}$ random possible acceleration trajectories for the sensor, $\lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_{N_{\text{traj}}}\}$, from a Gaussian distribution with parameter $v_i$.

2. For all $\lambda$ we simulate the behavior of the camera running an Extended Kalman filter.

3. Once the state has been estimated at any time instant we can evaluate the cost function and perform the CE algorithm. We have to select the $\rho - th$ best performing percentile, i.e. the trajectories with an associated lower cost.

4. From these reduced subset of samples the new parameters for the distribution are inferred. The aforementioned procedure is repeated up to the convergence of the Cross Entropy method and then the optimal solution is applied.

### 7.9.3 Cross Entropy optimization for orbital self-localization

Even in the 3D case, the controller acts on the angular velocity of the sensor, $\omega_\theta$. Recalling Eq. 250, we can rewrite the discrete cost as:

$$\hat{\mathcal{L}}(\mathbf{x}, \mathbf{u}) \approx \psi(\mathbf{x}_{t_N}) + \sum_{k=0}^{N} \left( \frac{1}{2} \mathbf{u}(\mathbf{t_k})^T R \mathbf{u}(\mathbf{t_k}) \right), \tag{290}$$

where in Eq. 250 we let $Q(\mathbf{x}) = 0$ and $\psi(\mathbf{x}_{t_N}) = \|e^2(t_N)\|$. The control law is parametrized as follows:

$$\omega_\theta(t_k) = \mathbf{u}(\omega_\theta(t_{k-1}), \eta(t_{k-1}; \lambda)) \tag{291}$$

$$\omega_\theta(t_k) = \omega_\theta(t_{k-1}) + \eta(t_{k-1}; \lambda)\delta t, \tag{292}$$

In these simulations, and without loss of generality, we maintain the controller time-step constant $\delta t_i = \delta t_m = t_{\text{sect}}/m$. The accelerations $\eta_i$ are initially obtained from a uniform distribution $\mathcal{U}([\eta_{\min}, \eta_{\max}])$, where the bounds are dictated by the specifics of the sensor.

### 7.9.4 Algorithm Set Up

Control in this scenario means that the active rotation of the spacecraft about one of its axes is such that the sensor points to the landmarks detected in a previous time step. A certain cost function (e.g. the estimation accuracy of the detected features or the cumulative number of features seen) drives which feature(s) to be observed next, and hence also drives the control action. The control and estimation steps are therefore coupled. On the contrary, existing work in proximity operations solve the problem of control and estimation independently [53, 52].

## 7.10 SIMULATION RESULTS

This updated proposed algorithm has been used to simulate the acquisition and tracking of a set of landmarks on a virtual object located in the centroid of the closed Hill orbit.

In both cases, in order not to lose generality, landmarks are randomly placed according to a uniform distribution, so that the presence of particular geometrical properties/symmetries will not affect performance. The landmarks are thus generated according to the following distribution:

$$\mathbf{p} = \mathcal{U}(0, [\alpha, \beta, \gamma])$$

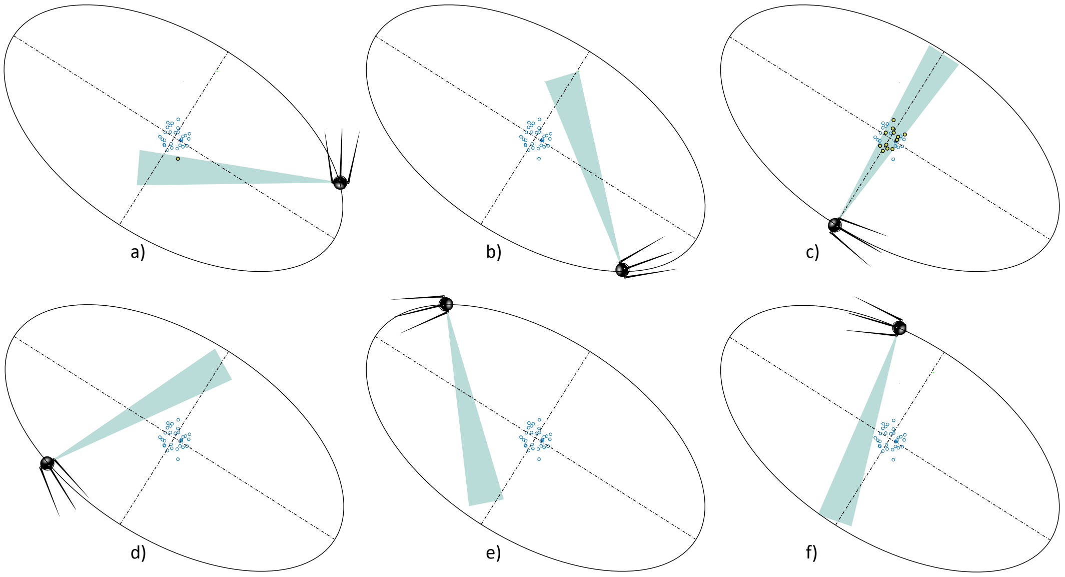


Figure 155: Reconnaissance orbit: the sensor is fixed with respect to the satellite ($\omega_\theta = 0$).

The distribution coefficient can be chosen to mimic the primitive shape a particular object. In this case, we chose $\alpha = \beta = \gamma = 0.2\,d_{\text{min}}$, with $d_{\text{min}}$ being the semi-minor axis of the relative CW orbit. In Table 12 we report the characteristics of the simulated scenario. We treat the case of a single reconnaissance orbit followed by a single optimization orbit.

The first orbit allows for the recognition of all the landmarks that fall in the field of view of the sensor: they are stored in the state and they are assigned a progressive number. The first orbit, during which no control is applied ($\omega_\theta = 0$), is represented in Fig. 155. The sensor rotates together with the spacecraft, whose angular velocity is equal to the mean motion of the relative orbit. That is, the satellite completes a full revolution around its axis for each orbit. Due to the elliptical shape of the orbit, the uncontrolled sensor spends a substantial amount of the orbit without acquiring any landmarks, even though its range would allow for potential observations (Fig. 155). In the figures, the observed landmarks are represented as green circles: note that, since the landmarks are positioned in 3D space,

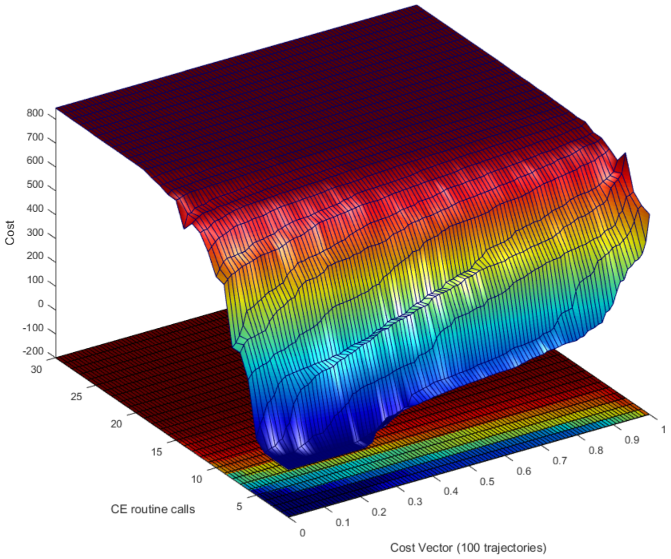


Figure 156: Visualization framework for the optimized results: cost vector against CE iterations

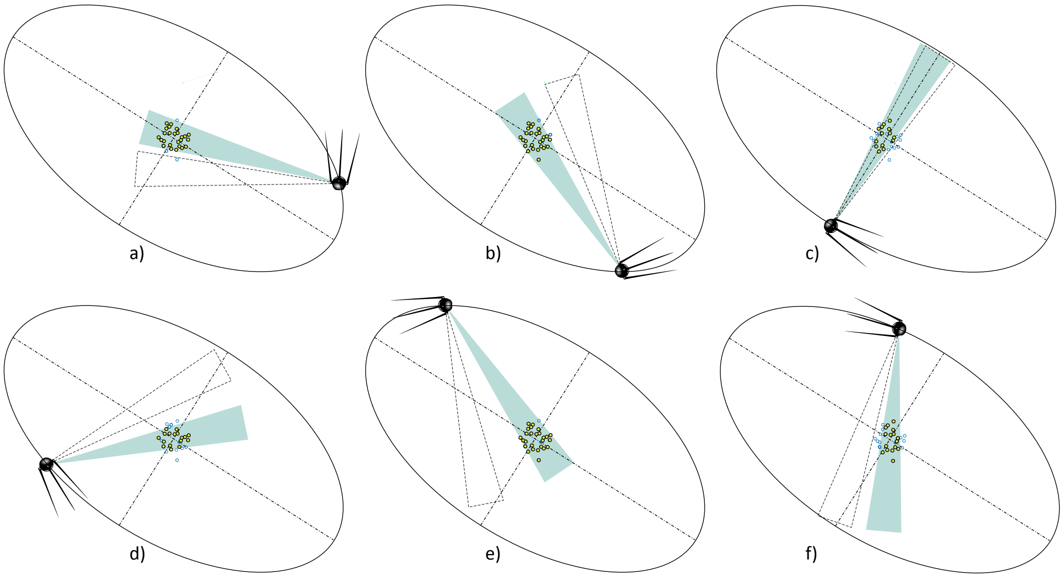


Figure 157: Controlled orbit through CE optimization: the uncontrolled sensor is represented as the ghost dashed shape.

some of them are not detected even if they appear in the 2D projection of the sensor's field of view in Fig. 155. After the first orbit is completed, the Cross-Entropy routine is applied, starting from the first sector. In reference [64], an optimization strategy based on the trace of the covariance matrix was presented and successfully simulated; in this section, we present the results obtained by applying the cost function based on the time under observation.

Although the CE routine can be computationally expensive if a fine trajectory quantization is seeked, the nature of this method allows for parallelization. The optimization algorithm, in fact, can be run as soon as the satellite has completed the acquisition of first sector's landmarks (beginning of the first orbit), in parallel with the acquisition of the upcoming sectors.

Since the time required for the CE algorithm to minimize the objective cost can be tailored by tuning the discretization step, the number of trial trajectories, CE iteration, etc., the optimized orbit can run in real time, keeping the sensor under control at all times. That is, by the time the satellite enters the second orbit, the optimized parameter vector pertaining the first sector, $\lambda_1$, is readily available.

The behavior of the satellite in the CE controlled orbit is shown in Fig. 157. The controlled sensor's FOV is the green triangular projection, whereas the dashed triangle represents the sensor's behavior when no controlled is applied. It is clearly noted that the controlled sensor is kept pointing at the landmarks at all times, maximizing the time under observation.

Apart from being limited in the acceleration profile, the sensor has to be also limited in terms of angular displacement. A switching control has been designed to avoid unrealistic trajectories: the parameters vectors $\lambda_i$ drawn from the normal distribution are screened and discarded if the control vector generates a trajectory such that $|\max(\theta_i)| \leq \theta_{\max}$.

A policy to limit the actuation cost has been used in order to prevent

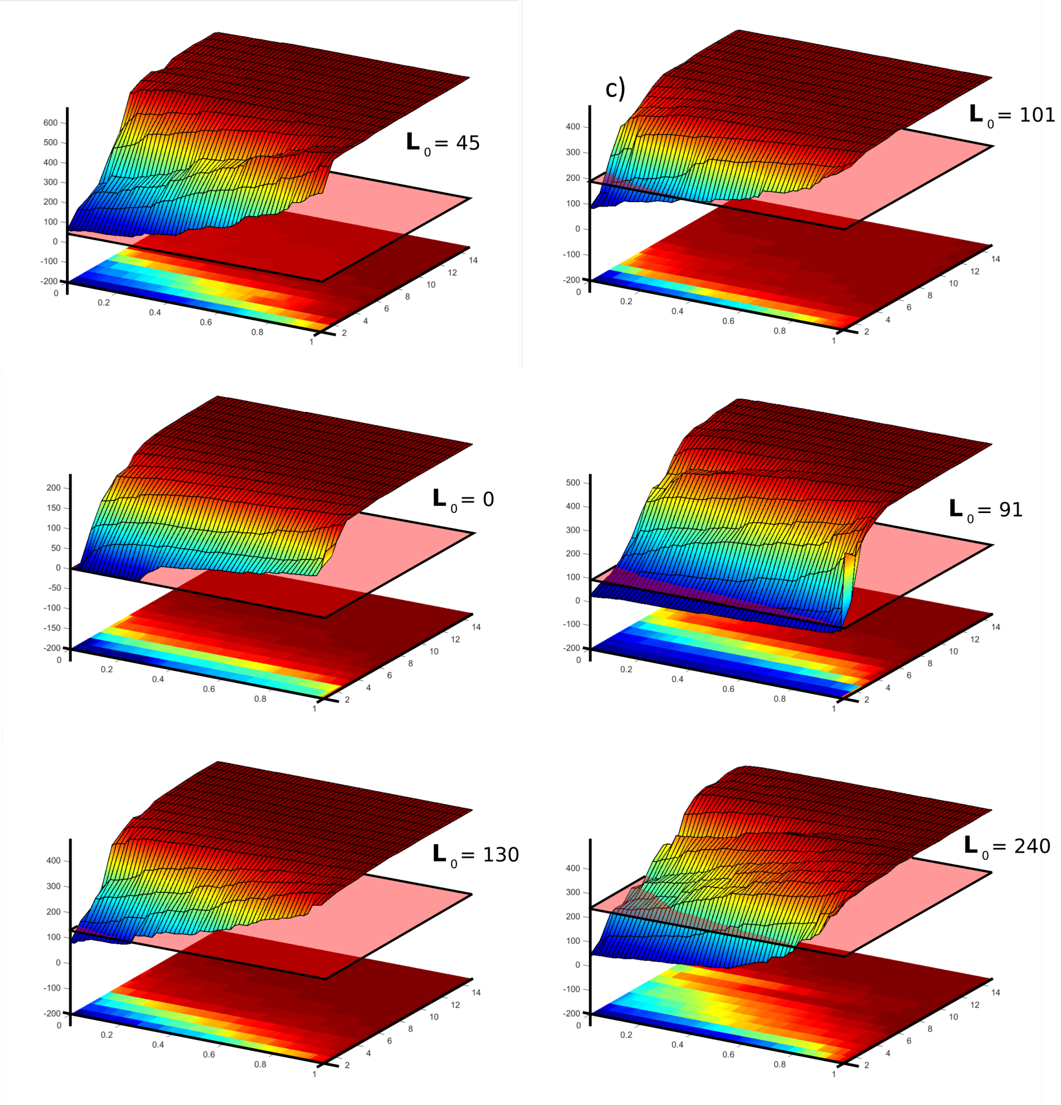


Figure 158: Controlled orbit through CE optimization: ordered cost vectors $\mathcal{L}_i$ plotted against the CE progressive optimization steps.

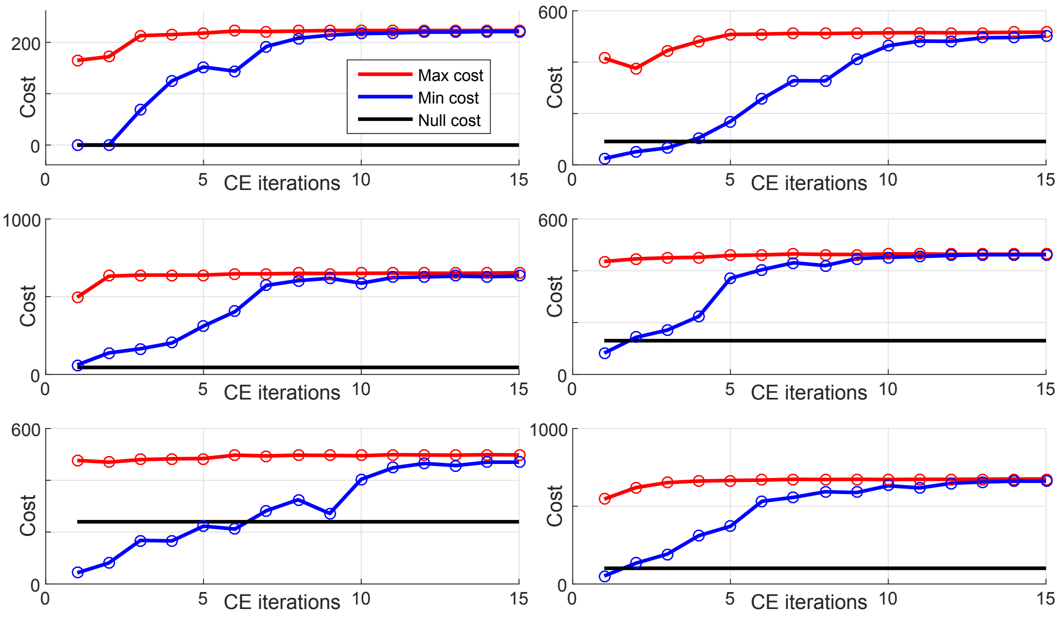


Figure 159: Cost performances for the CE progressive optimization.

unnecessary control. At the beginning of each CE optimization, the null cost $\mathcal{L}_{0,s}$ is computed for sector $s$. This is the cost in the case no control is being applied, i.e., the sensor kinematics is governed merely by the angular velocity at the previous time-step $\omega_{s-1,m}$.

The null cost is used as a reference to compare with the performance of the optimization routine. The plots in Fig. 158 represent the cost vector optimization process.

For each CE iteration (in this case $N_{CE} = 15$), the cost vector $\mathcal{L}_{N_{CE}}$ is computed and ordered: the stacking of the subsequent cost vectors creates the three-dimensional surfaces in the figures. The horizontal red plane depicts the null cost performance: when the surface is above the plane, the CE performances are superior than the non controlled case.

As can be seen from these results, even the lowest $\mathcal{L}$-vector percentiles are risen above the null-cost plane in the first two-to-three CE iterations. In all cases, the top percentiles are always above the null cost starting from the initial CE iteration [64].

