Politecnico di Torino

Master's Degree in Mechatronic Engineering
A.a. 2024/2025
Sessione di Laurea Aprile 2025

Design and control of a four degrees of freedom manipulator for a mobile robot for automatic wine sampling in a winery.

Supervisors:
Prof.ssa Daniela Maffiodo
Prof. Giuseppe Quaglia
Ing. Andrea Botta
Ing. Daniel Pacheco Quinones
Ing. Luigi Tagliavini

Candidate:
Cesare Basso

Abstract

The thesis project falls within the field of collaborative robotics and aims to automate the wine sampling process through the use of a mobile robot. The project can be divided into three main areas: the mobile platform, the robotic arm, and the end effector. This work focuses on the design of a four-degree-of-freedom robotic arm that allows the end effector to reach the necessary positions to complete the wine extraction from the barrels. After a brief review of existing technologies in the field of robotic arms, the study begins with the dimensioning of an arm featuring a double-parallelogram mechanism, defining the relevant component lengths and establishing an operational space to be satisfied. The double-parallelogram configuration allows to passively maintain a constant orientation of the end-effector. The work then proceeds with the analysis of forward and inverse kinematics to ensure that the chosen dimensions can cover the entire operational space. The D-H convention is employed for the kinematics analysis, which concludes with a dexterity analysis throughout the entire workspace. The core of the thesis is the mechanical design phase, where the creation of the prototype is achieved through the development of custom components and the selection of commercial ones. This is followed by a theoretical discussion on balancing systems for robotic arms, which can be integrated into the project to reduce the load on the actuators. Two balancing systems are analyzed: a spring-and-pulley system and a cam-spring-pulley system. Both systems are examined and compared for the balancing of each joint. Finally, the prototype is simulated and tested through the design of a controller, defining its logic to ensure that the wine sampling process is validated under realistic conditions. Through simulation testing, it has been verified that the manipulator is correctly designed and dimensioned to reach the entire operational space. The project shall proceed with the 3D-printing of components for testing on the physical model, incorporating the discussed balancing systems. The control algorithm needs to be extended to manage the entire mobile robot, integrating control for both the end effector and the mobile platform, which, in this initial phase, are only simulated to validate the controller of the robotic arm.

Index

Index of Figures

Index of Tables

1 Introduction

Wine has ancient origins and is among the beverages with the richest history and cultural significance. With a global market valued at €245.6 billion in 2021, projections estimate that by 2025 the wine market will reach a value of €305.2 billion, of which €19 billion solely in Italy [1].

Wine is generally obtained through the fermentation of grapes, which, after a meticulous series of processes, transform from vine-grown fruit into an alcoholic beverage. The first step is grape cultivation. Depending on the grape variety and the desired final product, wineries select the proper terroir, timing, and the use of chemical products such as fertilizers and nutrients before bringing their labeled product to market. The harvest, typically occurring between August and October, marks the period when grapes are carefully selected and collected, starting the pressing stage, essentially the extraction of juice from the grapes. Traditionally, this was done by foot-stomping, whereas modern winemaking relies on mechanical presses. After pressing, the must is transferred into vats, followed by the destemming phase, which involves removing the seeds. Vinification is the fermentation process, during which sugars are converted into alcohol and carbon dioxide, typically lasting between 7 and 10 days. Once fermentation is complete and the wine has been carefully filtered, it undergoes aging and maturation. During this stage, the wine is transferred into steel, wooden, or concrete barrels, where the fermentation continues. The choice of barrel material significantly influences the final product. Finally, the wine is bottled, and for many red wines, further refinement continues in the bottle, enhancing its characteristics over time. [2]

From the vine to bottling, both the grapes and the resulting wine undergo rigorous laboratory analysis to assess all key characteristics. During the harvest, quality control ensures that the grapes have reached optimal maturity by measuring acidity, sugar levels, pH, malic acid, and nitrogen content. Phenolic compounds are also analyzed to evaluate the potential characteristics of the final wine. Once the wine is transferred into vats, fermentation is carefully monitored. In addition to the initial parameters, the conversion of sugars into alcohol is tracked to ensure a proper fermentation process. Biochemical markers related to yeast activity are also analyzed, providing insights into the fermentation's progress and overall success. [3]

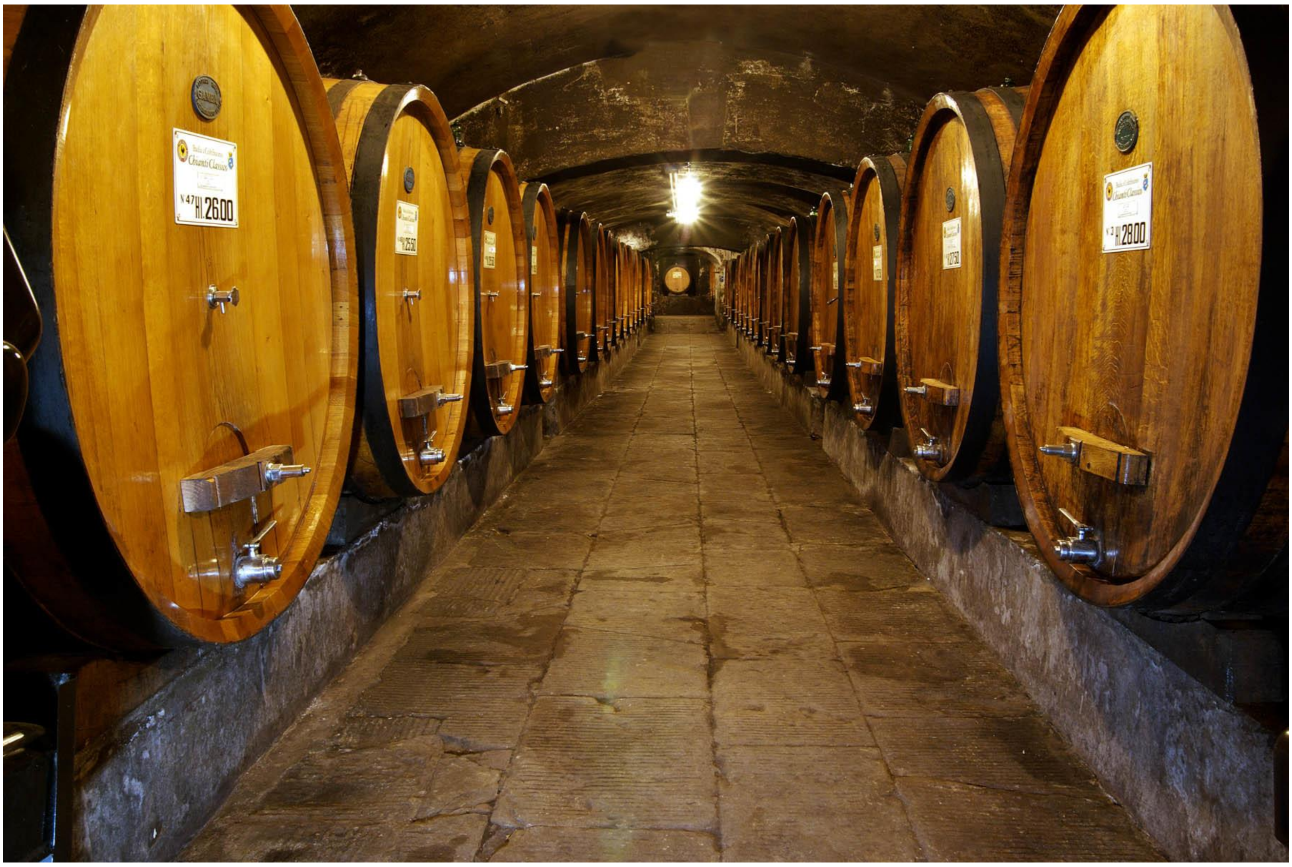


Figure 1 Barrels in a winery.

The aim of the thesis is to design a mobile robot that facilitates the sampling of wine from barrels during the winemaking process. This essential task can be automated by integrating a robotic arm onto a mobile platform capable of autonomously navigating the winery. The robot would be responsible for extracting wine samples from each barrel and transporting the test tubes to the laboratory for analysis. In the winery under consideration for the purpose of this thesis, the barrels are arranged in corridors approximately 2 meters wide, where the mobile platform must move autonomously, stopping in front of each barrel. The taps used for wine sampling are positioned at a height varying between 1 meter and 1.80 meters from the ground, which must be reached by the end effector.

The design process starts with an analysis of the state of the art of mobile manipulators. After the dimensional and kinematics analysis, the robotic arm is designed on SolidWorks and then simulated on Simulink.

Chapter 1

2 State of the art

2.1 Mobile Manipulators

The mechanical structure of a robotic manipulator consists of a sequence of *links*, interconnected by *joints*. The joints are moved by the actuators, which could be electric, hydraulic, pneumatic, etc. Other two characteristics of a manipulator are the *wrist,* which confers dexterity, and etc. *d-effector* that performs the required tasks. Most of the robotic manipulators may be classified by their [4]:

- *Structure:*
    - *Parallel manipulator*: consists in a mechanism in which the end-effector is supported by multiple serial chain.
    - *Serial manipulator*: is the most common industrial robot, designed as a series of chains links connected by motor-driven joints extended from the base to the end-effector.
- *Redundancy*: a redundant manipulator possesses more degrees of freedom than the task needed.
- *Mobility of the manipulator base: a mobile manipulator* is the result of the implementation of a manipulator on a mobile robot.
- *Rigidity of the links:*
    - *Elastic robotic manipulator* is a robotic systems with elastic links or flexible joints whose behavior is affected by parasitic elastic effects.
    - *Soft robotic manipulator:* which is inspired by boneless biological organisms.

Robotic systems known as mobile manipulators (MMs) have become a creative way to improve manufacturing flexibility and efficiency by combining the mobility of a mobile robot with the ability to manipulate objects through robotic arms, and have led to performing several tasks with high level of flexibility and accuracy [5]. Robotic manipulators are utilized in a variety of fields, such as space, agriculture, healthcare, manufacturing and military operations. In the industrial field they replace roles previously taken by humans, dealing with repetitive or hazardous tasks. The most observed applications are logistics for the 49%, and manufacturing for the 33%. [5]

As stated above, a mobile manipulator consists of one or more robotic arms placed on a mobile platform, that combined with sensors, actuators and tools, enabling the mobile robot to move autonomously and safely [5]. In the next lines, some project of mobile manipulators will be presented.

One of the first autonomous mobile manipulators was MORO (1994), that was capable of executing pick and place tasks. After MORO, the research has advanced with other projects in this field.

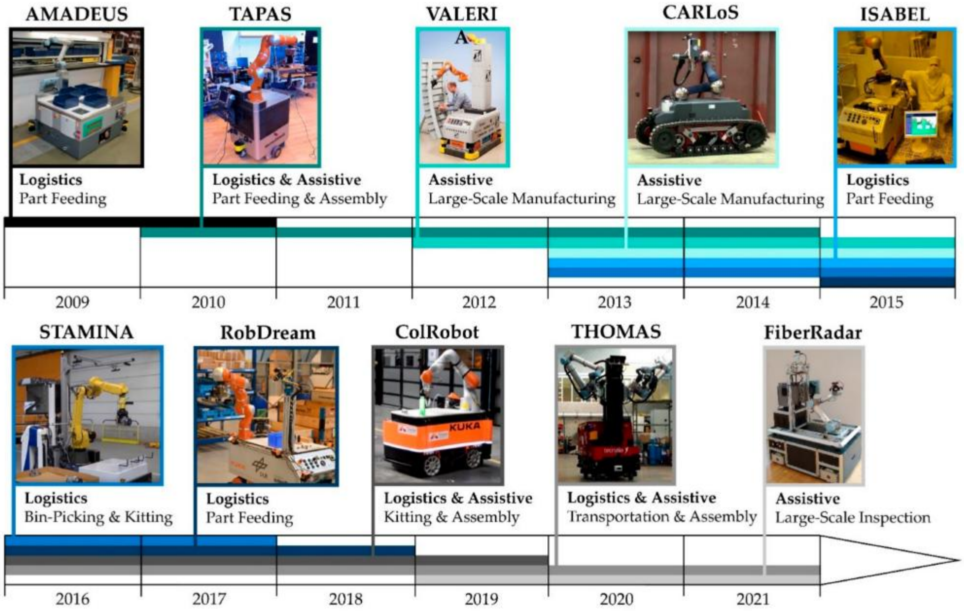


Figure 2 Related research projects from 2009 to 2021. [5]

In the TAPAS project, **Little Helpers** (LH) are a group of autonomous mobile manipulators developed by Aalborg University, that aim to produce flexible and efficient systems through a profound degree of automation. [5] The research project Little helper has been in work since 2009, and consists in seven generation of MMs:

- LH1 designed for logistic applications.
- LH2 designed for gesture based teaching.
- LH3 designed for assembly and machine tending.
- LH4 designed for hardware independence.
- LH5 designed for understanding human readable instructions.
- LH6 designed for robot co-worker.
- LH7 dual-arm robot co-worker

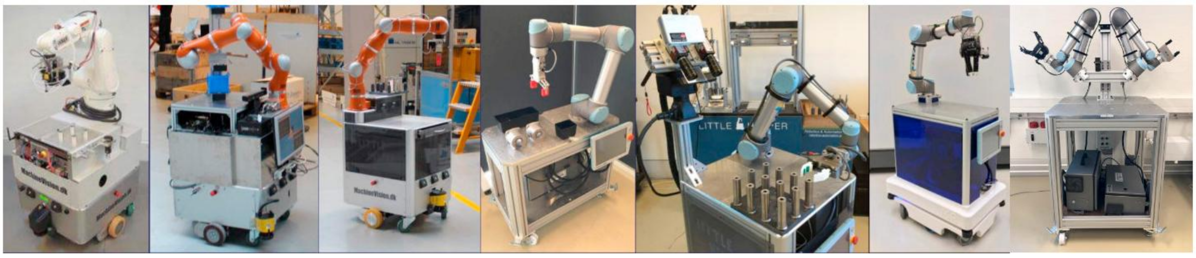


Figure 3 LH generations from one (left) to seven(right). [5]

The **STAMINA** mobile robot is designed for kitting process in the automotive industry: it groups and packs the items in a container, and then transports them in a specific car in a specific location [5]. It is composed of a six degrees of freedom manipulator mounted on an automatic guided vehicle and equipped with a 3 finger gripper and a series of sensors.

A recent mobile robot is **OMNIVIL** (2021), belonging to the project FiberRadar provided by IAAM (Institute for Applied Automation and Mechatronics). It has a self-designed mobile platform with holonomic kinematic for agility in dynamic and unstructured environments. It has three main advancements, one of which is a redundant workspace monitoring system that employs thermal and RGB cameras, lidar sensors, and deep convolutional neural networks to identify and categorize barriers and human employees. [6]

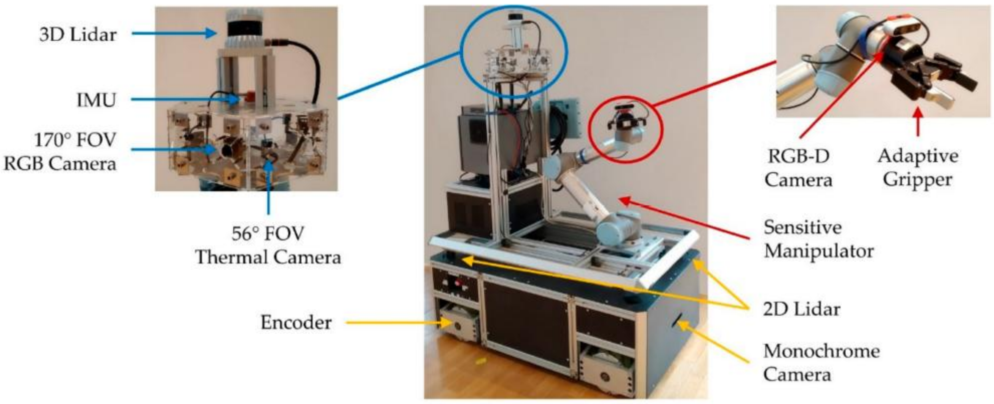


Figure 4 Mobile Manipulator OMNIVIL. [6]

Focusing on manipulators, RACHIE is a manipulator developed by a group of students from University of Minho, Portugal. Its primary goal is the detection and organization of cans, discriminating by colors and the presence of defects. The manipulator is characterized by 4 degrees of freedom: the torso θ_1 has 360° of freedom, the shoulder θ_2 90°, and up to 180° degrees of freedom for the elbow and the wrist. The implementation of a collapsible structure in aluminum and PLA proved to be a reliable and robust low-cost approach, and the robot can handle loads up to 150g. [7]

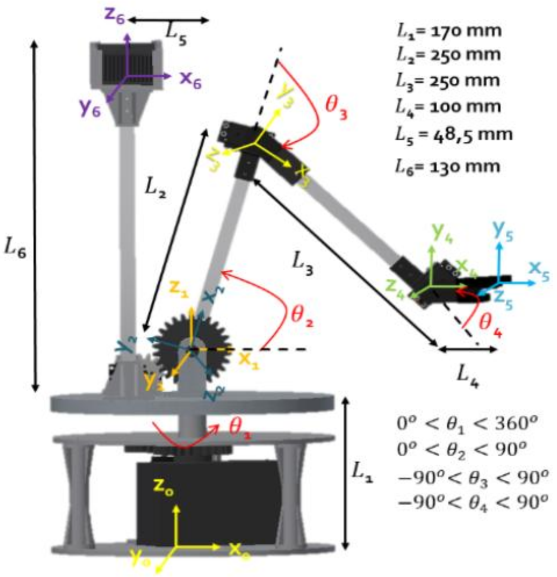


Figure 5 Coordinate axes of the manipulator's joint of RACHIE. [7]

In the paper "*Trajectory Planning with Obstacle Avoidance of 3 DoF Robotic Arm for Test Tube Handling System*" [8], is presented a prototype of 3-DoF parallel link robotic arm developed to demonstrate the process of autonomous test tube handling system for infectious materials in a table-top environment. Such manipulator is designed to perform tasks of picking and placing test tube from one place to another.

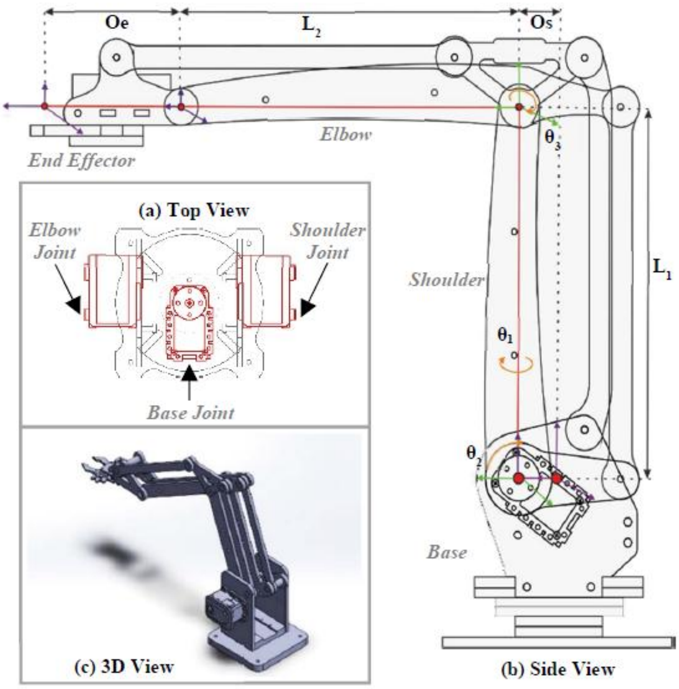


Figure 6 Robotic arm joint configuration from (a) top view, (b) side view, and (c) 3D view design of prototype. [8]

This robot is characterized by 3 joints configured as parallel links and it is equipped with a force sensing gripper which is attached to the end-effector. The robotic arm is developed using a parallel link mechanism, where all the motor servos are situated at the base of the arm. The term "parallel link" in this context refers to a link within the robot framework that runs parallel to the main link. This parallel link mechanism is intended to maintain the orientation of the pitch end effector of the robot in a horizontal position. Furthermore, this mechanism helps alleviate the torque load on the servo shoulder joint. Each joint has an operation range of 300° (0 to 300°). The paper presents a trajectory planning method with obstacle avoidance, and the testing phase shows an average position error of trajectory tracking is not more than 0.43 cm during the experiment with obstacle, and 0.35 cm without obstacle.

2.2 Balancing systems for robotic arms

Dynamic balancing involves force balancing, that can be realized by having the center of masses settled at a spot, and moment balancing, that can be realized by letting the angular momentum be constant. To achieve dynamic balancing, both force and moment balancing shall be fulfilled. Typically it is possible to achieve such balance by employing external components (counter-masses, springs, etc.), but this solution gets the system heavier, which means more energy is needed to drive the actuators.

In the paper "*A Review of Dynamic Balancing for Robotic Mechanisms* [9]" the authors show the following balancing methods. The dynamical balancing systems may be classified in *Balancing prior to kinematic synthesis* and *Balancing after the design.* In this chapter some of the systems of the second method will be presented.

- **Add separate counter-masses and counter-rotations**: adding counter-masses, the COM is relocated to a fixed point, leading to balance the shaking force, instead adding a counter-rotation leads to a shaking moment balance condition. It is a simple approach but the overall weight increases.
- **Add springs**: incorporate springs in the robotic system. With respect to the counter-masses and counter-rotations solution, adding springs results in a better solution for the weight of the system, but it is only applicable to a static balancing.
- **Add CRCM**: A CRCM (counter-rotary counter-mass) is a device to achieve both force balancing and moment balancing instead of using separate counter-masses and counter-moment, reducing extra masses and inertia. The CRCM can be an active driven (AD-CRCM), that results in a better solution with respect to the normal CRCM method.
- **Active dynamic balancing unit**: it is placed on the base of the system, and it produces same counter forces and counter moment to balance the shaking forces and moments of the systems.
- **Add secondary linkages.**
- **Balance through trajectory planning and control**: perform an optimal control planning to achieve a dynamic balancing condition.

In the article "*A Passive Weight Compensation Mechanism with a Non-Circular Pulley and a Spring*" [10] some applications of the passive methods are presented. In Figure 7 are shown typical examples of passive weigh compensation mechanism for a one degree of freedom inverted pendulum system.