In general, the CE optimization allows convergence to the optimal solution by progressively reducing the difference between the lower and upper percentiles of the cost vectors $\mathcal{L}$.

In Fig. 159, the maximum and minimum cost for each CE optimization step is represented: the strategy is capable of finding the maximum cost very early in the process (red lines). Then, the method takes 7-to-10 iterations to even out (in most cases monotonically) the range between the minimum and maximum cost vectors (blu lines). The solid black lines represent the null cost case, which is substantially outperformed in each of the orbital sectors.

## 7.11 RESULTS DISCUSSION

This chapter presented a novel approach for solving the active self localization problem during relative navigation in orbit using Cross Entropy (CE) minimization, expanding the previous work in a 2D framework [64]. Using the Clohessy-Wiltshire model, a real case chaser-target orbital scenario was presented.

By jointly considering the planning, control and estimation problems it was possible to balance the control actuation costs and the obtainable localization uncertainty: this has been obtained by incorporating an uncertainty measure in the cost functions, which is then utilized to select near-optimal trajectories in terms of estimation uncertainty. Results for the cost function based on the time under observation confirmed the validity of the method.

It is well known [65] that the main drawback of Cross Entropy implementation in control design is due to the substantial computational efforts required during optimization: to overcome this issue, by discretizing the orbit in a finite time horizon sectors, it was possible to use parallelization and to hence design a real-time controller. In our approach, the optimization

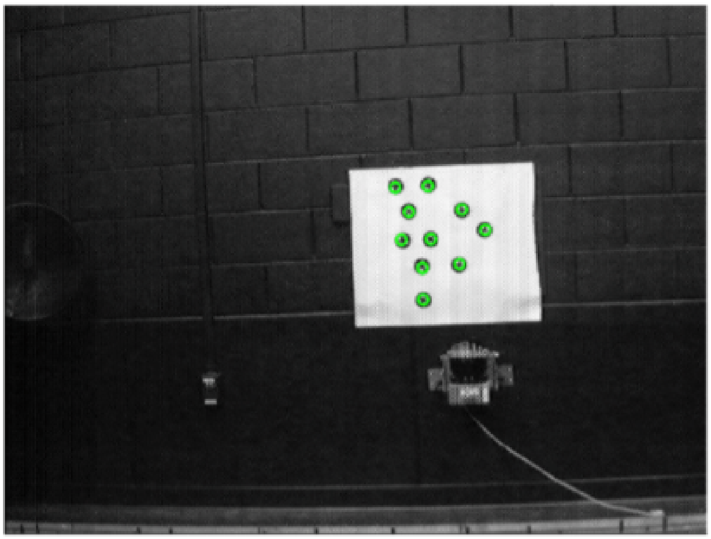


Figure 160: Example of a target pattern as seen by the camera onboard the ASTROS platform.

is run in parallel with landmark acquisition and no hold-time is needed for computation. Future work will focus on the validation of the presented method through the aid of experimental data, and high-fidelity simulation using the robotic simulator and a realistic orbital scenario.

## 7.12 APPLICATION OF THE ALGORITHM TO THE ROBOTIC FACILITY

Currently, the ASTROS platform has been used, with the aid of a satellite mockup, for several relative navigation simulations. The main idea behind this chapter is to apply the Cross Entropy technique to the robotic facility.

Since the base of the robot is not free to translate as in the case of the ASTROS platform, the architecture of the simulations will have to be different.

A way to overcome this limited dexterity is to transfer the orbital motion of the chaser satellite to the target. That is, the target will move with its own tumbling dynamics (if present) with the addition of the relative motion of the target with respect to the chaser (represented by the robot end effector). The robotic facility will hence be responsible for the simulation of the sensor controls and for the eventual rendezvous maneuver.

The diagram in Fig. 161 represents the operating principle of an ASTROS simulation: the target is typically fixed and the platform, using the low friction surface, translates as if it was in orbit. A sensor/camera mounted on the platform (represented by the yellow cone) can then be controlled using the algorithm under analysis to detect some target (Fig. 160) and eventually perform proximity operations.

In the case of the robotic arm facility, the simulation principle is shown in Fig. 162. The target, represented by the satellite mockup, is fixed in terms of translation but has available the three rotative degrees of freedom. To this extent, the University of Padova - CISAS is developing, as of November 2016, a facility that is similar to the upper stages of a 5DOF simulator, using an air bearing joint to provide the rotational degrees of freedom. This will ultimately become the target part of the entire facility and will allow complete OOS simulation capabilities. Regarding the Cross Entropy algorithm, as depicted in the figure, the arm is capable of simulating the motion of the controlled sensor, thus isolating the attitude of the chaser from the control of the sensor itself; in addition, due to the dexterity in the $z$ direction, the arm is able to simulate the approach maneuver up to the docking phase. In the case of a 5DOF platform, this is typically difficult to accomplish since the $z$ coordinate is fixed and limits the rendezvous simulations to a plane.

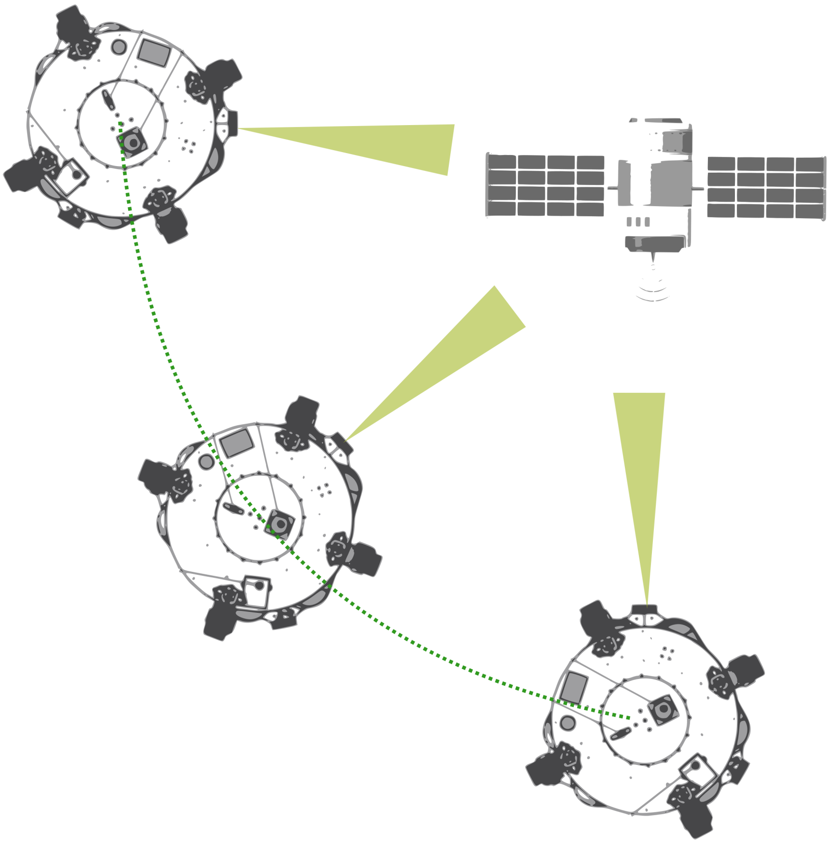


Figure 161: Typical stroboscopic simulation for the ASTROS platform.

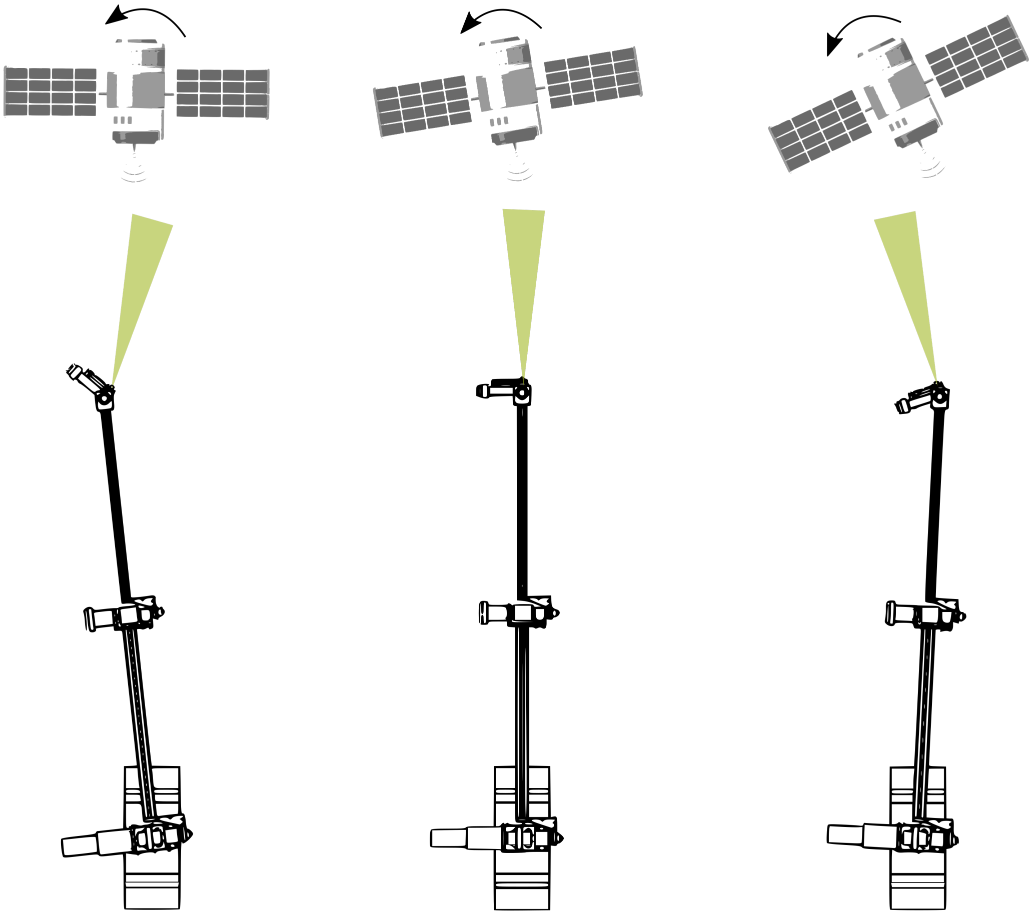


Figure 162: Possible stroboscopic simulation for the robotic arm facility.

# 8 CONCLUSIONS

In this chapter, the results of the doctoral work are summarized and the near and long term future goals are discussed. The thesis goal was to develop a robotic facility for the simulation of orbital servicing operations, from the mechanical design of the anthropomorphic arm to the simulation scenarios to be performed.

The logical process behind thesis was to follow the main research path defined at the beginning of the curriculum, while simultaneously produce original work both in terms of software and hardware, not necessarily niche to the actual facility under study. This vision, which is analyzed in detail in the following paragraphs, has been successfully fulfilled by the author, and led to the development of three main original hardware devices (the robotic facility, a haptic force sensor and a miniaturized attitude sensor) and an information theoretic technique for the performing of SLAM in a rendezvous scenario. The main results obtained are chapter specific and are summarized within the respective section. In synthesis, the findings can be grouped as follows:

The initial modeling, developed in a previous work by this author, was used as the baseline working package from which the subsequent research originated. For this reason, the author did not analyzed the kinematics and dynamics of the arm in depth, since the complete analysis can be found in [10]: however, the dynamics results in terms of torque efforts were accurately revised for the expected simulation maneuvers and were used to adequately size the facility.

A preliminary concept was laid out as a first iteration of the mechanical design. By defining some requirements on the rigidity of the system, a close loop logical process was established in order to find the optimal trade-off among all the mandatory, user defined design parameters. To this extent, an extensive structural model was created *ex-novo* and the analysis of the key loading configurations led to the choice of the mechanical parts that satisfied the requirements. As a by-product of this analysis, a software for the computation of the flexibility-induced end effector-displacement was coded and cross-verified with several finite element simulations

A cohort of FEM static simulations, performed for different joint angles, led to the final verification of the system. In addition, in order to assess the modal response for the application to contact operations, a frequency simulation campaign was performed. Using the results of the modeler, nonlinear regression led to the creation of a continuous model, in terms of possible arm configurations, for the empirical modal response of the facility.

Regarding the control system, a general discussion for the possible strategies conducted the researcher to the choice of a decentralized control. In order to characterize the facility for simulations in which a contact at the end effector is expected to happen (i.e. docking and berthing operations), a novel approach for the impedance matching was presented. This technique, which is based on the virtual force method, enables the facility to extend its capabilities not only to the reproduction of faithful orbital trajectory (already discussed in [10]) but also to perform realistic maneuvers in which the dynamics induced by the contact is simulated according to the (software defined) inertial characteristic of the target-chaser tuple. Thus, this enables to detach the cohort of possible scenarios from the limited inertial combinations obtainable without impedance matching. The control system design chapter introduces ultimately the hardware and software chosen for this application, compatibly with the motor controllers, and defines the bus protocols that will be implemented for the communication with the actuators. Future work consists in the experimental verification of the control software performances.

The simulation scenarios chapter discusses in depth the set of trajectories that the arm will perform, strictly linking the results of the maneuvers with the mechanical design in terms of torque efforts. After a preliminary assessment of the baseline operations (rectilinear and circular trajectories), the chapter focuses on navigation strategies to be applied to the facility; the work originates from a wider simulation framework, in which the application is not limited to the manipulator arm but strives to design a technique to be implemented in synergy with an air bearing platform currently under development at the University of Padova.

An information theoretic approach enabled to solve the rare probability event of simultaneous localization and mapping in a 2D circumnavigation scenario; subsequently, the method was extended to a fully 3D orbital scenario and proved to have promising performances in terms of rate of convergence to the near-optimum. This approach outperformed existing rare-probability techniques (e.g. Montecarlo) and classical PID controllers. Future goal (currently being performed as of November 2016) is the experimental validation of the technique in a relevant environment: following a collaboration with the Georgia Institute of Technology, a simulation campaign is to be carried out on the ASTROS 5DOF simulator at the Dynamics and Control Systems Laboratory [46].

Finally, attention was given to the end effector sensing capabilities, with the aim of creating a facility which is independent from external attitude measurements. That is, the goal was to the enable the bodies under analysis to acquire independent measurements on their state; under this framework, several sensors combinations have been investigated and led to the research and development of a force and attitude sensor. The first one, jointly developed with MIT, is a force sensor based on the GelSight® technology and has not been presented in this thesis due to undergoing intellectual property screenings. The second one, developed at the University of Padova, is an attitude sensor tailored to the rising market segment of small sized vehicles (CubeSats and nanosats). The device, whose working principle is derived from traditional Sun sensors, has been designed to be a low cost, low power, lightweight and high performance sensor and ultimately led to the manufacturing of a working prototype (TRL 4 as of October 2016). The measured performances proved to be consistent with the simulations and an order of magnitude superior with respect to the current academic and commercial state-of-the-art devices.

Future work (currently being performed as of November 2016), will consist in the miniaturization of the prototype and in the increase of the actual maturity level by testing the device in a relevant environment.

# 9 PUBLICATIONS

2017 Antonello A., Olivieri L., Francesconi A.; *“Development of a low-cost Sun Sensor for nanosatellites”*, Acta Astronautica (under review).

2017 Ghassabian H., Calzada A., Hettrich S., De Quattro N., Antonello A., Bielicki D.; *“Alcides: A Novel Lunar Mission Concept Study For The Demonstration Of Enabling Technologies In Deep-Space Exploration And Human-Robots Interaction”* , Acta Astronautica (under review).

2016 Antonello A., Tsiotras P.; *“Vision-Based Attitude Determination Using A SLAM Algorithm During Relative Circumnavigation Of Non-cooperative Objects”* , 67th International Astronautical Congress, Guadalajara, Mexico.

2016 Antonello A., Carron A., Carli R., Tsiotras P.; *“Performance Analysis Of Four Different Cost Policies For The Control Of A Camera In Relative Circumnavigation Scenarios”* , 67th International Astronautical Congress, Guadalajara, Mexico.

2016 Antonello A., Olivieri L., Francesconi A.; *“Development of a low-cost Sun Sensor for nanosatellites”* , 67th International Astronautical Congress, Guadalajara, Mexico.

2016 Feltrin F., Antonello A., Francesconi A.; *“An economic flavored algorithm for spacecraft subsystem management and optimization ”* , 67th International Astronautical Congress, Guadalajara, Mexico.

2016 Ghassabian H., Calzada A., Hettrich S., De Quattro N., Antonello A., Bielicki D.; *"Alcides: A Novel Lunar Mission Concept Study For The Demonstration Of Enabling Technologies In Deep-Space Exploration And Human-Robots Interaction"* , $67^{th}$ International Astronautical Congress, Guadalajara, Mexico.

2016 Inamdar K., Ghassabian H., Garcia A., Antonello A., De Quattro N.; *"Cubesats As Platform For Remote Sensing Applications With Satellite Navigation Signal"* , $67^{th}$ International Astronautical Congress, Guadalajara, Mexico.

2016 Antonello A., Olivieri L., Francesconi A.; *"Low-cost, high-resolution, self-powered, miniaturized Sun sensor for space applications"* ESA 4S Symposium on Small Satellites Systems and Services.

2015 Olivieri L., Antonello A., Duzzi M., Sansone F., Francesconi, A.; *"Semi-androgynous multifunctional interface for expandable space structures"*, Cod: 2-s2.0-84991577920, $66^{th}$ International Astronautical Congress, Jerusalem, Israel.

2015 Branz, F., Antonello A., Francesconi A.; *"Dielectric Elastomer Space Manipulator: Design and Testing"*, Cod: 2-s2.0-84991491464, $66^{th}$ International Astronautical Congress, Jerusalem, Israel.

2015 Francesconi A., Giacomuzzo C., Feltrin F., Antonello A., Savioli C.; *"An Engineering Model to Describe Fragments Clouds Propagating Inside Spacecraft in Consequence of Space Debris Impact on Sandwich Panel Structures"*, Cod: 2-s2.0-84938096483, Acta Astronautica, Volume 116, 31 July 2015, Pages 222-228.