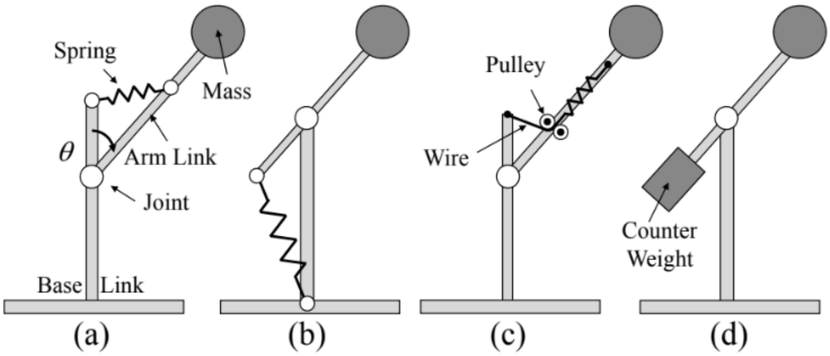


Figure 7 Conventional weight compensation systems. [10]

The solutions (a) and (b) use a spring and may be analyzed using the same kinematic model, however the example (b) becomes more popular due to a large workspace and small width of the robot. By using a an ordinary linear spring, it is not possible to perfectly compensate gravity torque within link's workspace because the mechanism requires an ideal linear spring whose natural length is zero to compensate gravity torque regardless of the link posture θ. To solve this problem, in the system (c) a spring is installed within the link and the end of the spring is connected to the base link using a wire-pulley mechanism.

The system (d) has a counterweight at the end of the link to compensate the weight of the end effector. This mechanism can keep the center of gravity on the joint axis, thereby ensuring that the moment of gravitational force of the base link remains zero regardless of the angle. This configuration is suitable for applications where the base link cannot be rigidly fixed to the ground.

In the article it is also presented a development of a five-bar linkage parallel link arm with active joints and apply the weight compensation mechanism. The balancing system chosen by the authors among the four mentioned is the second one (b) with the implementation of a non-circular pulley rotating with arm. Figure 7 shows the basic principle of the system for a one DOF pendulum.

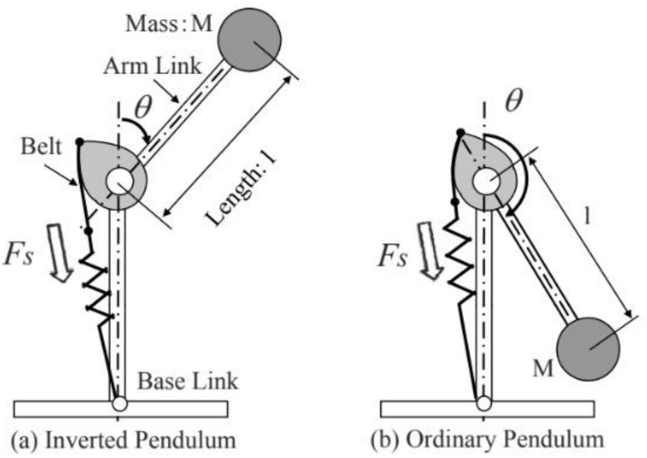


Figure 8 Weight compensation mechanism with a non-circular pulley and a spring. [10]

The arm link with the end-mass freely rotates w.r.t the base link while the non-circular pulley is fixed to the arm link. One end of the spring attaches to the base link while the other connects to the flexible section without elongation, like a wire or a belt. This flexible part's end is secured to the pulley, and it wraps around the non-circular pulley. Consequently, rotation of the arm link winds the flexible part, causing the stretched spring to produce compensatory torque, with magnitude equals to the spring force times the diameter of the circular pulley . If we can design pulley radius $r(\theta)$ satisfying the following identity, the system becomes totally balanced with zero gravity.

$$F_s \cdot r(\theta) = Mgl \sin(\theta), \tag{1}$$

where F_S is the spring force and M, g, l are the weight of the end mass, gravity acceleration and link length.

The basic principle is extended to a three DOF light weight manipulator designed for pick and place tasks.

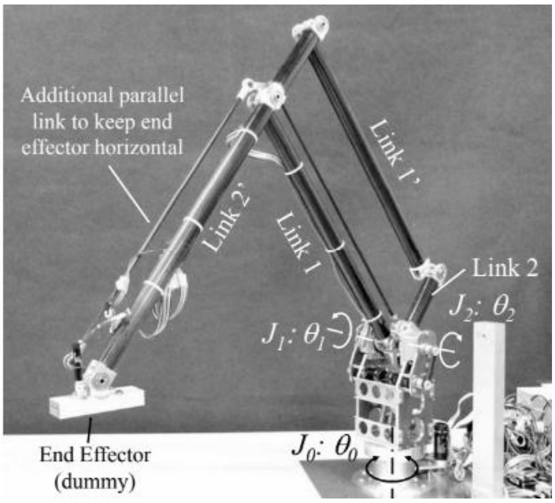


Figure 9 Prototype of a parallel five-bar linkage arm for a light duty operation. [10]

The gravity torque for each joint can be reduced to a 1-DOF pendulum system model as mentioned previously. The gravity torque τ1 for joint J1 is equivalent to an inverted pendulum with an end mass of 1.18kg and a link length of 0.5m. Similarly, τ2 is equivalent to an ordinary pendulum with a mass of 0.33kg and a link length of 0.5m. τ1 is much larger than τ2 due to the longer link length. The specifications of the springs for J1 and J2 are (x0 = 100mm, k = 1.15N/mm, F0 = 14.71N, MISUMI Group Inc.: AUFM12-100) and (x0 = 80mm, k = 1.57N/mm, F0 = 14.71N, AUFM12-80), respectively. These springs are connected in parallel to the joint actuation to improve weight compensation accuracy. To enhance weight compensation accuracy further, a steel belt is used to connect the spring and the non-circular pulley instead of a wire. The steel belt has a thickness of 0.1mm and negligible elongation with sufficient strength. Both J1 and J2 joints have the same driving mechanism. The output of the harmonic unit drives the non-circular pulley fixed to the link structure. The non-circular pulley winds the steel belt and stretches the spring to generate compensation torque. It's important to note that the introduction of the spring does not essentially decrease joint stiffness because the spring is connected parallel to the joint actuation. The introduction of the weight compensation mechanism significantly reduces the maximum static torque by 50-80% to sustain the same static posture.

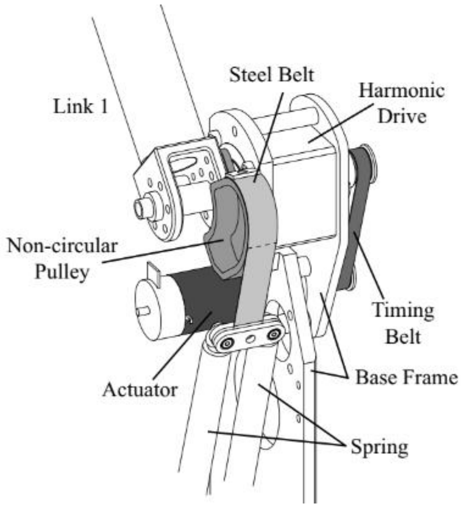


Figure 10 Joint driving mechanism with the non-circular pulleys and the springs. [10]

In the article “Multi-DOF Counterbalance Mechanism for a Service Robot Arm” [11], another counterbalance mechanism using springs and pulleys is discussed. The concept model of the proposed mechanism is shown in figure 10 for a single-DOF link, however the authors developed the mechanism for a 5-DOF manipulator.

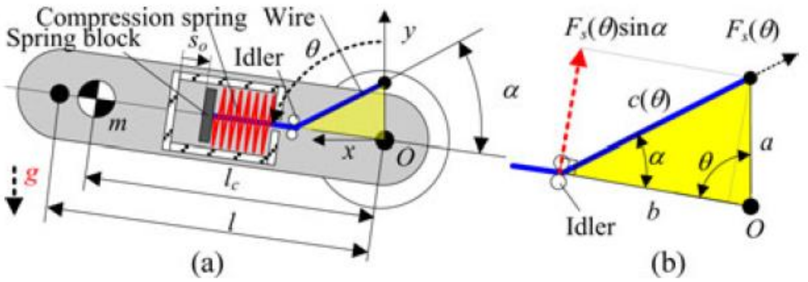


Figure 11 1-DOF counterbalance mechanism: (a) concept model and (b) closeup view of triangle in general position. [11]

To compensate for the gravitational torque of the robot, a compression spring and a wire are used: when the link is rotated, the spring is compressed by the spring block pulled by the wire.

The spring force is given by

$$F_S(\theta) = k\,\{\,s_o + c\,(\theta) - \,(b - a)\} \tag{2}$$

where k is the spring stiffness, s_o is the initial compression length of the spring, and α,a,b, $c(\theta)$ are defined in Figure 10. The counterbalancing torque T_c is given by:

$$T_c(\theta) = b\, F_s(\theta)\sin(\alpha) \tag{3}$$

with $c = (a^2 + b^2 - 2ab\cos(\theta))^{\frac{1}{2}}$ and $\sin(\alpha) = \frac{a\sin(\theta)}{c}$.

In serial robot arms, the gravitational torque of joint j ($j = 1, .., n$) is the gravitational torques of joint j to joint n. In the case of a 3-DOF manipulator, the difference torque T_d (the difference between gravitational and counterbalancing torques) at each joint may be calculated by

$$T_{d1} = T_{g1} - (T_{c1} + T_{c2} + T_{c3})$$

$$T_{d2} = T_{g2} - (T_{c2} + T_{c3})$$

$$T_{d3} = T_{g3} - T_{c3} \tag{4}$$

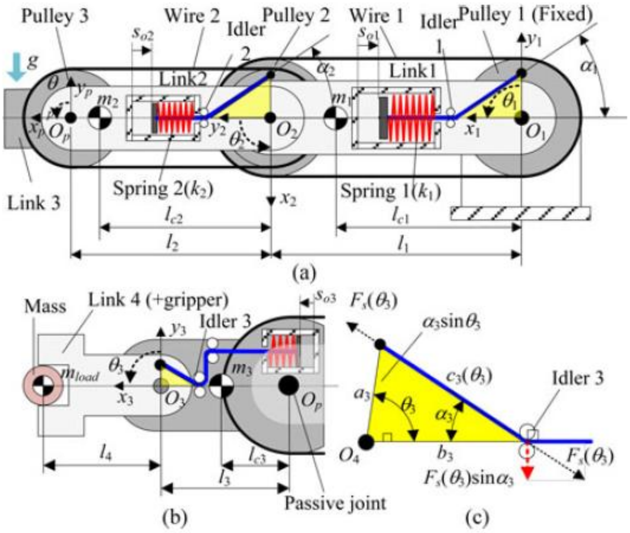


Figure 12 3-DOF counterbalance mechanism for: (a) lower 2 joints, (b) wrist joint, and (c) closeup view of triangle in general position. [11]

Then, a 5-DOF robotic arm is built using the proposed counterbalance mechanism. The authors declare that the required specifications of the motors and speed reducers can be greatly decreased. Among the three types of revolute joints (i.e., roll, pitch, and yaw joints), only pitch joints are affected by gravity. Therefore, the counterbalance mechanism is installed at each pitch joint of the robot. This robot arm is constructed of yaw–pitch–pitch–pitch (passive joint)–yaw–pitch joints (Figure 12). The passive pitch joint has no motor and speed reducer, and it is coupled with joints 1 and 2 through the double parallelogram mechanism to maintain a constant absolute angle. The counterbalance mechanism is not required for joints 1(yaw) and 4(yaw) since they always rotate parallel to the ground. Thus, the proposed 3-DOF counterbalance mechanism needs to be applied to only joints 2, 3, and 5.

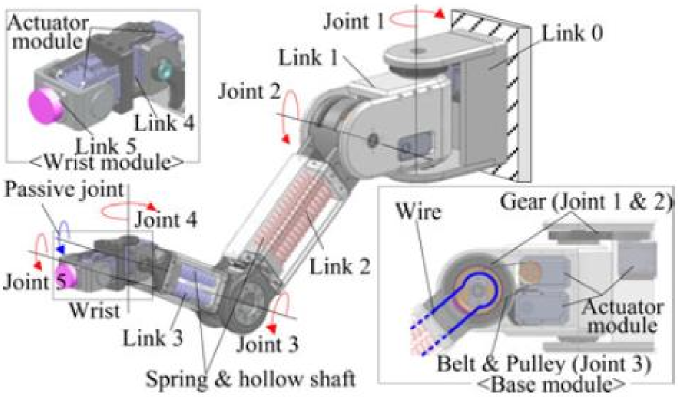


Figure 13 Design of the counterbalance robot arm. [11]

Joint 2 is connected with the actuator module through a pinion to provide an additional gear ratio of 1:1.9. To decrease the arm mass, joint 3 is operated by the actuator module placed at the base module of the robot via a tendon-driven mechanism with a gear ratio of 1:1.9. Wires of the tendon drive and parallelogram mechanism are placed inside hollow shafts. Most parts of the counterbalance robot arm are constructed with an aluminum alloy to achieve light weight. The proposed counterbalance mechanism was designed so that it can be embedded inside the robot arm. Since the counterbalance mechanisms at joints 2 and 3 should cancel the gravitational torques of the robot arm, the mass and the center of mass of each link are optimized to obtain the best results. Several springs were employed to provide the proper counterbalancing torque and sufficient compressible length. Fig. 14 shows the designed counterbalance mechanism for joint 2: spring blocks are pulled by wires, which are fixed at the reference planes through the idlers when each link is rotated in the CCW direction. Four bushes are installed inside the spring block to minimize the friction as the spring block moves along the hollow shafts. Wires for the parallelogram mechanism and tendon drives are passed through the hollow shafts and connected with the driven pulleys of the next joint. The proposed counterbalance mechanism enabled the construction of a 5-DOF arm with a reach of longer than 600 mm, a payload of 2 kg and an endpoint speed greater than 2 m/s.

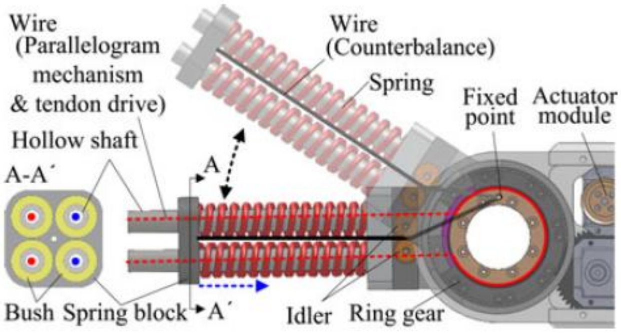


Figure 14 - Counterbalance mechanism for joint 2. [11]

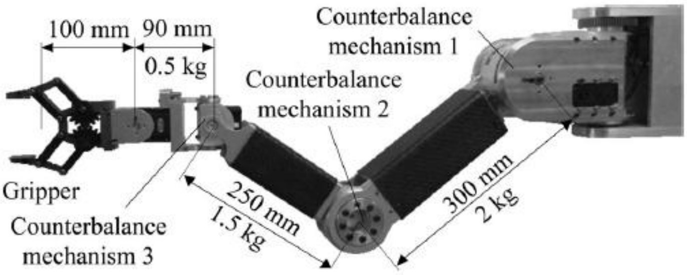


Figure 15 Prototype of the counterbalance robot arm. [11]

Table 1 Specification of the counterbalance robotic arm. [11]

Specifications				
DOF	5 DOF			
Structure	yaw-pitch-pitch-pitch(passive joint)-yaw-pitch			
Payload	2.5 kg			
Arm length	640 mm (Joint 2 ~ joint 5)			
Maximum Speed	2 m/s			
Allowable torque (with additional gear ratio)	Joint	Actuator torque	Additional gear ratio	Allowable joint torque
	1	6.0 Nm	1 : 1	6.0 Nm
	2	8.4 Nm	1 : 1.9	16.0 Nm
	3	6.0 Nm	1 : 1.9	11.4 Nm
	4	2.5 Nm	1 : 1	2.5 Nm
	5	2.5 Nm	1 : 1	2.5 Nm

Table 2 Design parameters for the counterbalance mechanism [11]

Joint	Join 2	Joint 3	Joint 5
Spring k_j	8.2 kN/m	6.4 kN/m	1 kN/m
Initial com. s_{ij}	25 mm	20 mm	10 mm
Moment arm l_c	150 mm	120 mm	90 mm
a	25 mm	20 mm	15 mm
b	50 mm	40 mm	25 mm

The torques required to operate the robot arm were greatly reduced since the gravitational torques were compensated for by the counterbalancing torques from the counterbalance mechanisms. Gravitational torques of up to 29.6, 13.4, and 2.0 Nm were applied to the three joints, respectively. Note that these values are above the maximum allowable torque of each joint. On the other hand, with the proposed mechanism, the required torque greatly reduced to 13.8, 7.9 and 1.6 Nm. They fell below the maximum allowable torque of the robot arm, which implies that the arm can be fully controlled with the actuator modules.

Suppose that the spring constant at this joint is increased four times from 1 to 4 kN/m. The 4 kN/m spring would generate a compensating torque of –1.6 Nm. This torque would obviously overcompensate for the gravitational torque in the case of no payload, and the actuator would have to provide 1.2 Nm in the direction of gravity. However, with this increased stiffness of the spring, the robot could handle even a 4-kg payload, which would result in a maximum of 3.5 Nm of the gravitational torque. In conclusion, a spring with higher stiffness could lead to a further increased payload capacity of the robot arm, which would allow joint 5 to handle more than a 2-kg load.

In the next lines, a basic principle of the CRCM behavior is presented [12]. The double pendulum in Figure 16 is balanced with the two CRCMs m_1^* and m_2^*.

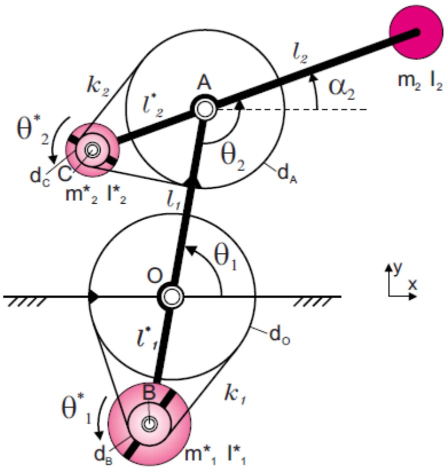


Figure 16 - Basic CRCM-balanced double pendulum. [12]

Mass m_2 is the force balanced about A by CRCM m_2^*, which is a lumped mass with inertia I_2^*. CRCM m_1^* with inertia I_1^* is also lumped mass and is used to force balance the linkage about the origin O.

For the moment balance of link 2, a gear about A is mounted on link 1 and drives m_2^* with a chain by which it rotates with transmission ratio k_2. This CRCM rotates in the opposite direction of link 2, which makes $k_2$negative. The same is done at O where a gear attached to the base drives m_1^* in the opposite direction of link 1, k_1 is negative. It is also possible to drive the CRCMs by using a set of external gears, with internal gears or with pulleys and belts. Balancing problems by using gears and chains can occur due to backlash, while for a belt transmission, they are due to the elasticity of the belt.

2.3 Actuation systems

The motion imposed on a manipulator's joint is realized by an actuating system which in general consist of:

- Power supply,
- A power amplifier,
- A servomotor,
- A transmission,

Actuators can be classified in into three groups:

- *Pneumatic motors*: utilize the pneumatic energy provided by a compressor and transform it into mechanical energy by means of pistons or turbines.
- *Hydraulic motors*: transform the hydraulic energy stored in a reservoir into mechanical energy by means of suitable pumps.
- *Electric motors*: transform electric energy into mechanical energy.

In the robotics field, the most employed actuators are electric motors. Among them, the most popular are permanent-magnet direct-current (DC) servomotors and brushless DC servomotors, due to their control flexibility. Instead, hydraulic servomotors are employed when the manipulator have to carry out heavy payloads.

The joints in a manipulator require high torques for low speed, thus interposing a transmission between the motor and the joint is needed in order to optimize the transfer of the mechanical power. The choice of the transmission depends on the power requirements, the kind of desired motion and the allocation of the motor with respect to the joints. Transmissions allow the motor to mount on the basis of the manipulator leading to reduce the weight on joints.
The typical transmission used in industrial robots are:

- *Spur gears* that modify the characteristics of the rotational motion of the motor by changing the axis of rotation and/or by translating the application point; spur gears are usually constructed with wide cross-section teeth and squat shafts.
- *Lead screws* that convert rotational motion of the motor into translational motion, as needed for actuation of prismatic joints; in order to reduce friction, ball screws are usually employed that are preloaded so as to increase.
- *Timing belts* and *chains* are equivalent from a kinematic viewpoint and are employed to locate the motor remotely from the axis of the actuated joint. The stress on timing belts may cause strain, and then these are used in applications requiring high speed and low forces. On the other hand, chains are used in applications requiring low speeds, since their large mass may induce vibration at high speeds.

Chapter 2

3 Project requirements

3.1 Operational space definition

The mobile robot shall be designed for taking a wine sample from the barrels, whose taps are in a range between 1m and 1.80m above the ground. The barrel taps are accessible from a 2-meter-wide corridor in which the barrels are arranged side by side along the length, both on the right and left, as shown in Figure 17. The mobile manipulator should move down the aisle, stop in front of the first two barrels, center the end effector upon the tap and take a sample from the left barrel, move and center the end effector upon the tap of the right barrel and take a sample, and then advance to the next barrels in order to perform the same tasks. Once the target position of the mobile robot is reached, the manipulator must be capable of making a rotation of at least 180 degrees to bring the end effector from the left tap to the right tap (or vice versa). If the robot followed an ideal path, it would advance in the center of the corridor while maintaining the same distance of 1 m from both the left and right barrels. However, there could be obstacles or positioning errors of the moving platform that may cause the manipulator base to stop in a less than ideal position. Therefore, it is necessary to design the manipulator in a way that the end effector can reach the tap even if the manipulator base is in a suboptimal position. Considering the ideal position, the manipulator must be able to bring the end effector at a heigh between 1m and 1,8m off the ground to reach the tap, and it must have an extension that allows it to move the end effector 1 meter away from the base.