2015 Branz F., Antonello A., Carron A., Carli R., Francesconi A.; *"Kinematics and control of redundant robotic arm based on Dielectric Elastomer Actuators"*. Cod: 2-s2.0-84937239772, in SPIE Smart Structures/NDE 2015, San Diego, United States.

2014 Francesconi A., Giacomuzzo C., Feltrin F., Antonello A., Savioli C.; *"An Engineering Model to Describe Fragments Clouds Propagating Inside Spacecraft in Consequence of Space Debris Impact on Sandwich Panel Structures"*, Cod: s2.0-84937884427, 65th International Astronautical Congress, Toronto, Canada.

2014 Antonello A., Sansone F., Francesconi A., Carli R., Carron A.; *"A novel approach to the simulation of on-orbit rendezvous and docking maneuvers in a laboratory environment through the aid of an anthropomorphic robotic arm"*. Cod: 2-s2.0-84907334520, in Metrology for Aerospace (MetroAeroSpace), 2014 IEEE (pp. 347-352). IEEE.

2014 Olivieri L., Antonello A., Savioli L., Francesconi A.; *"Dynamic Behavior Of A Semi-Androgynous Small Satellite Docking Interface"*, Cod: 2-s2.0-84938232735, 65th International Astronautical Congress, Toronto, Canada.

# BIBLIOGRAPHY

[1] "The robot, http://en.wikipedia.org/wiki/Robot."

[2] M. Spong, *Robot dynamics and control*. Wiley, 1989.

[3] "Space robotics technical committee http://ewh.ieee.org/cmte/ras/tc/spacerobotics/."

[4] A. Long, M. Richards, and D. Hastings, "On-orbit servicing: A new value proposition for satellite design and operation," *Journal of Spacecraft and Rockets*, vol. 44, no. 4, July–August 2007.

[5] B. Sullivan and D. Akin, "A survey of serviceable spacecraft failures," *AIAA Paper*, 2001.

[6] J. Saleh, R. Hassan, J. Torres-Padilla, D. Hastings, and D. Newman, "To reduce or extend a spacecraft design lifetime," *Journal of Spacecraft and Rockets*, vol. 43, no. 1, pp. 207–217, 2006.

[7] "Space infrastructure servicing http://en.wikipedia.org/wiki/Space_Infrastructure_Servicing."

[8] P. B. Selding, "Intelsat signs up for mda's satellite refueling service http://www.spacenews.com/article/intelsat-signs-satellite-refueling-service."

[9] P. B. Selding, "Mda, intelsat scrap in-orbit servicing deal."

[10] A. Antonello, "Design of a robotic arm for laboratory simulations of spacecraft proximity navigation and docking," Master's thesis, University of Padova, 2013.

[11] A. Antonello, F. Sansone, A. Francesconi, R. Carli, and A. Carron, "A novel approach to the simulation of on-orbit rendezvous and docking maneuvers in a laboratory environment through the aid of an anthropomorphic robotic arm," in *Metrology for Aerospace (MetroAeroSpace), 2014 IEEE*, pp. 347–352, IEEE, 2014.

[12] "European proximity operations simulator (epos) - dlr http://www.weblab.dlr.de/rbrt/OOS/EPOS/EPOS.html."

[13] L. Sciavicco and B. Siciliano, *Robotics: Modelling, Planning and Control*. Springer Verlag, 2010.

[14] J. Craig, *Intorduction to Robotics: Mechanics and Control*. Prentice Hall, 2005.

[15] K. R. Symon, *Mechanics*. Addison-Wesley, 1971.

[16] L. Sciavicco and B. Siciliano, *Modelling and control of robot manipulators*. Springer Verlag, 2010.

[17] J. Y. Luh, M. W. Walker, and R. P. Paul, "On-line computational scheme for mechanical manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 102, no. 2, pp. 69–76, 1980.

[18] W. Yeadon and A. Yeadon, *Handbook of small electric motors*. McGraw Hill Professional, 2001.

[19] R. W. Clough and J. Penzien, "Dynamics of structures," tech. rep., 1975.

[20] T. H. G. Megson, *An introduction to Aircraft Structural Analysis*. Elsevier, 2010.

[21] S. Ramachandran, T. Nagarajan, and N. S. Prasad, "A finite element approach to the design and dynamic analysis of platform type robot

manipulators," *Finite elements in Analysis and Design*, vol. 10, no. 4, pp. 335–350, 1992.

[22] G. Palmieri, M. Martarelli, M. Palpacelli, and L. Carbonari, "Configuration-dependent modal analysis of a cartesian parallel kinematics manipulator: numerical modeling and experimental validation," *Meccanica*, vol. 49, no. 4, pp. 961–972, 2014.

[23] M. Barbetta, A. Boesso, F. Branz, A. Carron, L. Olivieri, J. Prendin, G. Rodeghiero, F. Sansone, L. Savioli, F. Spinello, *et al.*, "Arcade-r2 experiment on board bexus 17 stratospheric balloon," *CEAS Space Journal*, vol. 7, no. 3, pp. 347–358, 2015.

[24] L. Olivieri and A. Francesconi, "Design and test of a semiandrogynous docking mechanism for small satellites," *Acta Astronautica*, vol. 122, pp. 219–230, 2016.

[25] F. Sansone, F. Branz, A. Francesconi, M. Barbetta, and M. Pelizzo, "2d close-range navigation sensor for miniature cooperative spacecraft," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 1, pp. 160–169, 2014.

[26] F. Sansone, L. Olivieri, and A. Francesconi, "New optical communication capabilities using nanosatellites," in *Proc. Fifth International Conference on Advances in Satellite and Space Communications*, 2013.

[27] F. Sansone, A. Francesconi, L. Olivieri, and F. Branz, "Low-cost relative navigation sensors for miniature spacecraft and drones," in *Metrology for Aerospace (MetroAeroSpace), 2015 IEEE*, pp. 389–394, IEEE, 2015.

[28] "Sun sensor for small satellites with analog interface, http://www.solar-mems.com/smt_pdf/commercial_brochure_SSOC-A.pdf."

[29] M. A. Post, J. Li, and R. Lee, "A low-cost photodiode sun sensor for cubesat and planetary microrover," *International Journal of Aerospace Engineering*, vol. 2013, 2013.

[30] A. Ali and F. Tanveer, "Low-cost design and development of 2-axis digital sun sensor," *Journal of Space Technology*, vol. 1, no. 1, 2011.

[31] J. Hales and M. Pedersen, "Two-axis moems sun sensor for pico satellites," in *Small Satellite Conference*, 2002.

[32] M. Buonocore, M. Grassi, and G. Rufino, "Aps-based miniature sun sensor for earth observation nanosatellites," *Acta Astronautica*, vol. 56, no. 1, pp. 139–145, 2005.

[33] G. Rufino and M. Grassi, "Digital sun sensor multi-spot operation," *Sensors*, vol. 12, no. 12, pp. 16451–16465, 2012.

[34] G. Rufino and M. Grassi, "Multi-aperture cmos sun sensor for microsatellite attitude determination," *Sensors*, vol. 9, no. 6, pp. 4503–4524, 2009.

[35] A. Antonello, L. Olivieri, and A. Francesconi, "Low cost, high resolution, self powered, miniaturized sun sensor for space applications," in *4S 2016 Proceedings*, 2016.

[36] P. Chiu, D. Law, R. Woo, S. Singer, D. Bhusari, W. Hong, A. Zakaria, J. Boisvert, S. Mesropian, R. King, *et al.*, "Direct semiconductor bonded 5j cell for space and terrestrial applications," *IEEE Journal of Photovoltaics*, vol. 4, no. 1, pp. 493–497, 2014.

[37] R. C. Dorf and B. R. H., *Modern Control Systems*. Prentice Hall, 12th ed., 2012.

[38] T. Bajd and M. Mihelj, *Robotics*. Springer Verlag, 2010.

[39] H. D. Curtis, *Orbital Mechanics for Engineering Students*. Elsevier, 2010.

[40] G. Gilardi and I. Sharf, "Literature survey of contact dynamics modelling," *Mechanism and machine theory*, vol. 37, no. 10, pp. 1213–1239, 2002.

[41] M. Zebenay, R. Lampariello, T. Boge, and D. Choukroun, "A new contact dynamics model tool for hardware-in-the-loop docking dimulation," *i-SAIRAS, Turin, Italy*, 2012.

[42] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Johns Hopkins University Press, Baltimore, 1989.

[43] D. Sternberg, A. Hilton, D. Miller, B. McCarthy, C. Jewison, D. Roascio, J. James, and A. Saenz-Otero, "Reconfigurable ground and flight testing facility for robotic servicing, capture, and assembly," in *Aerospace Conference, 2016 IEEE*, pp. 1–13, IEEE, 2016.

[44] J. L. Schwartz, M. A. Peck, and C. D. Hall, "Historical review of air-bearing spacecraft simulators," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 4, pp. 513–522, 2003.

[45] X. Jian, B. Gang, Y. QinJun, and L. Jun, "Design and development of a 5-dof air-bearing spacecraft simulator," in *2009 International Asia Conference on Informatics in Control, Automation and Robotics*, pp. 126–130, IEEE, 2009.

[46] P. Tsiotras, "Astros: A 5dof experimental facility for research in space proximity operations," in *37th AAS Guidance and Control Conference, Breckenridge, CO. AAS Paper*, vol. 114, 2014.

[47] D.-M. Cho, D. Jung, and P. Tsiotras, "A 5-dof experimental platform for autonomous spacecraft rendezvous and docking," *AIAA Paper*, vol. 1869, p. 2009, 2009.

[48] R. Ambrose, B. Wilcox, B. Reed, L. Matthies, D. Lavery, and D. Korsmeyer, "Draft robotics, tele-robotics and autonomous systems

roadmap," *NASAs Space Technology Roadmaps, National Aeronautics and Space Administration*, 2010.

[49] J. L. Junkins, D. C. Hughes, D. C. Hughes, K. P. Wazni, K. P. Wazni, V. Pariyapong, and V. Pariyapong, "Vision-based navigation for rendezvous, docking and proximity operations," in *AAS Guidance and Control Conference*, (Breckenridge, Colorado), pp. 3–7, 7-10 February 1999.

[50] P. Singla, K. Subbarao, and J. L. Junkins, "Adaptive output feedback control for spacecraft rendezvous and docking under measurement uncertainty," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 4, pp. 892–902, 2006.

[51] J. M. Kelsey, J. Byrne, M. Cosgrove, S. Seereeram, and R. K. Mehra, "Vision-based relative pose estimation for autonomous rendezvous and docking," in *IEEE Aerospace Conference*, (Big Sky, MT, USA), p. 20, 4-11 March 2006.

[52] B. E. Tweddle, *Computer Vision-based Localization And Mapping Of An Unknown, Uncooperative And Spinning Target For Spacecraft Proximity Operations*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2013.

[53] S. Augenstein, S. M. Rock, P. Enge, and C. J. Tomlin, *Monocular Pose And Shape Estimation Of Moving Targets, For Autonomous Rendezvous And Docking*. PhD thesis, Stanford University, 2011.

[54] H. Curtis, *Orbital Mechanics for Engineering Students*. Butterworth-Heinemann, Burlington, MA, 2013.

[55] L. Ljung, "Asymptotic behavior of the extended kalman filter as a parameter estimator for linear systems," *IEEE Transactions on Automatic Control*, vol. 24, no. 1, pp. 36–50, 1979.

[56] K. Fujii, "Extended kalman filter," *Refernce Manual*, 2013.

[57] D. Rodriguez-Losada, F. Matia, A. Jimenez, and R. Galán, "Consistency improvement for slam-ekf for indoor environments," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 2006.

[58] S. Huang and G. Dissanayake, "Convergence analysis for extended kalman filter based slam," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pp. 412–417, IEEE, 2006.

[59] Z. I. Botev, D. P. Kroese, R. Y. Rubinstein, P. L'Ecuyer, *et al.*, *The Cross-Entropy Method For Optimization*, vol. 31. 2013.

[60] D. P. Kroese and R. Y. Rubinstein, "Monte carlo methods," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 4, no. 1, pp. 48–58, 2012.

[61] K. Alfriend, S. R. Vadali, P. Gurfil, J. How, and L. Breger, *Spacecraft Formation Flying: Dynamics, Control And Navigation*, vol. 2. Butterworth-Heinemann, Burlington, MA, 2009.

[62] H. Schaub and J. L. Junkins, *Analytical Mechanics of Space Systems*. AIAA, 2003.

[63] D. A. Vallado, *Fundamentals of Astrodynamics and Applications*, vol. 12. Springer Science & Business Media, 2001.

[64] M. Kontitsis, P. Tsiotras, and E. Theodorou, "An information-theoretic active localization approach during relative circumnavigation in orbit," in *AIAA Guidance, Navigation, and Control Conference*, (San Diego, California, USA), p. 872, 4-8 January 2016.

[65] C. Li and P. K.-S. Tam, "An iterative algorithm for minimum cross entropy thresholding," *Pattern Recognition Letters*, vol. 19, no. 8, pp. 771–776, 1998.

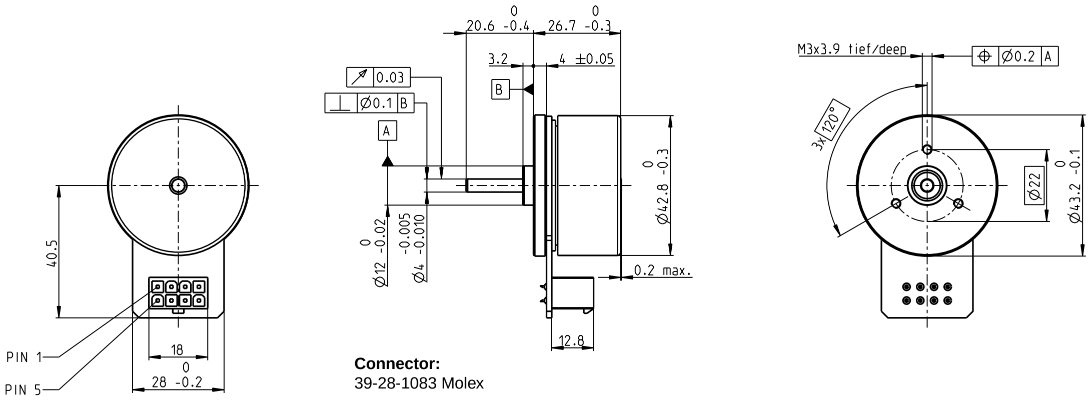
# LIST OF FIGURES

# LIST OF TABLES

# APPENDIX

1 x 397172

# EC 45 flat ∅42.8 mm, brushless, 70 Watt



**Connector:**
39-28-1083 Molex

M 1:2

🟥 Stock program
□ Standard program
▨ Special program (on request)

## Part Numbers

| | | | | 397172 | 402685 | 402686 | 402687 |
|---|---|---|---|---|---|---|---|
| | with Hall sensors | | | 397172 | 402685 | 402686 | 402687 |
| **Motor Data** (provisional) | | | | | | | |
| | **Values at nominal voltage** | | | | | | |
| 1 | Nominal voltage | V | | 24 | 30 | 36 | 48 |
| 2 | No load speed | rpm | | 6110 | 6230 | 6330 | 3440 |
| 3 | No load current | mA | | 234 | 194 | 166 | 48.1 |
| 4 | Nominal speed | rpm | | 4860 | 4990 | 5080 | 2540 |
| 5 | Nominal torque (max. continuous torque) | mNm | | 128 | 112 | 108 | 134 |
| 6 | Nominal current (max. continuous current) | A | | 3.21 | 2.36 | 1.93 | 0.936 |
| 7 | Stall torque | mNm | | 1460 | 1170 | 1100 | 915 |
| 8 | Stall current | A | | 39.5 | 25.8 | 20.7 | 6.97 |
| 9 | Max. efficiency | % | | 85 | 84 | 83 | 84 |
| | **Characteristics** | | | | | | |
| 10 | Terminal resistance phase to phase | Ω | | 0.608 | 1.16 | 1.74 | 6.89 |
| 11 | Terminal inductance phase to phase | mH | | 0.463 | 0.691 | 0.966 | 5.85 |
| 12 | Torque constant | mNm / A | | 36.9 | 45.1 | 53.3 | 131 |
| 13 | Speed constant | rpm / V | | 259 | 212 | 179 | 72.7 |
| 14 | Speed / torque gradient | rpm / mNm | | 4.26 | 5.44 | 5.85 | 3.82 |
| 15 | Mechanical time constant | ms | | 8.07 | 10.3 | 11.1 | 7.24 |
| 16 | Rotor inertia | gcm$^2$ | | 181 | 181 | 181 | 181 |

## Specifications

**Thermal data**

| | | |
|---|---|---|
| 17 | Thermal resistance housing-ambient | 3.56 K/W |
| 18 | Thermal resistance winding-housing | 4.1 K/W |
| 19 | Thermal time constant winding | 29.6 s |
| 20 | Thermal time constant motor | 178 s |
| 21 | Ambient temperature | -40 ... +100°C |
| 22 | Max. winding temperature | +125°C |

**Mechanical data (preloaded ball bearings)**

| | | | |
|---|---|---|---|
| 23 | Max. speed | | 10 000 rpm |
| 24 | Axial play at axial load | < 4.0 N | 0 mm |
| | | > 4.0 N | 0.14 mm |
| 25 | Radial play | | preloaded |
| 26 | Max. axial load (dynamic) | | 3.8 N |
| 27 | Max. force for press fits (static) | | 50 N |
| | (static, shaft supported) | | 1000 N |
| 28 | Max. radial load, 5 mm from flange | | 21 N |

**Other specifications**

| | | |
|---|---|---|
| 29 | Number of pole pairs | 8 |
| 30 | Number of phases | 3 |
| 31 | Weight of motor | 141 g |

Values listed in the table are nominal.

**Connection**

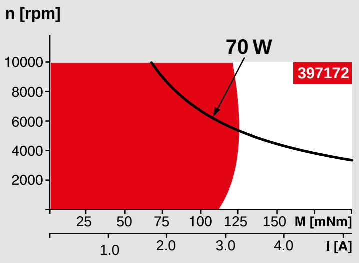
| | |
|---|---|
| Pin 1 | Hall sensor 1* |
| Pin 2 | Hall sensor 2* |
| Pin 3 | $V_{Hall}$ 4.5 ... 18 VDC |
| Pin 4 | Motor winding 3 |
| Pin 5 | Hall sensor 3* |
| Pin 6 | GND |
| Pin 7 | Motor winding 1 |
| Pin 8 | Motor winding 2 |

*Internal pull-up (7 … 13 kΩ) on pin 3
Wiring diagram for Hall sensors see p. 35

**Cable**

| | |
|---|---|
| Connection cable Universal, L = 500 mm | **339380** |
| Connection cable to EPOS, L = 500 mm | **354045** |

## Operating Range



## Comments

🟥 **Continuous operation**
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.
= Thermal limit.

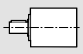
□ **Short term operation**
The motor may be briefly overloaded (recurring).

— **Assigned power rating**

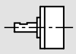
## maxon Modular System

Overview on page 20–25

**Planetary Gearhead**
∅42 mm
3 - 15 Nm
Page 316



**Spur Gearhead**
∅45 mm
0.5 - 2.0 Nm
Page 317



**Encoder MILE**
256 - 2048 CPT,
2 channels
Page 342

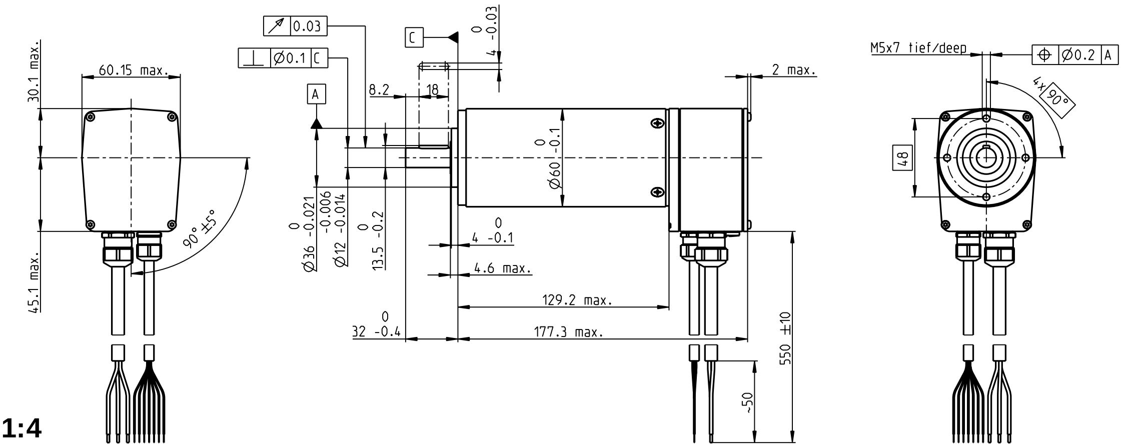
**Recommended Electronics:**

| **Notes** | **Page 24** |
|---|---|
| ESCON 36/3 EC | 379 |
| ESCON Mod. 50/4 EC-S | 379 |
| ESCON Module 50/5 | 379 |
| ESCON 50/5 | 380 |
| DEC Module 50/5 | 382 |
| EPOS2 Module 36/2 | 386 |
| EPOS2 24/5, 50/5 | 387 |
| EPOS2 P 24/5 | 390 |
| EPOS3 70/10 EtherCAT | 393 |
| MAXPOS 50/5 | 396 |

**Option**
With Cable and Connector
(Ambient temperature -20 ... +100°C)

1 x 167131

# EC 60 ∅60 mm, brushless, 400 Watt



M 1:4

| | |
|---|---|
| 🟥 | Stock program |
| □ | Standard program |
| ░ | Special program (on request) |

## Part Numbers

## Motor Data

| | | | 167132 | 167131 |
|---|---|---|---|---|
| | **Values at nominal voltage** | | | |
| 1 | Nominal voltage | V | 48 | 48 |
| 2 | No load speed | rpm | 5380 | 3100 |
| 3 | No load current | mA | 449 | 194 |
| 4 | Nominal speed | rpm | 4940 | 2670 |
| 5 | Nominal torque (max. continuous torque) | mNm | 810 | 859 |
| 6 | Nominal current (max. continuous current) | A | 9.84 | 5.94 |
| 7 | Stall torque | mNm | 11800 | 6820 |
| 8 | Stall current | A | 139 | 46.4 |
| 9 | Max. efficiency | % | 89 | 88 |
| | **Characteristics** | | | |
| 10 | Terminal resistance phase to phase | Ω | 0.345 | 1.03 |
| 11 | Terminal inductance phase to phase | mH | 0.273 | 0.82 |
| 12 | Torque constant | mNm/A | 84.9 | 147 |
| 13 | Speed constant | rpm/V | 113 | 65 |
| 14 | Speed/torque gradient | rpm/mNm | 0.457 | 0.457 |
| 15 | Mechanical time constant | ms | 3.98 | 3.98 |
| 16 | Rotor inertia | $gcm^2$ | 831 | 831 |

## Specifications

| | | |
|---|---|---|
| | **Thermal data** | |
| 17 | Thermal resistance housing-ambient | 1.3 K/W |
| 18 | Thermal resistance winding-housing | 0.5 K/W |
| 19 | Thermal time constant winding | 33.9 s |
| 20 | Thermal time constant motor | 1200 s |
| 21 | Ambient temperature | -20…+100°C |
| 22 | Max. winding temperature | +125°C |
| | **Mechanical data** (preloaded ball bearings) | |
| 23 | Max. speed | 7000 rpm |
| 24 | Axial play at axial load < 30 N | 0 mm |
| | > 30 N | max. 0.14 mm |
| 25 | Radial play | preloaded |
| 26 | Max. axial load (dynamic) | 24 N |
| 27 | Max. force for press fits (static) | 390 N |
| | (static, shaft supported) | 6000 N |
| 28 | Max. radial load, 5 mm from flange | 240 N |
| | **Other specifications** | |
| 29 | Number of pole pairs | 1 |
| 30 | Number of phases | 3 |
| 31 | Weight of motor | 2400 g |
| | Protection to | IP54* |

Values listed in the table are nominal.

**Connection motor** (Cable AWG 16)

| | |
|---|---|
| Cable 1 | Motor winding 1 |
| Cable 2 | Motor winding 2 |
| Cable 3 | Motor winding 3 |

**Connection sensors** (Cable AWG 24)[1)]

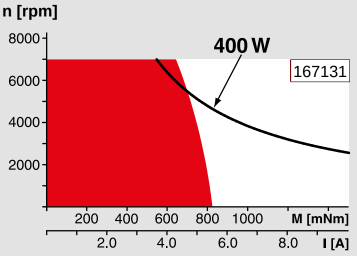
| | |
|---|---|
| white | Hall sensor 3 |
| brown | Hall sensor 2 |
| green | Hall sensor 1 |
| yellow | GND |
| grey | $V_{Hall}$ 4.5 … 24 VDC |
| blue | Temperature sensor (PTC) |
| pink | Temperature sensor (PTC) |

[1)] Not lead through in combination with resolver.

Temperature monitoring, PTC resistance Micropille 110°C, R 25°C < 0.5 kΩ, R 105°C = 1.2…1.5 kΩ, R 115°C = 7…13 kΩ, R 120°C = 18…35 kΩ

Wiring diagram for Hall sensors see p. 33

## Operating Range



Comments

🟥 **Continuous operation**
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.
= Thermal limit.

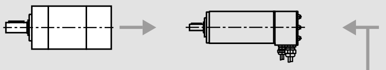
□ **Short term operation**
The motor may be briefly overloaded (recurring).

— **Assigned power rating**

## maxon Modular System

Overview on page 20–25

**Planetary Gearhead**
∅81 mm
20 - 120 Nm
Page 321



**Encoder HEDL 9140**
500 CPT,
3 channels
Page 368

**Resolver Res**
∅26 mm
10 V
Page 374

**Brake AB 41**
24 VDC
2.0 Nm
Page 411

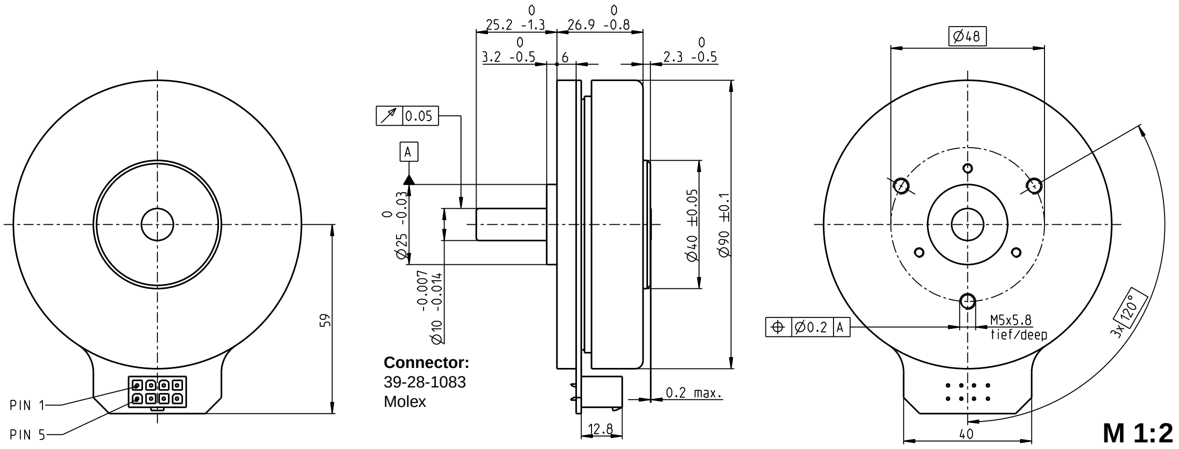
**Recommended Electronics:**

| **Notes** | Page **24** |
|---|---|
| ESCON Mod. 50/5 | 379 |
| ESCON Mod. 50/4 EC-S | 379 |
| ESCON 50/5 | 380 |
| ESCON 70/10 | 380 |
| DEC Module 50/5 | 382 |
| EPOS2 50/5, 70/10 | 387 |
| EPOS3 70/10 EtherCAT | 393 |
| MAXPOS 50/5 | 396 |

*Protection level only when installed with flange-side seal.

# 2 x 323772

## EC 90 flat ∅90 mm, brushless, 90 Watt



Connector:
39-28-1083
Molex

🟥 Stock program
□ Standard program
▨ Special program (on request)

**Part Numbers**

| | | | 323772 | 429271 | 244879 |
|---|---|---|---|---|---|
| | with Hall sensors | | 323772 | 429271 | 244879 |
| | **Motor Data** | | | | |
| | **Values at nominal voltage** | | | | |
| 1 | Nominal voltage | V | 24 | 36 | 48 |
| 2 | No load speed | rpm | 3190 | 3120 | 2080 |
| 3 | No load current | mA | 544 | 348 | 135 |
| 4 | Nominal speed | rpm | 2590 | 2510 | 1610 |
| 5 | Nominal torque (max. continuous torque) | mNm | 444 | 560 | 533 |
| 6 | Nominal current (max. continuous current) | A | 6.06 | 4.76 | 2.27 |
| 7 | Stall torque | mNm | 4940 | 7480 | 4570 |
| 8 | Stall current | A | 70 | 69 | 21.1 |
| 9 | Max. efficiency | % | 84 | 87 | 85 |
| | **Characteristics** | | | | |
| 10 | Terminal resistance phase to phase | Ω | 0.343 | 0.522 | 2.28 |
| 11 | Terminal inductance phase to phase | mH | 0.264 | 0.625 | 2.5 |
| 12 | Torque constant | mNm/A | 70.5 | 109 | 217 |
| 13 | Speed constant | rpm/V | 135 | 88 | 44 |
| 14 | Speed/torque gradient | rpm/mNm | 0.659 | 0.423 | 0.462 |
| 15 | Mechanical time constant | ms | 21.1 | 13.6 | 14.8 |
| 16 | Rotor inertia | gcm² | 3060 | 3060 | 3060 |

**Specifications**

| | | |
|---|---|---|
| | **Thermal data** | |
| 17 | Thermal resistance housing-ambient | 1.91 K/W |
| 18 | Thermal resistance winding-housing | 2.6 K/W |
| 19 | Thermal time constant winding | 46 s |
| 20 | Thermal time constant motor | 283 s |
| 21 | Ambient temperature | -40…+100°C |
| 22 | Max. winding temperature | +125°C |
| | **Mechanical data** (preloaded ball bearings) | |
| 23 | Max. speed | 5000 rpm |
| 24 | Axial play at axial load < 15 N | 0 mm |
| | > 15 N | 0.14 mm |
| 25 | Radial play | preloaded |
| 26 | Max. axial load (dynamic) | 12 N |
| 27 | Max. force for press fits (static) | 183 N |
| | (static, shaft supported) | 8000 N |
| 28 | Max. radial load, 5 mm from flange | 68 N |
| | **Other specifications** | |
| 29 | Number of pole pairs | 12 |
| 30 | Number of phases | 3 |
| 31 | Weight of motor | 600 g |

Values listed in the table are nominal.

**Connection**

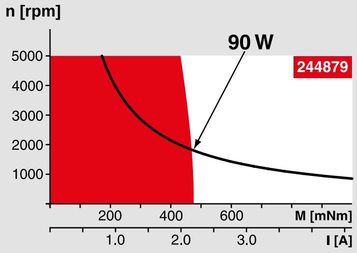
| | |
|---|---|
| Pin 1 | Hall sensor 1 |
| Pin 2 | Hall sensor 2 |
| Pin 3 | $V_{Hall}$ 4.5…18 VDC |
| Pin 4 | Motor winding 3 |
| Pin 5 | Hall sensor 3 |
| Pin 6 | GND |
| Pin 7 | Motor winding 1 |
| Pin 8 | Motor winding 2 |

Wiring diagram for Hall sensors see p. 35

**Cable**

| | |
|---|---|
| Connection cable Universal, L = 500 mm | **339380** |
| Connection cable to EPOS2, L = 500 mm | **354045** |

**Operating Range**



**Comments**

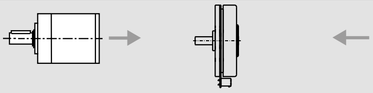
🟥 **Continuous operation**
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.
= Thermal limit.

□ **Short term operation**
The motor may be briefly overloaded (recurring).

— **Assigned power rating**

**maxon Modular System** Overview on page 20–25

**Planetary Gearhead**
∅52 mm
4 - 30 Nm
Page 319



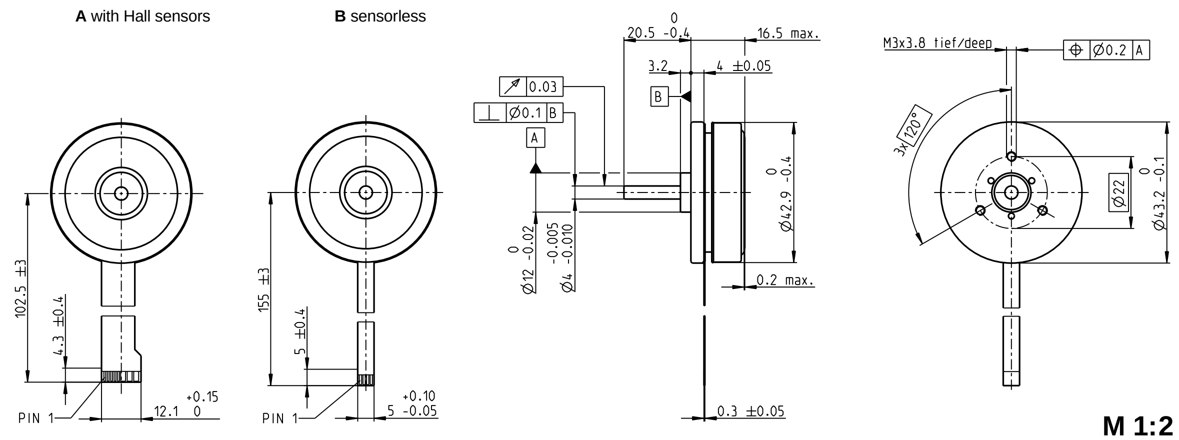
**Encoder MILE**
512 - 6400 CPT,
2 channels
Page 344

**Recommended Electronics:**

| **Notes** | Page **24** |
|---|---|
| ESCON Mod. 50/4 EC-S | 379 |
| ESCON Mod. 50/5 | 379 |
| ESCON 50/5 | 380 |
| ESCON 70/10 | 380 |
| DEC Module 50/5 | 382 |
| EPOS2 24/5, 50/5, 70/10 | 387 |
| EPOS2 P 24/5 | 390 |
| EPOS3 70/10 EtherCAT | 393 |
| MAXPOS 50/5 | 396 |