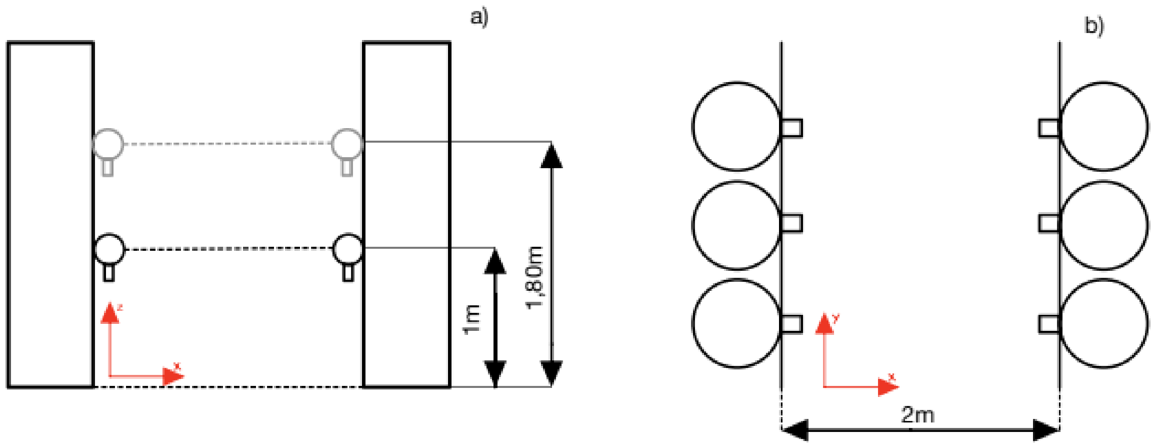


Figure 17 Winery representation, a) front view - b) top view.

In Figure 18 are shown three cases where the mobile platform is in a position that differs from the ideal one. The square stands for the mobile platform while the point inside represents the base of the manipulator.

- Case (a): the mobile platform is placed at a distance l_x from the center of the corridor along the x-axis. To allow the end-effector to reach the barrels tap, the lengths of the manipulator links shall be designed to achieve a correct work of the manipulator even in case of translations along the x-axis of the corridor.
- Case (b): the mobile platform is placed at a distance l_y from the axis of the barrel tap. This problem can be solved by changing the orientation of the end effector through an actuated joint or allowing the manipulator base to move along the direction of the y axis through a guide placed on the mobile platform, with the goal of bringing $l_y = 0$.
- Case (c): the mobile platform is inclined at an angle θ with respect to the y-axis. To reach the correct position, the manipulator shall compensate the angle with a rotation around the z-axis, through a rotative joint.

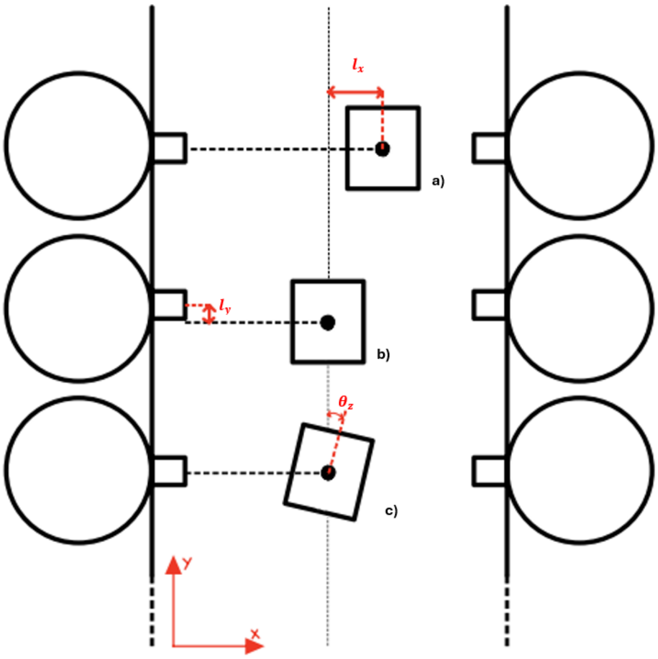


Figure 18 Positioning of the mobile platform different from the ideal one, a) translation along the x-axis, b) translation along the y-axis, c) rotation around z-axis.

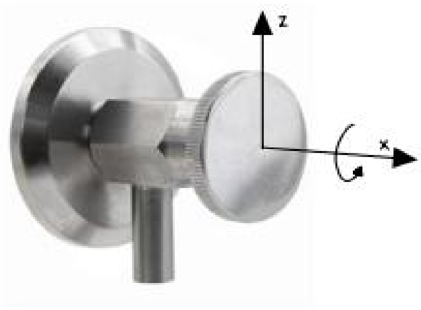


Figure 19 Valve to open and close the tap of the barrel.

The design phase of the robotic arm is aimed at setting the end effector in the correct position in front of the valve, shown in Figure 19. Once the end effector reaches the position, it must perform a rotation around the x-axis to open or close the tap, however the methods of screwing and unscrewing are treated in the design of the end effector. The end effector design also considers the possibility of correcting any small centering errors with the use of flexible materials that lead to a more accurate centering of the valve. Since the end effector only has to unscrew the valve and position the funnel below the tap without having to make any further movements to place the sample elsewhere, the axis of the end effector can be kept parallel to the x-axis in a passive manner, as in the manipulator shown in Figure 6 in Chapter 1. With this assumption, it becomes more advantageous to choose the guide on the moving platform to correct the positioning of case B.

It is hypothesized that the moving platform can have maximum positioning errors of 10 cm along the x and y axis and a maximum angular positioning error of 15 degrees. These parameters are used to calculate the distances from the base to be reached by the end effector.

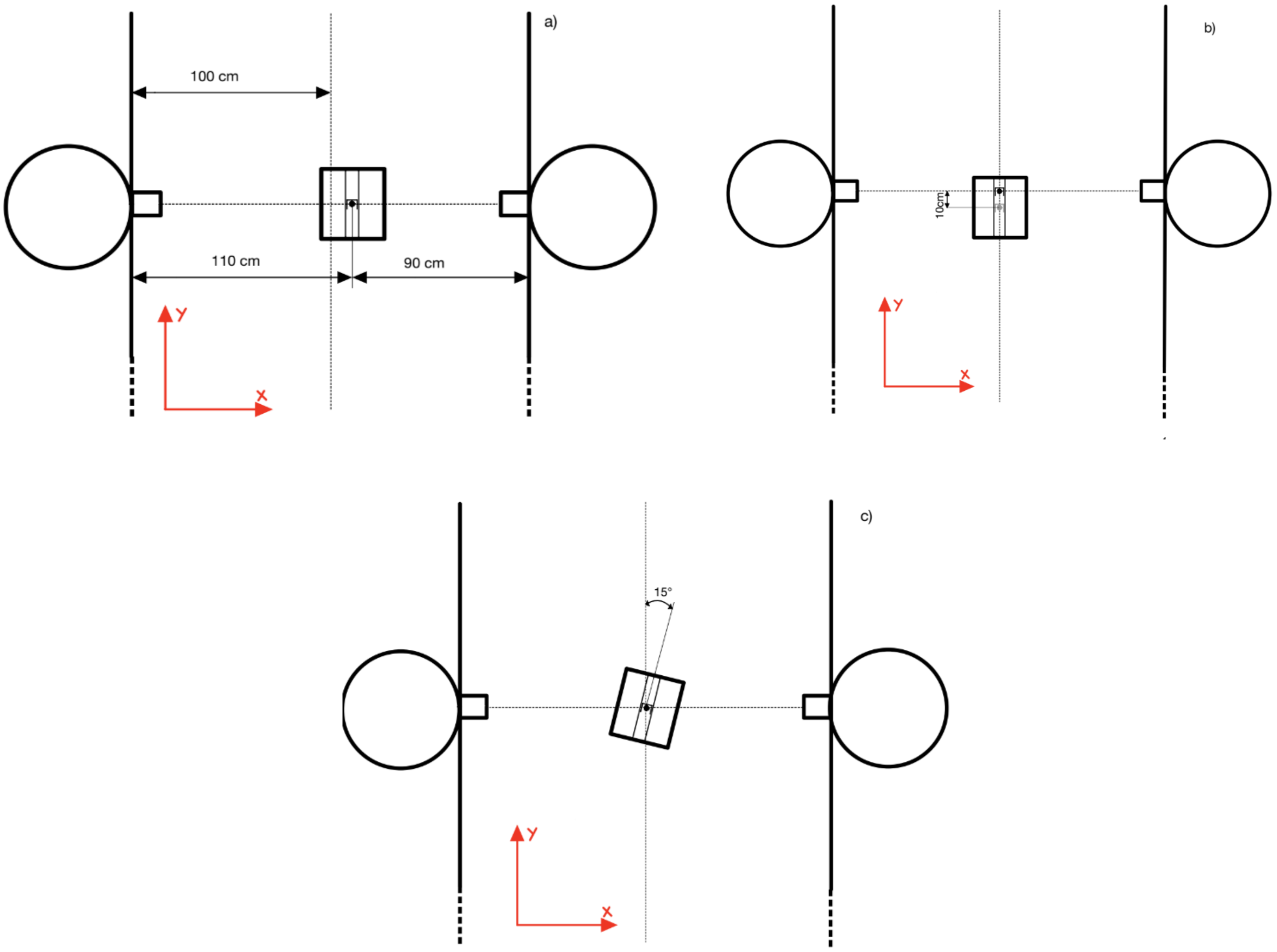


Figure 20 Distances between the base and the tap, a) case positioning error along x axis, b) case positioning error along y axis, c) case positioning error around z axis.

Considering the x direction, and assuming a positioning error equal to 10 cm along the x axis, the manipulator shall pose the end effector at a distance of 110 cm from the base, and, after a rotation of 180° around the z axis, pose the end effector in front of the other tap at 90 cm from the base. Thus, the robotic arm must be designed to work properly at both distances. Regarding the y axis, it has been considered to correct the error with a prismatic guide installed on the mobile platform that must ensure a movement up to 10 cm. The angular positioning error can be easily fixed by a rotative joint which can rotate up to 360° around the z axis. The mobile robot is assumed to be between 60 and 80 cm high, and in Figure 21 are shown the distances to be reached by the end effector along the z axis. Considering the base of the manipulator installed 70 cm from the ground, the end effector must be able to reach heights in a range between 30 and 110 cm from the moving platform, these taking into account that along the x axis the base can be between 90 and 110 cm away from the tap.

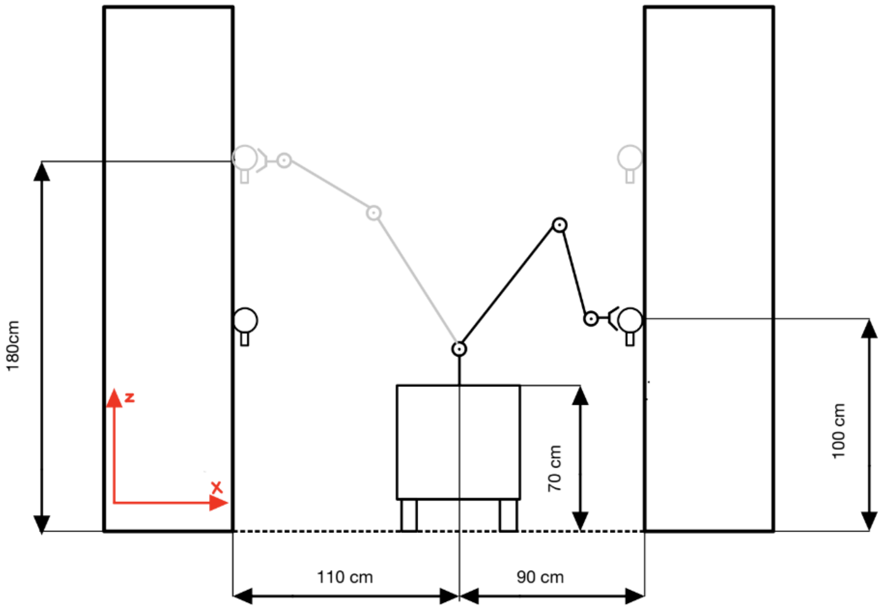


Figure 21 Heights to be reached by the end effector.

However, the case in which an obstacle is present in the corridor should also be considered, which could lead to a further movement away from the center along the x axis. The worst-case scenario occurs when the obstacle is in the center of the corridor precisely in front of the taps, forcing the moving platform to position itself to one side. Assuming a moving platform width between 60 and 70 cm, the base of the manipulator could be positioned 50 cm from one tap and 150 cm from the opposite tap. Therefore, even in this case, the robotic arm must work properly at both distances. The cylindrical operational space is shown in Figure 22, with the two most critical cases to be reached i.e., the farthest and the nearest tap.

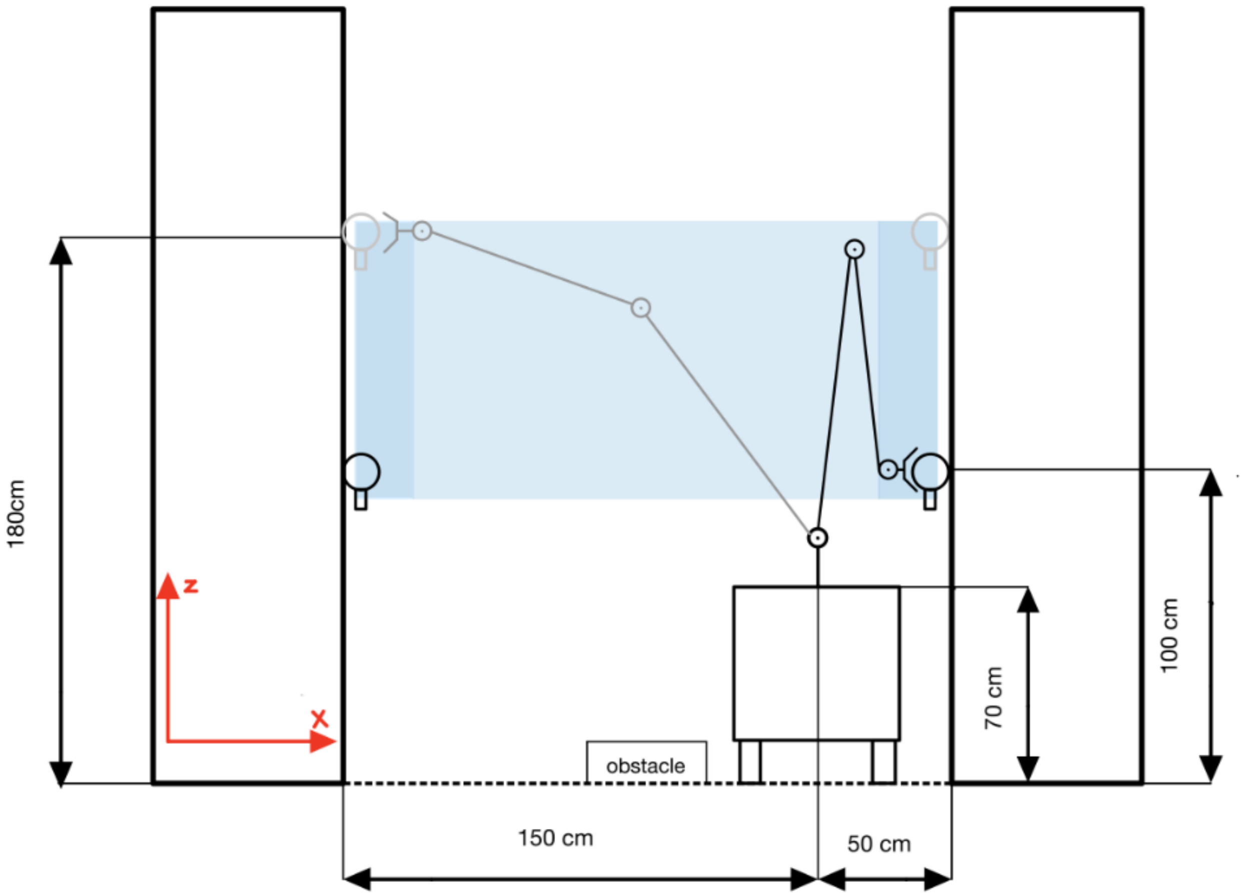


Figure 22 Cylindrical operational space with the presence of an obstacle in the center of the corridor.

The goal is to design a manipulator with a double parallelogram configuration with 4 degrees of freedom. The required actuated joints consist of 1 prismatic joint along y (J1), 1 rotative joint around z (J2) and 2 rotative joints around x (fixed frame in Figure 23).

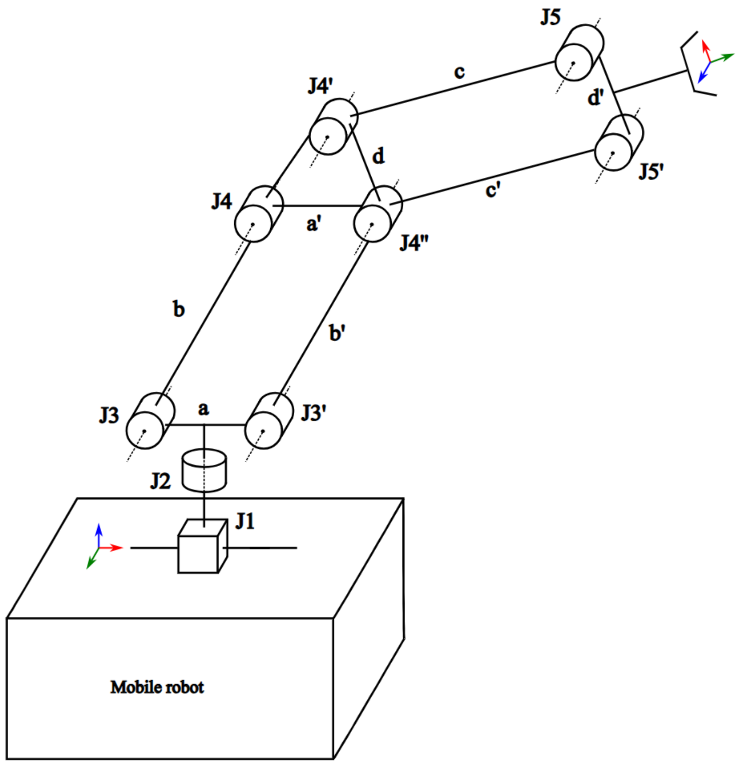


Figure 23 Double parallelogram manipulator with 4 DOF.

3.2 Sizing the double parallelogram manipulator

The first step in sizing the double-parallelogram manipulator is to define the lengths of links b and c with reference to Figure 24. In choosing the size of the two links, we need to make sure that the manipulator will reach both the farthest tap and the nearest tap in case of an obstacle in the corridor, as previously shown in Figure 22. Before starting with the dimensioning of the double parallelogram, the manipulator is approximated with an R-R configuration. The maximum and minimum lengths between the base and the required position of the end-effector are considered to be equal to $L_{max}(x,y) = (1.50\ m, 1.10\ m)$ and $L_{min}(x,y) = (0.50\ m, 0.30\ m)$.

Starting from L_{max} and considering l_{max} as the length of the diagonal of the rectangle 1.50 x 1.10 m, the equation (5) must be satisfied to reach the point $(1.50\ m, 1.10\ m)$:

$$b + c \geq\ k_M\ l_{max}. \tag{5}$$

For L_{min}:

$$b - c \leq\ k_m l_{min} \tag{6}$$

with k_M and k_m coefficients to increase l_{max} and to decrease l_{min} respectively. From the equations (5-6):

$$b = \frac{(l_{max}+l_{min})}{2}, \tag{7}$$

$$c = \frac{(l_{max}-l_{min})}{2}. \tag{8}$$

By choosing $k_M = 1.1$ and $k_m = 0.4$, the lengths of the links b and c are $b = 1.14\ m$ and $c = 0.9m$.

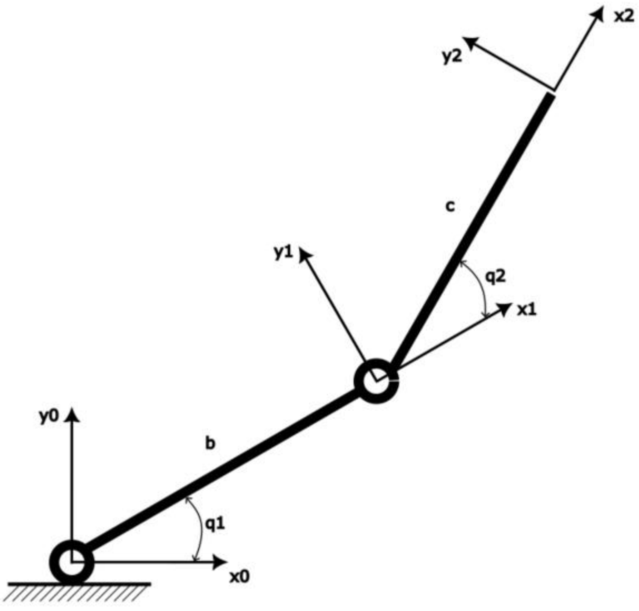


Figure 24 R-R configuration.

Through the equations (9-10), it is possible to compute the coordinates of the mobile frame 2 with respect to the fixed frame 0 depending on the angles q_1 and q_2.

$$x(q_1, q_2) = b\cos(q_1) + c\cos(q_1 + q_2), \tag{9}$$

$$y(q_1, q_2) = b\sin(q_1) + c\sin(q_1 + q_2). \tag{10}$$

Considering the maximum and minimum distances to be reached by the end effector:

$$x_{min} = b\cos(q_1) + c\cos(q_1 + q_2) = 0.5m, \tag{11}$$

$$y_{min} = b\sin(q_1) + c\sin(q_1 + q_2) = 0.3m, \tag{12}$$

$$x_{max} = b\cos(q_1{'}) + c\cos(q_1{'} + q_2{'}) = 1.50m, \tag{13}$$

$$y_{max} = b\sin(q_1{'}) + c\sin(q_1{'} + q_2{'}) = 1.10m, \tag{14}$$

with $q1\ and\ q1' \in [0°, 180°]$.