# 2 x 200142

## **EC 45 flat** ∅42.9 mm, brushless, 30 Watt



**M 1:2**

| | |
|---|---|
| 🟥 | Stock program |
| □ | Standard program |
| ▨ | Special program (on request) |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | **Part Numbers** | | | | | | | |
| | **A** with Hall sensors | | 200142 | | 339281 | | 339282 | | | |
| | **B** sensorless | | | 200189 | | 339283 | | 339284 | | |
| | **Motor Data** | | | | | | | | | |
| | **Values at nominal voltage** | | | | | | | | | |
| 1 | Nominal voltage | V | 12 | 12 | 24 | 24 | 36 | 36 | | |
| 2 | No load speed | rpm | 4370 | 4350 | 4360 | 4380 | 4750 | 4760 | | |
| 3 | No load current | mA | 163 | 163 | 81.4 | 73 | 61.6 | 55.3 | | |
| 4 | Nominal speed | rpm | 2940 | 2800 | 2940 | 2900 | 3290 | 3270 | | |
| 5 | Nominal torque (max. continuous torque) | mNm | 55 | 54.7 | 54.8 | 55.2 | 66 | 66.6 | | |
| 6 | Nominal current (max. continuous current) | A | 2.02 | 2.02 | 1.01 | 1.01 | 0.847 | 0.849 | | |
| 7 | Stall torque | mNm | 255 | 219 | 253 | 243 | 380 | 369 | | |
| 8 | Stall current | A | 10 | 8.58 | 4.97 | 4.77 | 5.38 | 5.22 | | |
| 9 | Max. efficiency | % | 76 | 75 | 76 | 77 | 80 | 81 | | |
| | **Characteristics** | | | | | | | | | |
| 10 | Terminal resistance phase to phase | Ω | 1.2 | 1.4 | 4.83 | 5.03 | 6.69 | 6.89 | | |
| 11 | Terminal inductance phase to phase | mH | 0.56 | 0.56 | 2.24 | 2.24 | 4.29 | 4.29 | | |
| 12 | Torque constant | mNm/A | 25.5 | 25.5 | 51 | 51 | 70.6 | 70.6 | | |
| 13 | Speed constant | rpm/V | 374 | 374 | 187 | 187 | 135 | 135 | | |
| 14 | Speed/torque gradient | rpm/mNm | 17.6 | 20.5 | 17.7 | 18.5 | 12.8 | 13.2 | | |
| 15 | Mechanical time constant | ms | 17.1 | 19.9 | 17.2 | 17.9 | 12.4 | 12.8 | | |
| 16 | Rotor inertia | gcm$^2$ | 92.5 | 92.5 | 92.5 | 92.5 | 92.5 | 92.5 | | |

## Specifications

| | | |
|---|---|---|
| | **Thermal data** | |
| 17 | Thermal resistance housing-ambient | 6.69 K/W |
| 18 | Thermal resistance winding-housing | 3.92 K/W |
| 19 | Thermal time constant winding | 11.4 s |
| 20 | Thermal time constant motor | 295 s |
| 21 | Ambient temperature | -40…+100°C |
| 22 | Max. winding temperature | +125°C |
| | **Mechanical data (preloaded ball bearings)** | |
| 23 | Max. speed | 10000 rpm |
| 24 | Axial play at axial load < 5.0 N | 0 mm |
| | > 5.0 N | typ. 0.14 mm |
| 25 | Radial play | preloaded |
| 26 | Max. axial load (dynamic) | 4.8 N |
| 27 | Max. force for press fits (static)<br>(static, shaft supported) | 53 N<br>1000 N |
| 28 | Max. radial load, 5 mm from flange | 18 N |
| | **Other specifications** | |
| 29 | Number of pole pairs | 8 |
| 30 | Number of phases | 3 |
| 31 | Weight of motor | 75 g |

Values listed in the table are nominal.

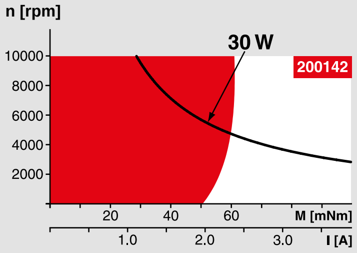
| Connection | with Hall sensors | sensorless |
|---|---|---|
| Pin 1 | $V_{Hall}$ 4.5…18 VDC | Motor winding 1 |
| Pin 2 | Hall sensor 3* | Motor winding 2 |
| Pin 3 | Hall sensor 1* | Motor winding 3 |
| Pin 4 | Hall sensor 2* | ⅄ neutral point |
| Pin 5 | GND | |
| Pin 6 | Motor winding 3 | |
| Pin 7 | Motor winding 2 | |
| Pin 8 | Motor winding 1 | |

*Internal pull-up (7…13 kΩ) on pin 1
Wiring diagram for Hall sensors see p. 35

| Adapter | Part number | Part number |
|---|---|---|
| see p. 398 | 220300 | 220310 |
| **Connector** | **Part number** | **Part number** |
| Tyco | 1-84953-1 | 84953-4 |
| Molex | 52207-1133 | 52207-0433 |
| Molex | 52089-1119 | 52089-0419 |

Pin for design with Hall sensors:
FPC, 11-pol, Pitch 1.0 mm, top contact style

## Operating Range



## Comments

🟥 **Continuous operation**
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.
= Thermal limit.

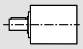
□ **Short term operation**
The motor may be briefly overloaded (recurring).

— **Assigned power rating**

## maxon Modular System

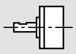
Overview on page 20–25

**Planetary Gearhead**
∅42 mm
3 - 15 Nm
Page 316



**Encoder MILE**
256 - 2048 CPT,
2 channels
Page 342

**Spur Gearhead**
∅45 mm
0.5 - 2.0 Nm
Page 317

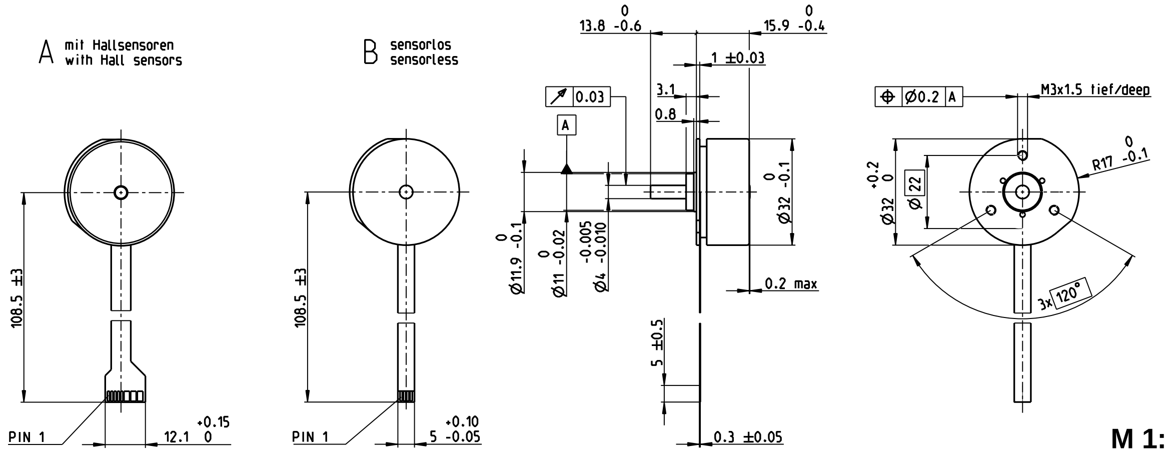


**Recommended Electronics:**

| **Notes** | Page **24** |
|---|---|
| ESCON Module 24/2 | 378 |
| ESCON 36/3 EC | 379 |
| ESCON Mod. 50/4 EC-S | 379 |
| ESCON Module 50/5 | 379 |
| ESCON 50/5 | 380 |
| DEC Module 24/2 | 382 |
| DEC Module 50/5 | 382 |
| EPOS2 24/2, Module 36/2 | 386 |
| EPOS2 24/5, 50/5 | 387 |
| EPOS2 P 24/5 | 390 |
| EPOS3 70/10 EtherCAT | 393 |
| MAXPOS 50/5 | 396 |

**Option**
With Cable and Connector
(Motor length +1.3 mm,
Ambient temperature -20…+100°C)

# EC 32 flat ∅32 mm, brushless, 15 Watt



M 1:2

| | |
|---|---|
| 🟥 | Stock program |
| □ | Standard program |
| ▨ | Special program (on request) |

## Part Numbers

| | | | | |
|---|---|---|---|---|
| **A** with Hall sensors | 339267 | 339268 | **267121** | 339269 |
| **B** sensorless | 339271 | 339272 | **226006** | 339273 |

## Motor Data

| | | | | | | |
|---|---|---|---|---|---|---|
| | **Values at nominal voltage** | | | | | |
| 1 | Nominal voltage | V | 9 | 12 | 24 | 48 |
| 2 | No load speed | rpm | 3720 | 4610 | 4530 | 4780 |
| 3 | No load current | mA | 74.7 | 75.7 | 36.9 | 19.9 |
| 4 | Nominal speed | rpm | 2090 | 2810 | 2760 | 2940 |
| 5 | Nominal torque (max. continuous torque) | mNm | 24.6 | 25.1 | 25.5 | 24.7 |
| 6 | Nominal current (max. continuous current) | A | 1.06 | 1 | 0.5 | 0.257 |
| 7 | Stall torque | mNm | 70 | 84.1 | 85.8 | 84.1 |
| 8 | Stall current | A | 3.13 | 3.49 | 1.75 | 0.906 |
| 9 | Max. efficiency | % | 72 | 73 | 74 | 73 |
| | **Characteristics** | | | | | |
| 10 | Terminal resistance phase to phase | Ω | 2.87 | 3.43 | 13.7 | 53 |
| 11 | Terminal inductance phase to phase | mH | 1.61 | 1.87 | 7.73 | 27.8 |
| 12 | Torque constant | mNm/A | 22.4 | 24.1 | 49 | 92.8 |
| 13 | Speed constant | rpm/V | 427 | 397 | 195 | 103 |
| 14 | Speed/torque gradient | rpm/mNm | 54.9 | 56.6 | 54.5 | 58.7 |
| 15 | Mechanical time constant | ms | 20.1 | 20.7 | 20 | 21.5 |
| 16 | Rotor inertia | $gcm^2$ | 35 | 35 | 35 | 35 |

## Specifications

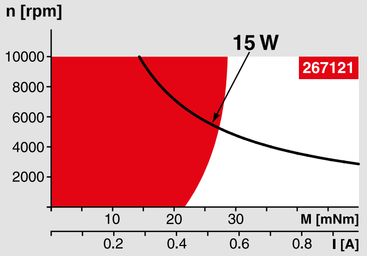
| | | |
|---|---|---|
| | **Thermal data** | |
| 17 | Thermal resistance housing-ambient | 10.8 K/W |
| 18 | Thermal resistance winding-housing | 4.99 K/W |
| 19 | Thermal time constant winding | 8.78 s |
| 20 | Thermal time constant motor | 120 s |
| 21 | Ambient temperature | -40…+100°C |
| 22 | Max. winding temperature | +125°C |
| | **Mechanical data (preloaded ball bearings)** | |
| 23 | Max. speed | 10 000 rpm |
| 24 | Axial play at axial load < 5.0 N | 0 mm |
| | > 5.0 N | typ. 0.6 mm |
| 25 | Radial play | preloaded |
| 26 | Max. axial load (dynamic) | 4.8 N |
| 27 | Max. force for press fits (static) | 45 N |
| | (static, shaft supported) | 1000 N |
| 28 | Max. radial load, 5 mm from flange | 14 N |
| | **Other specifications** | |
| 29 | Number of pole pairs | 4 |
| 30 | Number of phases | 3 |
| 31 | Weight of motor | 46 g |

Values listed in the table are nominal.

| **Connection** | **with Hall sensors** | **sensorless** |
|---|---|---|
| Pin 1 | $V_{Hall}$ 3.5…24 VDC | Motor winding 1 |
| Pin 2 | Hall sensor 3 | Motor winding 2 |
| Pin 3 | Hall sensor 1 | Motor winding 3 |
| Pin 4 | Hall sensor 2 | ⅄ neutral point |
| Pin 5 | GND | |
| Pin 6 | Motor winding 3 | |
| Pin 7 | Motor winding 2 | |
| Pin 8 | Motor winding 1 | |
| **Adapter** | **Part number** | **Part number** |
| see p. 398 | 220300 | 220310 |
| **Connector** | **Part number** | **Part number** |
| Tyco | 1-84953-1 | 84953-4 |
| Molex | 52207-1133 | 52207-0433 |
| Molex | 52089-1119 | 52089-0419 |

Pin for design with Hall sensors:
FPC, 11-pol, Pitch 1.0 mm, top contact style
Wiring diagram for Hall sensors see p. 35

## Operating Range



## Comments

🟥 **Continuous operation**
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.
= Thermal limit.

□ **Short term operation**
The motor may be briefly overloaded (recurring).

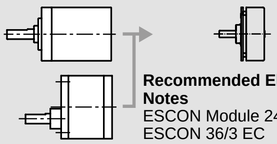
— **Assigned power rating**

## maxon Modular System

Overview on page 20–25

**Planetary Gearhead**
∅32 mm
0.75 - 6 Nm
Page 305–308

**Spur Gearhead**
∅38 mm
0.1 - 0.6 Nm
Page 313



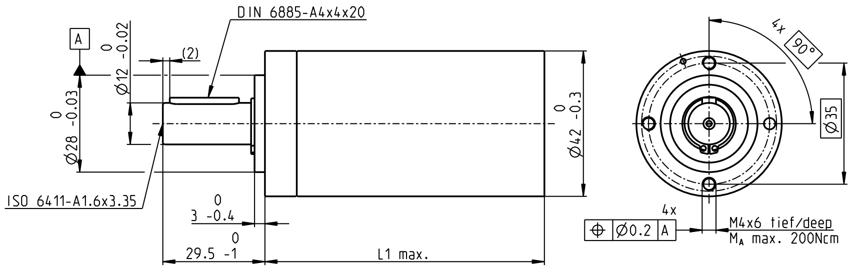
**Recommended Electronics:**

| **Notes** | Page **24** |
|---|---|
| ESCON Module 24/2 | 378 |
| ESCON 36/3 EC | 379 |
| ESCON Mod. 50/4 EC-S | 379 |
| ESCON Module 50/5 | 379 |
| ESCON 50/5 | 380 |
| DEC Module 24/2 | 382 |
| DEC Module 50/5 | 382 |
| EPOS2 24/2, 50/5 | 386 |
| EPOS2 Module 36/2 | 386 |
| EPOS3 70/10 EtherCAT | 393 |
| MAXPOS 50/5 | 396 |

1x203127

# Planetary Gearhead GP 42 C ∅42 mm, 3–15 Nm

Ceramic Version



**M 1:2**

## Technical Data

| | | |
|---|---|---|
| Planetary Gearhead | | straight teeth |
| Output shaft | | stainless steel |
| Bearing at output | | preloaded ball bearings |
| Radial play, 12 mm from flange | | max. 0.06 mm |
| Axial play at axial load | < 5 N | 0 mm |
| | > 5 N | max. 0.3 mm |
| Max. axial load (dynamic) | | 150 N |
| Max. force for press fits | | 300 N |
| Direction of rotation, drive to output | | = |
| Max. continuous input speed | | 8000 rpm |
| Recommended temperature range | | -40…+100°C |
| Number of stages | | 1 2 3 4 |
| Max. radial load, 12 mm from flange | | 120 N 240 N 360 N 360 N |

🟥 Stock program

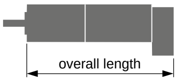
□ Standard program

▨ Special program (on request)

| | | | Part Numbers | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 203113 | 203115 | 203119 | 203120 | 203124 | 203129 | 203128 | 203133 | 203137 | 203141 |
| **Gearhead Data** | | | | | | | | | | | | |
| 1 | Reduction | | 3.5 : 1 | 12 : 1 | 26 : 1 | 43 : 1 | 81 : 1 | 156 : 1 | 150 : 1 | 285 : 1 | 441 : 1 | 756 : 1 |
| 2 | Absolute reduction | | $^{7}/_{2}$ | $^{49}/_{4}$ | 26 | $^{343}/_{8}$ | $^{2197}/_{27}$ | 156 | $^{2401}/_{16}$ | $^{15379}/_{54}$ | 441 | 756 |
| 10 | Mass inertia | gcm² | 14 | 15 | 9.1 | 15 | 9.4 | 9.1 | 15 | 15 | 14 | 14 |
| 3 | Max. motor shaft diameter | mm | 10 | 10 | 8 | 10 | 8 | 8 | 10 | 10 | 10 | 10 |
| | **Part Numbers** | | 203114 | 203116 | 260552* | 203121 | 203125 | 260553* | 203130 | 203134 | 203138 | 203142 |
| 1 | Reduction | | 4.3 : 1 | 15 : 1 | 36 : 1 | 53 : 1 | 91 : 1 | 216 : 1 | 186 : 1 | 319 : 1 | 488 : 1 | 936 : 1 |
| 2 | Absolute reduction | | $^{13}/_{3}$ | $^{91}/_{6}$ | $^{36}/_{1}$ | $^{637}/_{12}$ | 91 | $^{216}/_{1}$ | $^{4459}/_{24}$ | $^{637}/_{2}$ | $^{4394}/_{9}$ | 936 |
| 10 | Mass inertia | gcm² | 9.1 | 15 | 5.0 | 15 | 15 | 5.0 | 15 | 15 | 9.4 | 9.1 |
| 3 | Max. motor shaft diameter | mm | 8 | 10 | 4 | 10 | 10 | 4 | 10 | 10 | 8 | 8 |
| | **Part Numbers** | | 260551* | 203117 | | 203122 | 203126 | | 203131 | 203135 | 203139 | 260554* |
| 1 | Reduction | | 6 : 1 | 19 : 1 | | 66 : 1 | 113 : 1 | | 230 : 1 | 353 :1 | 546 : 1 | 1296 : 1 |
| 2 | Absolute reduction | | $^{6}/_{1}$ | $^{169}/_{9}$ | | $^{1183}/_{18}$ | $^{338}/_{3}$ | | $^{8281}/_{36}$ | $^{28561}/_{81}$ | 546 | $^{1296}/_{1}$ |
| 10 | Mass inertia | gcm² | 4.9 | 9.4 | | 15 | 9.4 | | 15 | 9.4 | 14 | 5.0 |
| 3 | Max. motor shaft diameter | mm | 4 | 8 | | 10 | 8 | | 10 | 8 | 10 | 4 |
| | **Part Numbers** | | | 203118 | | 203123 | 203127 | | 203132 | 203136 | 203140 | |
| 1 | Reduction | | | 21 : 1 | | 74 : 1 | 126 : 1 | | 257 : 1 | 394 : 1 | 676 : 1 | |
| 2 | Absolute reduction | | | 21 | | $^{147}/_{2}$ | 126 | | $^{1029}/_{4}$ | $^{1183}/_{3}$ | 676 | |
| 10 | Mass inertia | gcm² | | 14 | | 15 | 14 | | 15 | 15 | 9.1 | |
| 3 | Max. motor shaft diameter | mm | | 10 | | 10 | 10 | | 10 | 10 | 8 | |
| 4 | Number of stages | | 1 | 2 | 2 | 3 | 3 | 3 | 4 | 4 | 4 | 4 |
| 5 | Max. continuous torque | Nm | 3.0 | 7.5 | 7.5 | 15.0 | 15.0 | 15.0 | 15.0 | 15.0 | 15.0 | 15.0 |
| 6 | Max. intermittent torque at gear output | Nm | 4.5 | 11.3 | 11.3 | 22.5 | 22.5 | 22.5 | 22.5 | 22.5 | 22.5 | 22.5 |
| 7 | Max. efficiency | % | 90 | 81 | 81 | 72 | 72 | 72 | 64 | 64 | 64 | 64 |
| 8 | Weight | g | 260 | 360 | 360 | 460 | 460 | 460 | 560 | 560 | 560 | 560 |
| 9 | Average backlash no load | ° | 0.6 | 0.8 | 0.8 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| 11 | Gearhead length L1 | mm | 41.0 | 55.5 | 55.5 | 70.0 | 70.0 | 70.0 | 84.5 | 84.5 | 84.5 | 84.5 |