With $b = 1.14\ m$ and $c = 0.9\ m$, (11) and (12) are satisfied with $q1 = 82.3°$ and $q_2 = -149.6°$ while (13) and (14) are satisfied when $q1' = 57.6°$ and $q_2' = -48.8°$.

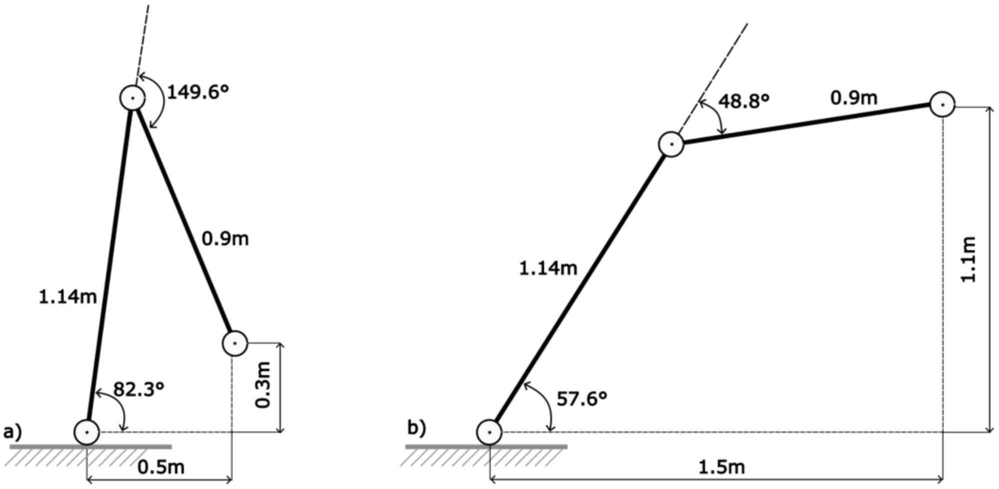


Figure 25 Pose to reach the nearest tap (a), pose to reach the farthest tap (b), R-R configuration.

The values of q_1 and q_2 needed to reach the corners of the operational space are listed in Table 4.

Table 3 - Values of q_1 and q_2 to reach the position (x,y) from the base of the manipulator (R-R configuration).

x	y	q_1	q_2
0.5m	0.3m	82.3°	-149.6°
1.5m	1.1m	57.6°	-48.8°
0.5m	1.1m	110°	-108.4°
1.5m	0.3m	47°	-83.5°

Given the results of q_1 and q_2 we proceed by dimensioning the double parallelogram configuration. Referring to the Figure 26, with $b = b' = 1.14m$ and $c = c' = 0.9m$, link lengths a, d, d' are assumed to be $a = d = d' = 0.12m$.

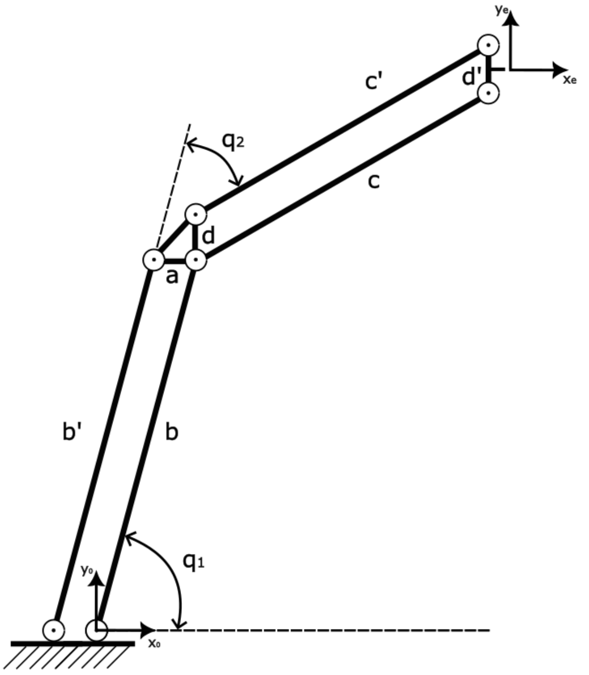


Figure 26 Double parallelogram configuration.

The coordinates of the end-effector with respect to the frame 0 are:

$$x(q_1, q_2) = b\cos(q_1) + c\cos(q_1 + q_2), \tag{15}$$

$$y(q_1, q_2) = b\sin(q_1) + c\sin(q_1 + q_2) + \frac{d}{2}. \tag{16}$$

The intention is to ensure, during the motion, that the angle θ (Figure 27) between the two links of the parallelogram is as close as possible to 90°. We intend to compute the angles needed to reach the extremes of the required workspace to orientate the two parallelograms of the manipulator by choosing a value of α. By the (15-16), the values of q_1 and q_2 are tabulated in Table 6.

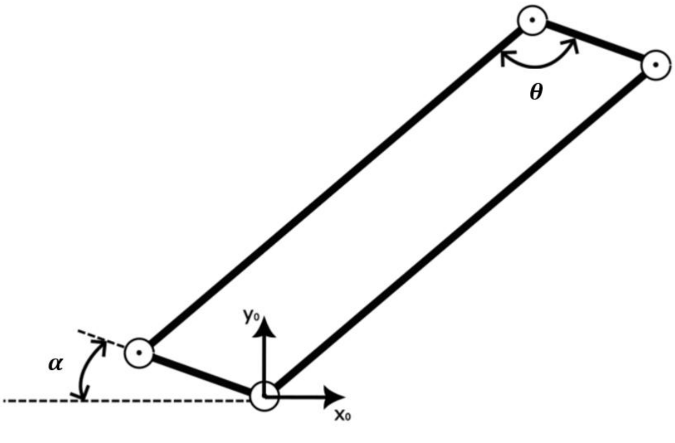


Figure 27 Orientation of the parallelogram through the angle α.

Table 4 Values of q_1 and q_2 to reach the position (x,y) from the base of the manipulator (double parallelogram configuration).

x	y	q_1	q_2	$q_1 + q_2$
0.5m	0.3m	76.5°	-151.4°	-74.8°
1.5m	1.1m	58°	-53°	4.6°
0.5m	1.1m	110.5°	-112.3°	-1.7°
1.5m	0.3m	45.2°	-84.5°	-39.2°

Starting from the configuration in Figure 26 where the link a is parallel to the x-axis while the link d is parallel to the y-axis, we choose the maximum and minimum values of both q_1 and $q_1 + q_2 = q_{12}$:

$$q_{1_{max}} = 110.5°,\ q_{1_{min}} = 45.2°,\ q_{12_{max}} = 4.6°, q_{12_{min}} = -74.8°.$$

The mean of q_1 *and* q_{12}:

$$q_{1_{mean}} = 77.85°,\ q_{12_{mean}} = -35°.$$

By averaging the maximum and minimum angles we orientate the two parallelograms changing α, ensuring that θ is equal to 90° when $q_{1=}\ q_{1_{mean}}$ for the first parallelogram (links a-b) and $q_{12=}\ q_{12_{mean}}$ for the second (links c-d). For the first parallelogram, when $q_1 = 77.85°$ and $\theta = 90°$, then $\alpha = 12.15°$, while for the second parallelogram, when $q_{12} = -35°$ and $\theta = 90°$, the angle α is $\alpha = 125°$. The final configuration is shown in Figure 28, where the maximum values of θ are respectively $\theta \approx 33°$and $\theta \approx 39°$ for the first and the second parallelogram.

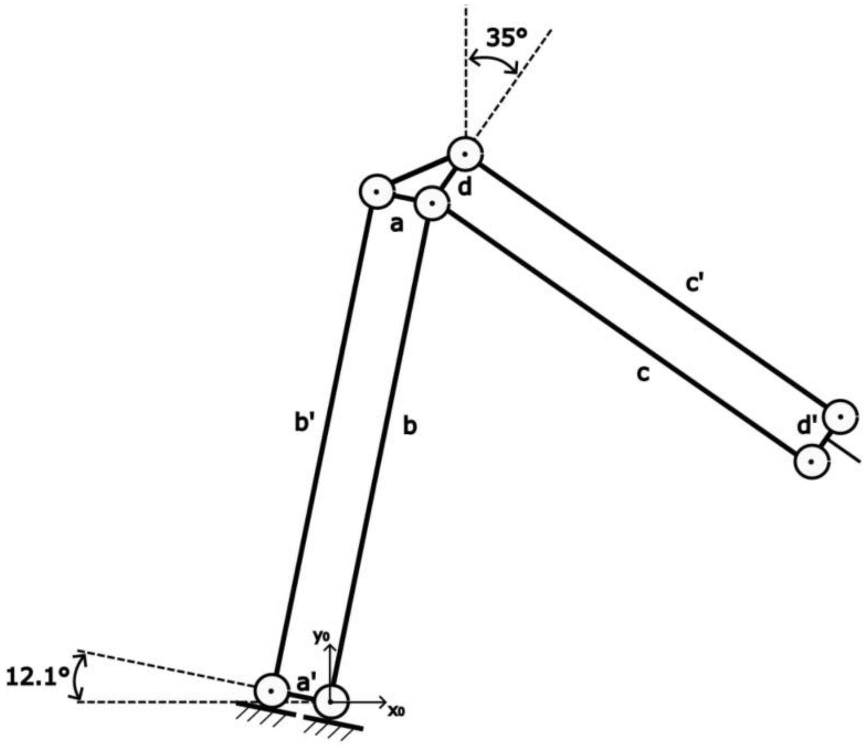


Figure 28 Links a and d orientation, double parallelogram manipulator.

With this configuration, the coordinates of the end-effector become:

$$x(q_1, q_2) = b\cos(q_1) + c\cos(q_1 + q_2) + \frac{d}{2}\sin(35°), \tag{17}$$

$$y(q_1, q_2) = b\sin(q_1) + c\sin(q_1 + q_2) + \frac{d}{2}\cos(35°). \tag{18}$$

The updated values of q_1 and q_2 to reach the corners of the operational space, computed through the equations (17-18) are listed in Table 6:

Table 5 Final values of q_1 and q_2 to reach the position (x,y) from the base of the manipulator (double parallelogram configuration).

x	y	q_1	q_2	$q_1 + q_2$
0.5m	0.3m	78.7°	-153.1°	-74.4°
1.5m	1.1m	60.1°	-56.1°	4°
0.5m	1.1m	112.4°	-112.6°	-0.2°
1.5m	0.3m	46.9°	-87.1°	-40.2°

Figure 29 illustrates the reachable workspace of the designed robotic arm, denoted by the blue area, within the parameter constraints of $140° \geq q_1 \geq 30°$ and $20° \geq q_1 + q_2 \geq -90°$. The orange rectangle delineates the required workspace for the end effector to achieve the sampling task.