*no combination with EC 45 (150 W and 250 W)

overall length

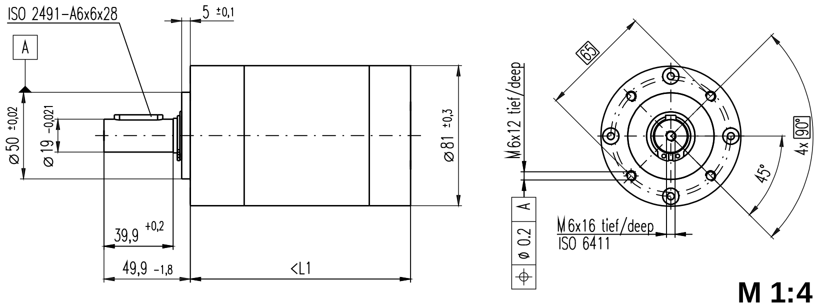

overall length

## maxon Modular System

| + Motor | Page | + Sensor | Page | Brake | Page | Overall length [mm] = Motor length + gearhead length + (sensor/brake) + assembly parts | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RE 35, 90 W | 140 | | | | | 112.1 | 126.6 | 126.6 | 141.1 | 141.1 | 141.1 | 155.6 | 155.6 | 155.6 | 155.6 |
| RE 35, 90 W | 140 | MR | 356 | | | 123.5 | 138.0 | 138.0 | 152.5 | 152.5 | 152.5 | 167.0 | 167.0 | 167.0 | 167.0 |
| RE 35, 90 W | 140 | HED_ 5540 | 362/364 | | | 132.8 | 147.3 | 147.3 | 161.8 | 161.8 | 161.8 | 176.3 | 176.3 | 176.3 | 176.3 |
| RE 35, 90 W | 140 | DCT 22 | 373 | | | 130.2 | 144.7 | 144.7 | 159.2 | 159.2 | 159.2 | 173.7 | 173.7 | 173.7 | 173.7 |
| RE 35, 90 W | 140 | | | AB 28 | 408 | 148.2 | 162.7 | 162.7 | 177.2 | 177.2 | 177.2 | 191.7 | 191.7 | 191.7 | 191.7 |
| RE 35, 90 W | 140 | HED_ 5540 | 362/364 | AB 28 | 408 | 165.4 | 179.9 | 179.9 | 194.4 | 194.4 | 194.4 | 208.9 | 208.9 | 208.9 | 208.9 |
| RE 40, 150 W | 142 | | | | | 112.1 | 126.6 | 126.6 | 141.1 | 141.1 | 141.1 | 155.6 | 155.6 | 155.6 | 155.6 |
| RE 40, 150 W | 142 | MR | 356 | | | 123.5 | 138.0 | 138.0 | 152.5 | 152.5 | 152.5 | 167.0 | 167.0 | 167.0 | 167.0 |
| RE 40, 150 W | 142 | HED_ 5540 | 362/365 | | | 132.8 | 147.3 | 147.3 | 161.8 | 161.8 | 161.8 | 176.3 | 176.3 | 176.3 | 176.3 |
| RE 40, 150 W | 142 | HEDL 9140 | 368 | | | 166.2 | 180.7 | 180.7 | 195.2 | 195.2 | 195.2 | 209.7 | 209.7 | 209.7 | 209.7 |
| RE 40, 150 W | 142 | | | AB 28 | 408 | 148.2 | 162.7 | 162.7 | 177.2 | 177.2 | 177.2 | 191.7 | 191.7 | 191.7 | 191.7 |
| RE 40, 150 W | 142 | | | AB 28 | 409 | 156.2 | 170.7 | 170.7 | 185.2 | 185.2 | 185.2 | 199.7 | 199.7 | 199.7 | 199.7 |
| RE 40, 150 W | 142 | HED_ 5540 | 362/365 | AB 28 | 408 | 165.4 | 179.9 | 179.9 | 194.4 | 194.4 | 194.4 | 208.9 | 208.9 | 208.9 | 208.9 |
| RE 40, 150 W | 142 | HEDL 9140 | 368 | AB 28 | 409 | 176.7 | 191.2 | 191.2 | 205.7 | 205.7 | 205.7 | 220.2 | 220.2 | 220.2 | 220.2 |
| EC 40, 170 W | 215 | | | | | 121.1 | 135.6 | 135.6 | 150.1 | 150.1 | 150.1 | 164.6 | 164.6 | 164.6 | 164.6 |
| EC 40, 170 W | 215 | HED_ 5540 | 363/366 | | | 144.5 | 159.0 | 159.0 | 175.5 | 175.5 | 175.5 | 188.0 | 188.0 | 188.0 | 188.0 |
| EC 40, 170 W | 215 | Res 26 | 374 | | | 148.3 | 162.8 | 162.8 | 177.3 | 177.3 | 177.3 | 191.8 | 191.8 | 191.8 | 191.8 |
| EC 40, 170 W | 215 | | | AB 32 | 410 | 163.8 | 178.3 | 178.3 | 192.8 | 192.8 | 192.8 | 207.3 | 207.3 | 207.3 | 207.3 |
| EC 40, 170 W | 215 | HED_ 5540 | 363/366 | AB 32 | 410 | 182.2 | 196.7 | 196.7 | 211.2 | 211.2 | 211.2 | 225.7 | 225.7 | 225.7 | 225.7 |
| EC 45, 150 W | 216 | | | | | 152.3 | 166.8 | 166.8 | 181.3 | 181.3 | 181.3 | 195.8 | 195.8 | 195.8 | 195.8 |
| EC 45, 150 W | 216 | HEDL 9140 | 368 | | | 167.9 | 182.4 | 182.4 | 196.9 | 196.9 | 196.9 | 211.4 | 211.4 | 211.4 | 211.4 |
| EC 45, 150 W | 216 | Res 26 | 374 | | | 152.3 | 166.8 | 166.8 | 181.3 | 181.3 | 181.3 | 195.8 | 195.8 | 195.8 | 195.8 |
| EC 45, 150 W | 216 | | | AB 28 | 409 | 159.7 | 174.2 | 174.2 | 188.7 | 188.7 | 188.7 | 203.2 | 203.2 | 203.2 | 203.2 |
| EC 45, 150 W | 216 | HEDL 9140 | 368 | AB 28 | 409 | 176.7 | 191.2 | 191.2 | 205.7 | 205.7 | 205.7 | 220.2 | 220.2 | 220.2 | 220.2 |
| EC 45, 250 W | 217 | | | | | 185.1 | 199.6 | 199.6 | 214.1 | 214.1 | 214.1 | 228.6 | 228.6 | 228.6 | 228.6 |
| EC 45, 250 W | 217 | HEDL 9140 | 368 | | | 200.7 | 215.2 | 215.2 | 229.7 | 229.7 | 229.7 | 244.2 | 244.2 | 244.2 | 244.2 |
| EC 45, 250 W | 217 | Res 26 | 374 | | | 185.1 | 199.6 | 199.6 | 214.1 | 214.1 | 214.1 | 228.6 | 228.6 | 228.6 | 228.6 |
| EC 45, 250 W | 217 | | | AB 28 | 409 | 192.5 | 207.0 | 207.0 | 221.5 | 221.5 | 221.5 | 236.0 | 236.0 | 236.0 | 236.0 |
| EC 45, 250 W | 217 | HEDL 9140 | 368 | AB 28 | 409 | 209.5 | 224.0 | 224.0 | 238.5 | 238.5 | 238.5 | 253.0 | 253.0 | 253.0 | 253.0 |

# 1x110413

# Planetary Gearhead GP 81 A ⌀81 mm, 20–120 Nm



M 1:4

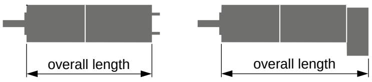
## Technical Data

| | | | |
|---|---|---|---|
| Planetary Gearhead | | | straight teeth |
| Output shaft | | | steel |
| Bearing at output | | | ball bearing |
| Radial play, 8 mm from flange | | | max. 0.1 mm |
| Axial play | | | max. 1 mm |
| Max. force for press fits | | | 1500 N |
| Direction of rotation, drive to output | | | = |
| Max. continuous input speed | | | 3000 rpm |
| Recommended temperature range | | | -30...+140°C |
| Number of stages | 1 | 2 | 3 |
| Max. radial load, 24 mm from flange | 400 N | 600 N | 1000 N |
| Max. axial load (dynamic) | 80 N | 120 N | 200 N |

🟥 Stock program
□ Standard program
▨ Special program (on request)

## Part Numbers

| | Gearhead Data | | 110408 | 110409 | 110410 | 110411 | 110412 | 110413 |
|---|---|---|---|---|---|---|---|---|
| 1 | Reduction | | 3.7 : 1 | 14 : 1 | 25 : 1 | 51 : 1 | 93 : 1 | 308 : 1 |
| 2 | Absolute reduction | | 63/17 | 3969/289 | 1701/68 | 250047/4913 | 107163/1156 | 19683/64 |
| 3 | Max. motor shaft diameter | mm | 14 | 14 | 14 | 14 | 14 | 14 |
| 4 | Number of stages | | 1 | 2 | 2 | 3 | 3 | 3 |
| 5 | Max. continuous torque | Nm | 20 | 60 | 60 | 120 | 120 | 120 |
| 6 | Max. intermittent torque at gear output | Nm | 30 | 90 | 90 | 180 | 180 | 180 |
| 7 | Max. efficiency | % | 80 | 75 | 75 | 70 | 70 | 70 |
| 8 | Weight | g | 2300 | 3000 | 3000 | 3700 | 3700 | 3700 |
| 9 | Average backlash no load | ° | 0.5 | 0.55 | 0.55 | 0.6 | 0.6 | 0.6 |
| 10 | Mass inertia | gcm² | 165 | 155 | 125 | 88 | 154 | 89 |
| 11 | Gearhead length L1 | mm | 92.0 | 113.7 | 113.7 | 135.3 | 135.3 | 135.3 |

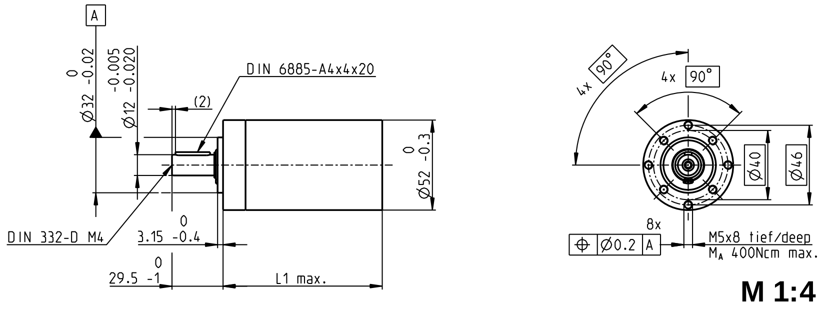


## maxon Modular System

| + Motor | Page | + Sensor | Page | Brake | Page | Overall length [mm] = Motor length + gearhead length + (sensor/brake) + assembly parts | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| RE 65, 250 W | 144 | | | | | 223.5 | 245.2 | 245.2 | 266.8 | 266.8 | 266.8 |
| RE 65, 250 W | 144 | HEDS 5540 | 363 | | | 249.4 | 271.1 | 271.1 | 292.7 | 292.7 | 292.7 |
| RE 65, 250 W | 144 | HEDL 5540 | 365 | | | 249.4 | 271.1 | 271.1 | 292.7 | 292.7 | 292.7 |
| RE 65, 250 W | 144 | HEDL 9140 | 369 | | | 279.6 | 301.3 | 301.3 | 322.9 | 322.9 | 322.9 |
| RE 65, 250 W | 144 | | | AB 44 | 412 | 279.6 | 301.3 | 301.3 | 322.9 | 322.9 | 322.9 |
| RE 65, 250 W | 144 | HEDL 9140 | 369 | AB 44 | 412 | 297.6 | 319.3 | 319.3 | 340.9 | 340.9 | 340.9 |
| EC 60, 400 W | 218 | | | | | 269.4 | 291.1 | 291.1 | 312.7 | 312.7 | 312.7 |
| EC 60, 400 W | 218 | HEDL 9140 | 368 | | | 269.4 | 291.1 | 291.1 | 312.7 | 312.7 | 312.7 |
| EC 60, 400 W | 218 | Res 26 | 374 | | | 269.4 | 291.1 | 291.1 | 312.7 | 312.7 | 312.7 |
| EC 60, 400 W | 218 | | | AB 41 | 411 | 283.0 | 304.7 | 304.7 | 326.3 | 326.3 | 326.3 |
| EC 60, 400 W | 218 | HEDL 9140 | 368 | AB 41 | 411 | 307.0 | 328.7 | 328.7 | 350.3 | 350.3 | 350.3 |

# Planetary Gearhead GP 52 C ∅52 mm, 4–30 Nm

Ceramic Version

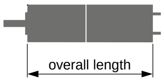


## Technical Data

| | | | | | |
|---|---|---|---|---|---|
| Planetary Gearhead | | | | | straight teeth |
| Output shaft | | | | | stainless steel |
| Bearing at output | | | | | preloaded ball bearings |
| Radial play, 12 mm from flange | | | | | max. 0.06 mm |
| Axial play at axial load | < 5 N | | | | 0 mm |
| | > 5 N | | | | max. 0.3 mm |
| Max. axial load (dynamic) | | | | | 200 N |
| Max. force for press fits | | | | | 500 N |
| Direction of rotation, drive to output | | | | | = |
| Max. continuous input speed | | | | | 6000 rpm |
| Recommended temperature range | | | | | -15…+80°C |
| Extended range as option | | | | | -40…+100°C |
| Number of stages | 1 | 2 | 3 | | 4 |
| Max. radial load, 12 mm from flange | 420 N | 630 N | 900 N | | 900 N |

🟥 Stock program
□ Standard program
▨ Special program (on request)

| | | | Part Numbers | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | 223080 | 223083 | 223089 | 223094 | 223097 | 223104 | 223109 |
| **Gearhead Data** | | | | | | | | | |
| 1 | Reduction | | 3.5 : 1 | 12 : 1 | 43 : 1 | 91 : 1 | 150 : 1 | 319 : 1 | 546 : 1 |
| 2 | Absolute reduction | | $^{7}/_{2}$ | $^{49}/_{4}$ | $^{343}/_{8}$ | 91 | $^{2401}/_{16}$ | $^{637}/_{2}$ | 546 |
| 10 | Mass inertia | gcm$^2$ | 20.7 | 17.6 | 17.3 | 16.7 | 17.3 | 16.8 | 16.4 |
| 3 | Max. motor shaft diameter | mm | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| | **Part Numbers** | | 223081 | 223084 | 223090 | 223095 | 223099 | 223105 | 223110 |
| 1 | Reduction | | 4.3 : 1 | 15 : 1 | 53 : 1 | 113 : 1 | 186 : 1 | 353 :1 | 676 : 1 |
| 2 | Absolute reduction | | $^{13}/_{3}$ | $^{91}/_{6}$ | $^{637}/_{12}$ | $^{338}/_{3}$ | $^{4459}/_{24}$ | $^{28561}/_{81}$ | 676 |
| 10 | Mass inertia | gcm$^2$ | 12 | 16.8 | 17.2 | 9.3 | 17.3 | 9.4 | 9.1 |
| 3 | Max. motor shaft diameter | mm | 8 | 10 | 10 | 8 | 10 | 8 | 8 |
| | **Part Numbers** | | | 223085 | 223091 | 223096 | 223101 | 223106 | 223111 |
| 1 | Reduction | | | 19 : 1 | 66 : 1 | 126 : 1 | 230 : 1 | 394 : 1 | 756 : 1 |
| 2 | Absolute reduction | | | $^{169}/_{9}$ | $^{1183}/_{18}$ | 126 | $^{8281}/_{36}$ | $^{1183}/_{3}$ | 756 |
| 10 | Mass inertia | gcm$^2$ | | 9.5 | 16.7 | 16.4 | 16.8 | 16.7 | 16.4 |
| 3 | Max. motor shaft diameter | mm | | 8 | 10 | 10 | 10 | 10 | 10 |
| | **Part Numbers** | | | 223086 | 223092 | 223098 | 223102 | 223107 | 223112 |
| 1 | Reduction | | | 21 : 1 | 74 : 1 | 156 : 1 | 257 : 1 | 441 : 1 | 936 : 1 |
| 2 | Absolute reduction | | | 21 | $^{147}/_{2}$ | 156 | $^{1029}/_{4}$ | 441 | 936 |
| 10 | Mass inertia | gcm$^2$ | | 16.5 | 17.2 | 9.1 | 17.3 | 16.5 | 9.1 |
| 3 | Max. motor shaft diameter | mm | | 10 | 10 | 8 | 10 | 10 | 8 |
| | **Part Numbers** | | | 223087 | 223093 | | 223103 | 223108 | |
| 1 | Reduction | | | 26 : 1 | 81 : 1 | | 285 : 1 | 488 : 1 | |
| 2 | Absolute reduction | | | 26 | 2197⁄27 | | 15379⁄54 | 4394⁄9 | |
| 10 | Mass inertia | gcm$^2$ | | 9.1 | 9.4 | | 16.7 | 9.4 | |
| 3 | Max. motor shaft diameter | mm | | 8 | 8 | | 10 | 8 | |
| 4 | Number of stages | | 1 | 2 | 3 | 3 | 4 | 4 | 4 |
| 5 | Max. continuous torque | Nm | 4 | 15 | 30 | 30 | 30 | 30 | 30 |
| 6 | Max. intermittent torque at gear output | Nm | 6 | 22.5 | 45 | 45 | 45 | 45 | 45 |
| 7 | Max. efficiency | % | 91 | 83 | 75 | 75 | 68 | 68 | 68 |
| 8 | Weight | g | 460 | 620 | 770 | 770 | 920 | 920 | 920 |
| 9 | Average backlash no load | ° | 0.6 | 0.8 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| 11 | Gearhead length L1 | mm | 49.0 | 65.0 | 78.5 | 78.5 | 92.0 | 92.0 | 92.0 |

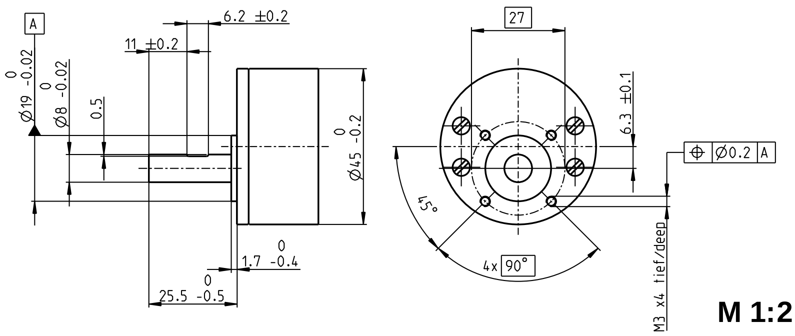


## maxon Modular System

| + Motor | Page | + Sensor | Page | Brake | Page | Overall length [mm] = Motor length + gearhead length + (sensor/brake) + assembly parts | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RE 40, 150 W | 142 | | | | | 120.1 | 136.1 | 149.6 | 149.6 | 163.1 | 163.1 | 163.1 |
| RE 40, 150 W | 142 | MR | 356 | | | 131.5 | 147.5 | 161.0 | 161.0 | 174.5 | 174.5 | 174.5 |
| RE 40, 150 W | 142 | HED_ 5540 | 362/365 | | | 140.8 | 156.8 | 170.3 | 170.3 | 183.8 | 183.8 | 183.8 |
| RE 40, 150 W | 142 | HEDL 9140 | 368 | | | 174.1 | 190.1 | 203.6 | 203.6 | 217.1 | 217.1 | 217.1 |
| RE 40, 150 W | 142 | | | AB 28 | 408 | 156.2 | 172.2 | 185.7 | 185.7 | 199.2 | 199.2 | 199.2 |
| RE 40, 150 W | 142 | | | AB 28 | 409 | 164.2 | 180.2 | 193.7 | 193.7 | 207.2 | 207.2 | 207.2 |
| RE 40, 150 W | 142 | HED_ 5540 | 362/365 | AB 28 | 408 | 173.4 | 189.4 | 202.9 | 202.9 | 216.4 | 216.4 | 216.4 |
| RE 40, 150 W | 142 | HEDL 9140 | 368 | AB 28 | 409 | 184.6 | 200.6 | 214.1 | 214.1 | 227.6 | 227.6 | 227.6 |
| RE 50, 200 W | 143 | | | | | 157.1 | 173.1 | 186.6 | 186.6 | 200.1 | 200.1 | 200.1 |
| RE 50, 200 W | 143 | HED_5540 | 363/365 | | | 177.8 | 193.8 | 207.3 | 207.3 | 220.8 | 220.8 | 220.8 |
| RE 50, 200 W | 143 | HEDL 9140 | 369 | | | 219.5 | 235.5 | 249.0 | 249.0 | 262.5 | 262.5 | 262.5 |
| RE 50, 200 W | 143 | | | AB 44 | 412 | 219.5 | 235.5 | 249.0 | 249.0 | 262.5 | 262.5 | 262.5 |
| RE 50, 200 W | 143 | HEDL 9140 | 369 | AB 44 | 412 | 232.5 | 248.5 | 262.0 | 262.0 | 275.5 | 275.5 | 275.5 |
| EC 40, 170 W | 215 | | | | | 129.1 | 145.1 | 158.6 | 158.6 | 172.1 | 172.1 | 172.1 |
| EC 40, 170 W | 215 | HED_5540 | 363/366 | | | 152.5 | 168.5 | 182.0 | 182.0 | 195.5 | 195.5 | 195.5 |
| EC 40, 170 W | 215 | Res 26 | 374 | | | 156.3 | 172.3 | 185.8 | 185.8 | 199.3 | 199.3 | 199.3 |
| EC 40, 170 W | 215 | | | AB 32 | 410 | 171.8 | 187.8 | 201.3 | 201.3 | 214.8 | 214.8 | 214.8 |
| EC 40, 170 W | 215 | HED_5540 | 363/366 | AB 32 | 410 | 190.2 | 206.2 | 219.7 | 219.7 | 233.2 | 233.2 | 233.2 |

# 2x301171

## Spur Gearhead GS 45 A ∅45 mm, 0.5–2.0 Nm



M 1:2

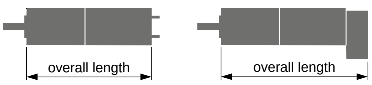
### Technical Data

| | |
|---|---|
| Spur Gearhead | straight teeth |
| Output shaft | stainless steel, hardened |
| Bearing at output | ball bearing |
| Radial play, 10 mm from flange | max. 0.15 mm |
| Axial play | 0.02–0.2 mm |
| Max. axial load (dynamic) | 60 N |
| Max. force for press fits | 60 N |
| Max. continuous input speed | 6000 rpm |
| Recommended temperature range | -15…+80°C |
| Number of stages | 2 3 4 5 6 |
| Max. radial load, 10 mm from flange | 120 N 180 N 190 N 190 N 190 N |

🟥 Stock program
□ Standard program
▨ Special program (on request)

| | Gearhead Data | | Part Numbers | | | | |
|---|---|---|---|---|---|---|---|
| | | | 301177 | 301175 | 301181 | 301186 | 301191 |
| 1 | Reduction | | 5 : 1 | 18 : 1 | 61 : 1 | 212 : 1 | 732 : 1 |
| 2 | Absolute reduction | | 51/10 | 459/26 | 20655/338 | 125862/595 | 492790/673 |
| 10 | Mass inertia | gcm² | 3.7 | 1.6 | 1.0 | 0.8 | 0.8 |
| 3 | Max. motor shaft diameter | mm | 3 | 3 | 3 | 3 | 3 |
| | **Part Numbers** | | 301178 | 301173 | 301182 | 301187 | 301192 |
| 1 | Reduction | | 7 : 1 | 26 : 1 | 89 : 1 | 310 : 1 | 1072 : 1 |
| 2 | Absolute reduction | | 209/28 | 9405/364 | 66632/745 | 183281/592 | 307572/287 |
| 10 | Mass inertia | gcm² | 3.1 | 1.4 | 1.0 | 0.8 | 0.8 |
| 3 | Max. motor shaft diameter | mm | 3 | 3 | 3 | 3 | 3 |
| | **Part Numbers** | | 301179 | 266595 | 301184 | 301188 | 301193 |
| 1 | Reduction | | 9 : 1 | 32 : 1 | 111 : 1 | 385 : 1 | 1334 : 1 |
| 2 | Absolute reduction | | 2295/247 | 8523/265 | 334/3 | 173808/451 | 198769/149 |
| 10 | Mass inertia | gcm² | 2.1 | 1.4 | 0.6 | 0.5 | 0.4 |
| 3 | Max. motor shaft diameter | mm | 3 | 3 | 3 | 3 | 3 |
| | **Part Numbers** | | 301180 | 301171 | 301185 | 301189 | 301194 |
| 1 | Reduction | | 14 : 1 | 47 : 1 | 163 : 1 | 564 : 1 | 1952 : 1 |
| 2 | Absolute reduction | | 2475/182 | 6221/132 | 141157/861 | 161880/287 | 1929023/988 |
| 10 | Mass inertia | gcm² | 2.2 | 0.9 | 0.5 | 0.5 | 0.4 |
| 3 | Max. motor shaft diameter | mm | 3 | 3 | 3 | 3 | 3 |
| 4 | Number of stages | | 2 | 3 | 4 | 5 | 6 |
| 5 | Max. continuous torque | Nm | 0.5 | 2.0 | 2.0 | 2.0 | 2.0 |
| 6 | Max. intermittent torque at gear output | Nm | 0.75 | 2.5 | 2.5 | 2.5 | 2.5 |
| 12 | Direction of rotation, drive to output | | = | ≠ | = | ≠ | = |
| 7 | Max. efficiency | % | 87 | 76 | 66 | 59 | 53 |
| 8 | Weight | g | 224 | 224 | 255 | 287 | 313 |
| 9 | Average backlash no load | ° | 1.6 | 2.0 | 2.4 | 2.8 | 3.2 |
| 11 | Gearhead length L1* | mm | 23.5 | 23.5 | 26.9 | 30.4 | 33.8 |

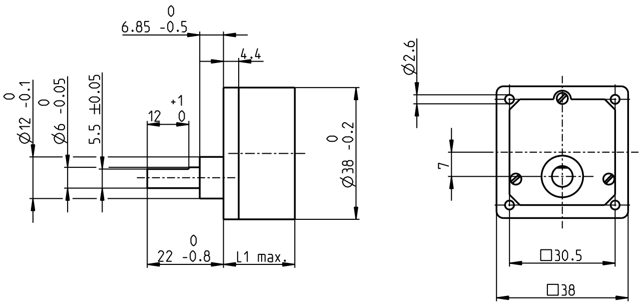
*for EC 45 flat, IE, L1 is max. + 4.0 mm



### maxon Modular System

| + Motor | Page | + Sensor/Brake | Page | Overall length [mm] = Motor length + gearhead length + (sensor/brake) + assembly parts | | | | |
|---|---|---|---|---|---|---|---|---|
| EC 45 flat, 30 W | 261 | | | 40.0 | 40.0 | 43.4 | 46.9 | 50.3 |
| EC 45 flat, 30 W | 261 | MILE | 342 | 43.0 | 43.0 | 46.4 | 49.9 | 53.3 |
| EC 45 flat, 50 W | 262 | | | 44.9 | 44.9 | 48.3 | 51.8 | 55.2 |
| EC 45 flat, 50 W | 262 | MILE | 342 | 46.2 | 46.2 | 49.6 | 53.1 | 56.5 |
| EC 45 flat, 70 W | 263 | | | 50.3 | 50.3 | 53.7 | 57.2 | 60.6 |
| EC 45 flat, 70 W | 263 | MILE | 342 | 52.0 | 52.0 | 55.4 | 58.9 | 62.3 |
| EC 45 flat, IE, IP 00 | 264 | | | 59.2 | 59.2 | 62.6 | 66.1 | 69.5 |
| EC 45 flat, IE, IP 40 | 264 | | | 61.4 | 61.4 | 64.8 | 68.3 | 71.7 |
| EC 45 flat, IE, IP 00 | 265 | | | 64.2 | 64.2 | 67.6 | 71.1 | 74.5 |
| EC 45 flat, IE, IP 40 | 265 | | | 66.4 | 66.4 | 69.8 | 73.3 | 76.7 |

# Spur Gearhead GS 38 A ∅38 mm, 0.1–0.6 Nm



M 1:2

## Technical Data

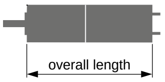
| | |
|---|---|
| Spur Gearhead | straight teeth |
| Output shaft | stainless steel |
| Bearing at output | sleeve bearing |
| Radial play, 12 mm from flange | max. 0.1 mm |
| Axial play | 0.03–0.2 mm |
| Max. axial load (dynamic) | 30 N |
| Max. force for press fits | 500 N |
| Max. continuous input speed | 5000 rpm |
| Recommended temperature range | -5…+80°C |
| Number of stages | 1 2 3 4 5 |
| Max. radial load, 12 mm from flange | 50 N 50 N 50 N 50 N 50 N |

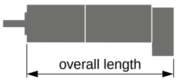
🟥 Stock program
□ Standard program
▨ Special program (on request)

## Part Numbers

| | | | 110451 | 110452 | 110453 | 110454 | 110455 | 110456 | 110457 | 110458 | 110459 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Gearhead Data** | | | | | | | | | | | |
| 1 | Reduction | | 6 : 1 | 10 : 1 | 18 : 1 | 30 : 1 | 60 : 1 | 100 : 1 | 200 : 1 | 500 : 1 | 900 : 1 |
| 2 | Absolute reduction | | 6 | 10 | 18 | 30 | 60 | 100 | 200 | 500 | 900 |
| 3 | Max. motor shaft diameter | mm | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 4 | Number of stages | | 2 | 2 | 3 | 3 | 4 | 4 | 5 | 6 | 6 |
| 5 | Max. continuous torque | Nm | 0.1 | 0.1 | 0.2 | 0.2 | 0.3 | 0.3 | 0.6 | 0.6 | 0.6 |
| 6 | Max. intermittent torque at gear output | Nm | 0.3 | 0.3 | 0.6 | 0.6 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 |
| 12 | Direction of rotation, drive to output | | = | = | ≠ | ≠ | = | = | ≠ | = | = |
| 7 | Max. efficiency | % | 81 | 81 | 73 | 73 | 66 | 66 | 59 | 53 | 53 |
| 8 | Weight | g | 55 | 55 | 60 | 60 | 65 | 65 | 70 | 75 | 75 |
| 9 | Average backlash no load | ° | 1.0 | 1.0 | 1.5 | 1.5 | 2.0 | 2.0 | 2.5 | 3.0 | 3.0 |
| 10 | Mass inertia | gcm$^2$ | 0.7 | 0.6 | 0.4 | 0.4 | 0.3 | 0.3 | 0.2 | 0.2 | 0.2 |
| 11 | Gearhead length L1* | mm | 20.6 | 20.6 | 23.1 | 23.1 | 25.6 | 25.6 | 28.1 | 30.6 | 30.6 |

*for EC 32 flat L1 is + 2.0 mm



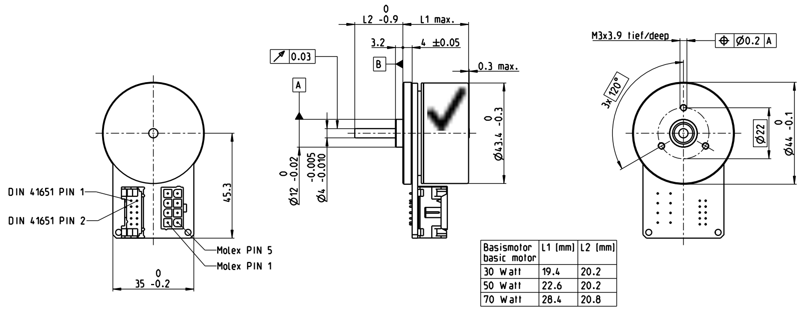


## maxon Modular System

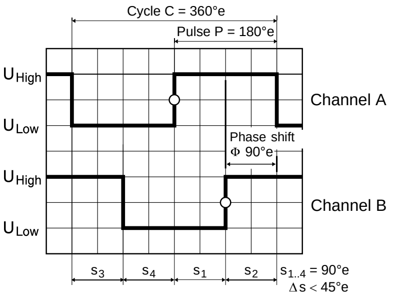
| + Motor | Page | + Sensor/Brake | Page | Overall length [mm] = Motor length + gearhead length + (sensor/brake) + assembly parts | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A-max 26 | 161-168 | | | 65.4 | 65.4 | 67.9 | 67.9 | 70.4 | 70.4 | 72.9 | 75.4 | 75.4 |
| A-max 26 | 162-168 | MEnc 13 | 372 | 72.5 | 72.5 | 75.0 | 75.0 | 77.5 | 77.5 | 80.0 | 82.5 | 82.5 |
| A-max 26 | 162-168 | MR | 355 | 74.2 | 74.2 | 76.7 | 76.7 | 79.2 | 79.2 | 81.7 | 84.2 | 84.2 |
| A-max 26 | 162-168 | Enc 22 | 361 | 79.8 | 79.8 | 82.3 | 82.3 | 84.8 | 84.8 | 87.3 | 89.8 | 89.8 |
| A-max 26 | 162-168 | HED_ 5540 | 363/365 | 83.8 | 83.8 | 86.3 | 86.3 | 88.8 | 88.8 | 91.3 | 93.8 | 93.8 |
| A-max 32 | 169/171 | | | 83.6 | 83.6 | 86.1 | 86.1 | 88.6 | 88.6 | 91.1 | 93.6 | 93.6 |
| A-max 32 | 170/172 | | | 82.2 | 82.2 | 84.7 | 84.7 | 87.2 | 87.2 | 89.7 | 92.2 | 92.2 |
| A-max 32 | 170/172 | MR | 356 | 93.4 | 93.4 | 95.9 | 95.9 | 98.4 | 98.4 | 100.9 | 103.4 | 103.4 |
| A-max 32 | 170/172 | HED_ 5540 | 363/365 | 103.0 | 103.0 | 105.5 | 105.5 | 108.0 | 108.0 | 110.5 | 113.0 | 113.0 |
| RE-max 21 | 179/180 | | | 49.6 | 49.6 | 52.1 | 52.1 | 54.6 | 54.6 | 57.1 | 59.6 | 59.6 |
| RE-max 21, 3.5 W | 180 | MR | 352/354 | 54.7 | 54.7 | 57.2 | 57.2 | 59.7 | 59.7 | 62.2 | 64.7 | 64.7 |
| RE-max 21 | 181/182 | | | 52.2 | 52.2 | 54.7 | 54.7 | 57.2 | 57.2 | 59.7 | 62.2 | 62.2 |
| RE-max 21, 6 W | 182 | MR | 352/354 | 56.5 | 56.5 | 59.0 | 59.0 | 61.5 | 61.5 | 64.0 | 66.5 | 66.5 |
| EC 32 flat, 15 W | 258 | | | 38.6 | 38.6 | 41.1 | 41.1 | 43.6 | 43.6 | 46.1 | 48.6 | 48.6 |
| EC 32 flat, IE, IP 00 | 259 | | | 48.7 | 48.7 | 51.2 | 51.2 | 53.7 | 53.7 | 56.2 | 58.7 | 58.7 |
| EC 32 flat, IE, IP 40 | 259 | | | 50.4 | 50.4 | 52.9 | 52.9 | 55.4 | 55.4 | 57.9 | 60.4 | 60.4 |

# Encoder MILE 256–2048 CPT, 2 Channels, with Line Driver

Integrated into motor



| Basismotor<br>basic motor | L1 [mm] | L2 [mm] |
|---|---|---|
| 30 Watt | 19.4 | 20.2 |
| 50 Watt | 22.6 | 20.2 |
| 70 Watt | 28.4 | 20.8 |

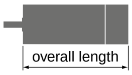


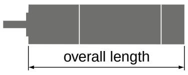
M 1:3

Direction of rotation cw (definition cw p. 106)

🟥 Stock program
□ Standard program
▨ Special program (on request)

| Article Numbers | 462002 | 462003 | 462004 | 462005 |
|---|---|---|---|---|
| **Type** | | | | |
| Counts per turn | 256 | 512 | 1024 | 2048 |
| Number of channels | 2 | 2 | 2 | 2 |
| Max. operating frequency (kHz) | 500 | 500 | 500 | 500 |
| Max. speed (rpm) | 10 000 | 10 000 | 10 000 | 10 000 |


overall length


overall length

## maxon Modular System

| + Motor | Page | + Gearhead | Page | + Brake | Page | Overall length [mm] / ● see Gearhead | | | |
|---|---|---|---|---|---|---|---|---|---|
| EC 45 flat, 30 W | 261 | | | | | 19.4 | 19.4 | 19.4 | 19.4 |
| EC 45 flat, 30 W | 261 | GP 42, 3 - 15 Nm | 316 | | | ● | ● | ● | ● |
| EC 45 flat, 30 W | 261 | GS 45, 0.5 - 2.0 Nm | 317 | | | ● | ● | ● | ● |
| EC 45 flat, 50 W | 262 | | | | | 22.6 | 22.6 | 22.6 | 22.6 |
| EC 45 flat, 50 W | 262 | GP 42, 3 - 15 Nm | 316 | | | ● | ● | ● | ● |
| EC 45 flat, 50 W | 262 | GS 45, 0.5 - 2.0 Nm | 317 | | | ● | ● | ● | ● |
| EC 45 flat, 70 W | 263 | | | | | 28.4 | 28.4 | 28.4 | 28.4 |
| EC 45 flat, 70 W | 263 | GP 42, 3 - 15 Nm | 316 | | | ● | ● | ● | ● |
| EC 45 flat, 70 W | 263 | GS 45, 0.5 - 2.0 Nm | 317 | | | ● | ● | ● | ● |

## Technical Data

| | |
|---|---|
| Supply voltage $V_{CC}$ | 5 V ± 10% |
| Output signal | CMOS and TTL compatible |
| State length $s_n$ 90°e (1000 rpm) | 45…135°e |
| Signal rise time<br>(typically, at $C_L$ = 25 pF, $R_L$ = 1 kΩ, 25 °C) | 100 ns |
| Signal fall time<br>(typically, at $C_L$ = 25 pF, $R_L$ = 1 kΩ, 25 °C) | 100 ns |
| Operating temperature range | -40…+100 °C |
| Moment of inertia of code wheel | ≤ 3.5 gcm$^2$ |
| Output current per channel | max. 4 mA |
| Open collector output of the Hall sensors<br>with integrated pull-up resistor | 10 kΩ ± 20% |
| Wiring diagram for Hall sensors see p. 35 | |

Additional information can be found under ‘Downloads’ in the maxon online shop.

## Pin Allocation

**Connection motor**

| | |
|---|---|
| Pin 1 | Hall sensor 1* |
| Pin 2 | Hall sensor 2* |
| Pin 3 | $V_{Hall}$ 4.5...18 VDC |
| Pin 4 | Motor winding 3 |
| Pin 5 | Hall sensor 3* |
| Pin 6 | GND |
| Pin 7 | Motor winding 1 |
| Pin 8 | Motor winding 2 |

*Internal pull-up (10 kΩ) on pin 3 ($V_{Hall}$)

**Connection Encoder**

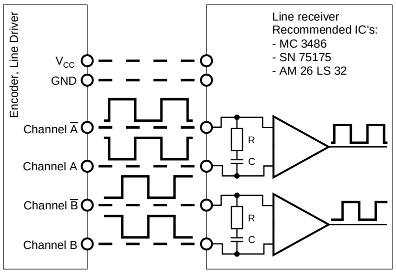
| | |
|---|---|
| Pin 1 | N.C. |
| Pin 2 | $V_{CC}$ |
| Pin 3 | GND |
| Pin 4 | N.C. |
| Pin 5 | Channel $\overline{A}$ |
| Pin 6 | Channel A |
| Pin 7 | Channel $\overline{B}$ |
| Pin 8 | Channel B |
| Pin 9 | Do not connect |
| Pin 10 | Do not connect |

**Connector:**

| | |
|---|---|
| 39-28-1083 | Molex |
| DIN 41651/EN 60603-13 | |

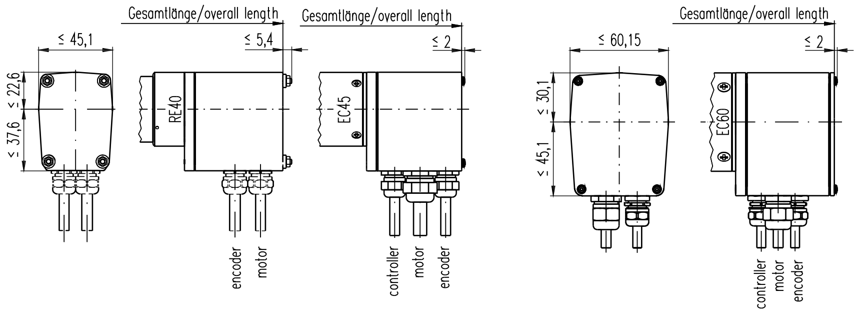
**Note:** Pull-down resistors < 100 kΩ on the encoder outputs are not permitted. Pull-up resistors are permitted, but not required.

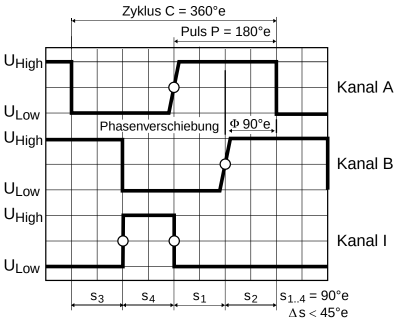
## Connection example



Opt. terminal resistance R = typical 120 Ω
Capacitor C ≥ 0.1 nF per m line length

# Encoder HEDL 9140 500 CPT, 3 Channels, with Line Driver RS 422



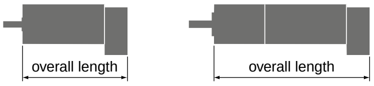


Direction of rotation cw (definition cw p. 106)

🟥 Stock program
□ Standard program
▨ Special program (on request)

| | Part Numbers |
|---|---|
| | 137959 |

| Type | |
|---|---|
| Counts per turn | 500 |
| Number of channels | 3 |
| Max. operating frequency (kHz) | 100 |
| Max. speed (rpm) | 12 000 |



## maxon Modular System

| + Motor | Page | + Gearhead | Page | + Brake | Page | Overall length [mm] / ● see Gearhead |
|---|---|---|---|---|---|---|
| RE 40, 150 W | 142 | | | | | 125.1 |
| RE 40, 150 W | 142 | GP 42, 3 - 15 Nm | 314 | | | ● |
| RE 40, 150 W | 142 | GP 52, 4 - 30 Nm | 318 | | | ● |
| RE 40, 150 W | 142 | | | AB 28 | 409 | 135.6 |
| RE 40, 150 W | 142 | GP 42, 3 - 15 Nm | 314 | AB 28 | 409 | ● |
| RE 40, 150 W | 142 | GP 52, 4 - 30 Nm | 318 | AB 28 | 409 | ● |
| EC 45, 150 W | 216 | | | | | 126.8 |
| EC 45, 150 W | 216 | GP 42, 3 - 15 Nm | 314 | | | ● |
| EC 45, 150 W | 216 | GP 52, 4 - 30 Nm | 318 | | | ● |
| EC 45, 150 W | 216 | | | AB 28 | 409 | 135.6 |
| EC 45, 150 W | 216 | GP 42, 3 - 15 Nm | 314 | AB 28 | 409 | ● |
| EC 45, 150 W | 216 | GP 52, 4 - 30 Nm | 318 | AB 28 | 409 | ● |
| EC 45, 250 W | 217 | | | | | 159.6 |
| EC 45, 250 W | 217 | GP 42, 3 - 15 Nm | 315 | | | ● |
| EC 45, 250 W | 217 | GP 52, 4 - 30 Nm | 318 | | | ● |
| EC 45, 250 W | 217 | GP 62, 8 - 50 Nm | 320 | | | ● |
| EC 45, 250 W | 217 | | | AB 28 | 409 | 168.4 |
| EC 45, 250 W | 217 | GP 42, 3 - 15 Nm | 315 | AB 28 | 409 | ● |
| EC 45, 250 W | 217 | GP 52, 4 - 30 Nm | 318 | AB 28 | 409 | ● |
| EC 45, 250 W | 217 | GP 62, 8 - 50 Nm | 320 | AB 28 | 409 | ● |
| EC 60, 400 W | 218 | | | | | 177.3 |
| EC 60, 400 W | 218 | GP 81, 20 - 120 Nm | 321 | | | ● |
| EC 60, 400 W | 218 | | | AB 41 | 411 | 214.9 |
| EC 60, 400 W | 218 | GP 81, 20 - 120 Nm | 321 | AB 41 | 411 | ● |

## Technical Data

| | |
|---|---|
| Supply voltage $V_{CC}$ | 5 V ± 10% |
| Output signal | EIA Standard RS 422 |
| driver used: | DS26LS31 |
| Phase shift Φ | 90°e ± 45°e |
| Signal rise time | |
| (typically, at $C_L$ = 25 pF, $R_L$ = 11 kΩ, 25 °C) | 180 ns |
| Signal fall time | |
| (typically, at $C_L$ = 25 pF, $R_L$ = 11 kΩ, 25 °C) | 40 ns |
| Index pulse width | 90°e |
| Operating temperature range | -40…+85 °C |
| Moment of inertia of code wheel | ≤ 0.6 gcm² |
| Max. angular acceleration | 250 000 rad s⁻² |
| Output current per channel | min. -20 mA, max. 20 mA |

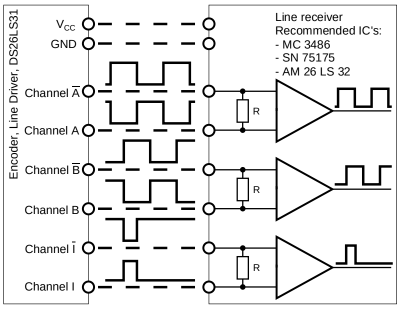
The index signal I is synchronized with channel A or B.

## Pin Allocation

| | | | |
|---|---|---|---|
| Cable white | = | 2 | $V_{CC}$ 5 VDC |
| Cable brown | = | 3 | GND |
| Cable green | = | 5 | Channel $\overline{A}$ |
| Cable yellow | = | 6 | Channel A |
| Cable grey | = | 7 | Channel $\overline{B}$ |
| Cable pink | = | 8 | Channel B |
| Cable blue | = | 9 | Channel $\overline{I}$ (Index) |
| Cable red | = | 10 | Channel I (Index) |

Cable size 8 × 0.25 mm²

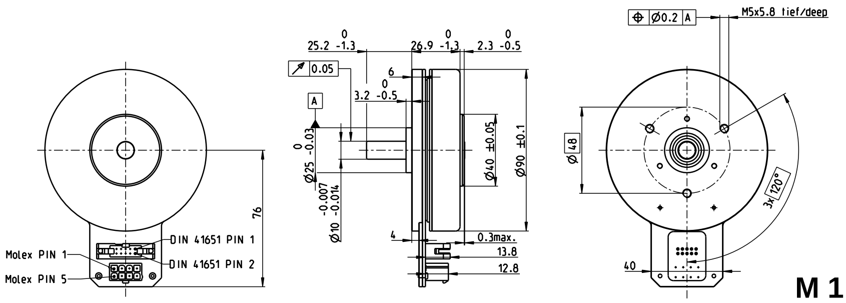
## Connection example



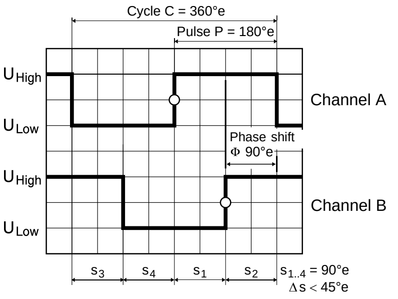
Terminal resistance R = typical 120 Ω

# **Encoder MILE** 512–6400 CPT, 2 Channels, with Line Driver RS 422

Integrated into motor



M 1:3

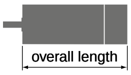


Direction of rotation cw (definition cw p. 106)

🟥 Stock program
□ Standard program
▒ Special program (on request)

## Part Numbers

| | 453234 | 409996 | 453233 | 411964 | 453232 | 411965 | 453231 | 411966 |
|---|---|---|---|---|---|---|---|---|
| **Type** | | | | | | | | |
| Counts per turn | 512 | 800 | 1024 | 1600 | 2048 | 3200 | 4096 | 6400 |
| Number of channels | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| Max. operating frequency (kHz) | 500 | 500 | 500 | 500 | 500 | 500 | 500 | 500 |
| Max. speed (rpm) | 5000 | 5000 | 5000 | 5000 | 5000 | 5000 | 5000 | 4650 |



## maxon Modular System

| + Motor | Page | + Gearhead | Page | + Brake | Page | Overall length [mm] / ● see Gearhead | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EC 90 flat | 267 | | | | | 29.2 | 29.2 | 29.2 | 29.2 | 29.2 | 29.2 | 29.2 | 29.2 |
| EC 90 flat | 267 | GP 52, 4 - 30 Nm | 319 | | | ● | ● | ● | ● | ● | ● | ● | ● |

## Technical Data

| | |
|---|---|
| Supply voltage $V_{CC}$ | 5 V ± 10% |
| Output signal | EIA Standard RS422 |
| driver used: | AM26C31QD |
| State length $s_n$ (500 rpm) | 90°e ± <45°e |
| Signal rise and fall times | |
| (typically, at $C_L$ = 120 pF, $R_L$ = 100 Ω) | 20 ns |
| Operating temperature range | -40…+100°C |
| Moment of inertia of code wheel | ≤ 65 gcm$^2$ |
| Output current per channel | min. -20 mA, max. 20 mA |
| Wiring diagram for Hall sensors see p. 35 | |

Additional information can be found under ‘Downloads’ in the maxon online shop.

## Pin Allocation

**Connection motor**

| | |
|---|---|
| Pin 1 | Hall sensor 1* |
| Pin 2 | Hall sensor 2* |
| Pin 3 | $V_{Hall}$ 4.5...18 VDC |
| Pin 4 | Motor winding 3 |
| Pin 5 | Hall sensor 3* |
| Pin 6 | GND |
| Pin 7 | Motor winding 1 |
| Pin 8 | Motor winding 2 |

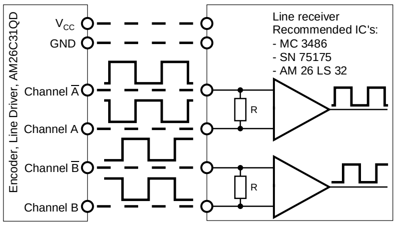
*Internal pull-up (10 kΩ) on pin 3 ($V_{Hall}$)

**Connection Encoder**

| | |
|---|---|
| Pin 1 | N.C. |
| Pin 2 | $V_{CC}$ |
| Pin 3 | GND |
| Pin 4 | N.C. |
| Pin 5 | Channel $\overline{A}$ |
| Pin 6 | Channel A |
| Pin 7 | Channel $\overline{B}$ |
| Pin 8 | Channel B |
| Pin 9 | Do not connect |
| Pin 10 | Do not connect |

**Connector:**
39-28-1083 Molex
DIN 41651/EN 60603-13

## Connection example



Opt. terminal resistance R = typical 120 Ω