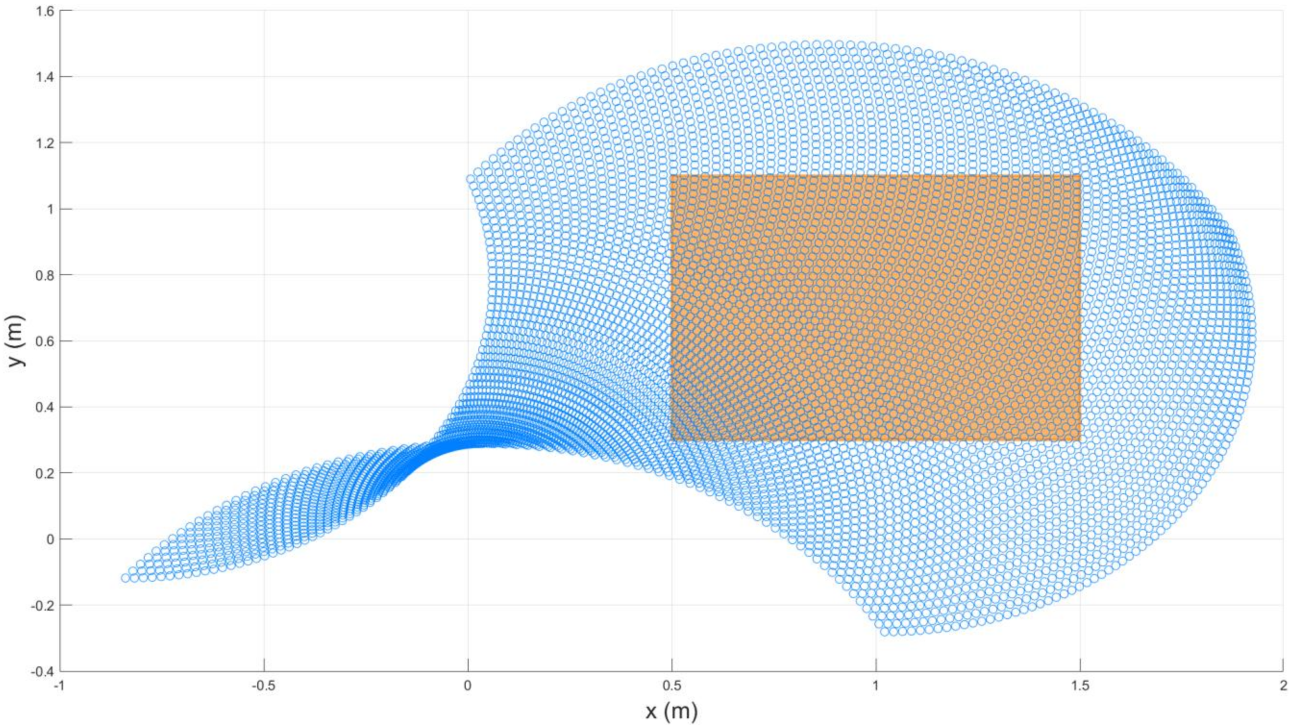


Figure 29 Workspace and operational space.

Chapter 3

4 Kinematics

4.1 Forward Kinematics

To define the forward kinematic model of our 4-degrees-of-freedom manipulator with a double parallelogram configuration, the Denavit-Hartenberg convention is employed. The objective is to compute the transformation matrix T_e^0, thus determining the orientation and the position of the frame {e} (end-effector), with respect to the frame {0}, fixed on the mobile platform of the robot. The manipulator structure, along with its respective joints and axes, is depicted in Figure 30. The robotic arm is characterized by 4 active joints (J1,J2,J3,J4), while the Joint J5 is a passive joint. Thus, we begin to compute T_4^0. The DH parameters are listed in Table 8.

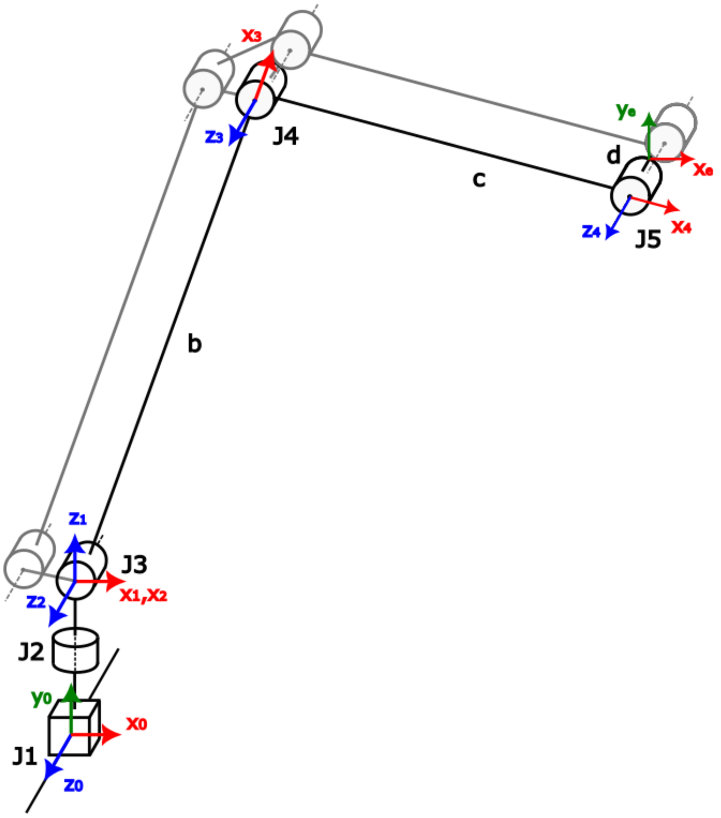


Figure 30 Representation of the frames according to the DH convention.

Table 6 DH parameters.

	a_i	α_i	d_i	q_i
1	0	$-\frac{\pi}{2}$	d_1	0
2	0	$\frac{\pi}{2}$	0	q_2
3	b	0	0	q_3
4	c	0	0	q_4

Notation: $\cos(q_i) = c_i$, $\sin(q_i) = s_i$, $\cos(q_i + q_j) = c_{ij}$, $\sin(q_i + q_j) = s_{ij}$.

$$T_1^0(d_1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(-\frac{\pi}{2}) & -sin(-\frac{\pi}{2}) & 0 \\ 0 & sin(-\frac{\pi}{2}) & cos(-\frac{\pi}{2}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{19}$$

$$T_2^1(q_2) = \begin{bmatrix} c_2 & -s_2 \cos\left(\frac{\pi}{2}\right) & s_2 \sin\left(\frac{\pi}{2}\right) & 0 \\ s_2 & c_2 \cos\left(\frac{\pi}{2}\right) & -c_2 \sin\left(\frac{\pi}{2}\right) & 0 \\ 0 & \sin\left(\frac{\pi}{2}\right) & \cos\left(\frac{\pi}{2}\right) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_2 & 0 & s_2 & 0 \\ s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{20}$$

$$T_3^2(q_3) = \begin{bmatrix} c_3 & -s_3\, cos(0) & s_3\, sin(0) & b\, c_3 \\ s_3 & c_3\, cos(0) & -c_3\, sin(0) & b\, s_3 \\ 0 & sin(0) & cos(0) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_3 & -s_3 & 0 & b\, c_3 \\ s_3 & c_3 & 0 & b\, s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{21}$$

$$T_4^3(q_4) = \begin{bmatrix} c_4 & -s_4 cos(0) & s_4\, sin(0) & c\, c_4 \\ s_4 & c_4\, cos(0) & -c_4\, sin(0) & c\, s_4 \\ 0 & sin(0) & cos(0) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_4 & -s_4 & 0 & c\, c_4 \\ s_4 & c_4 & 0 & c\, s_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{22}$$

T_4^0=$T_1^0 T_2^1\ T_3^2\ T_4^3$:

$$T_4^0(d_1, q_2, q_3, q_4) = \begin{bmatrix} c_2\, c_{34} & -c_2\, s_{34} & s_2 & c_2\,(b\, c_3 + c\, c_{34}) \\ s_{34} & c_{34} & 0 & b\, c_3\ + c\, c_{34} \\ -s_2 c_{34} & s_2 s_{34} & c_2 & d_1 + s_2 \cdot (-b\, c_3 - c\, c_{34}) \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{23}$$

Considering that Joint 5 is not actuated, the end effector orientation with respect to the frame {0} does not depend on q_3 and q_4. Thus, it is convenient to choose the end effector's frame in such a way that its y-axis is parallel to the y-axis of frame {0}. Is assumed the position of the end-effector at the midpoint of the length of link d. Based on the geometry shown in Figure 31, we proceed to calculate the transformation matrix between frame {4} and frame {e}.

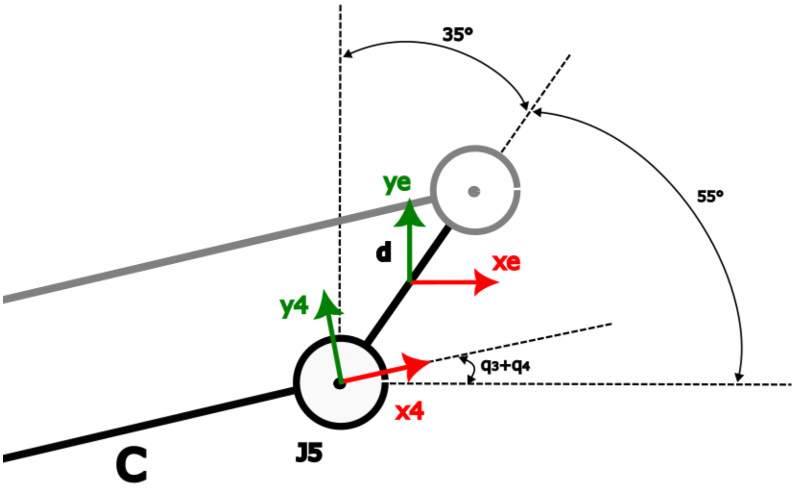


Figure 31 Frame {4} and frame {e}.

Defining $\beta = 55°$:

$$T_e^4(q_3,q_4) = \begin{bmatrix} c_{34} & s_{34} & 0 & c_{34}\frac{d}{2}\cos(\beta) + s_{34}\frac{d}{2}\sin(\beta) \\ -s_{34} & c_{34} & 0 & -s_{34}\frac{d}{2}\cos(\beta) + c_{34}\frac{d}{2}\sin(\beta) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{24}$$

The transformation matrix $T_e^0 = T_4^0 T_e^4$:

$$T_e^0(d_1,q_2,q_3,q_4) = \begin{bmatrix} c_2 & 0 & s_2 & c_2(b\,c_3 + c\,c_{34} + \frac{d}{2}\cos(\beta)) \\ 0 & 1 & 0 & b\,s_3 + c\,s_{34} + \frac{d}{2}\sin(\beta) \\ -s_2 & 0 & c_2 & d_1 + s_2(-\frac{d}{2}\cos(\beta) - b\,c_3 - c\,c_{34}) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{25}$$

The translation vector $t_e^0 = \begin{bmatrix} c_2(b\,c_3 + c\,c_{34} + \frac{d}{2}\cos(\beta)) \\ b\,s_3 + c\,s_{34} + \frac{d}{2}\sin(\beta) \\ d_1 + s_2(-\frac{d}{2}\cos(\beta) - b\,c_3 - c\,c_{34}) \end{bmatrix}$ express the displacement between the frame {0} and the end effector along the axis $[x_0, y_0, z_0]$.

Assuming the following range of values for d_1, q_2, q_3, q_4, the workspace of the double parallelogram manipulator is shown in Figure 32: $d_1 = 0, q_2 = [0{,}360°], q_3 = (30°, 140°), q_4 = (-90° - q_3, 20° - q_3)$.

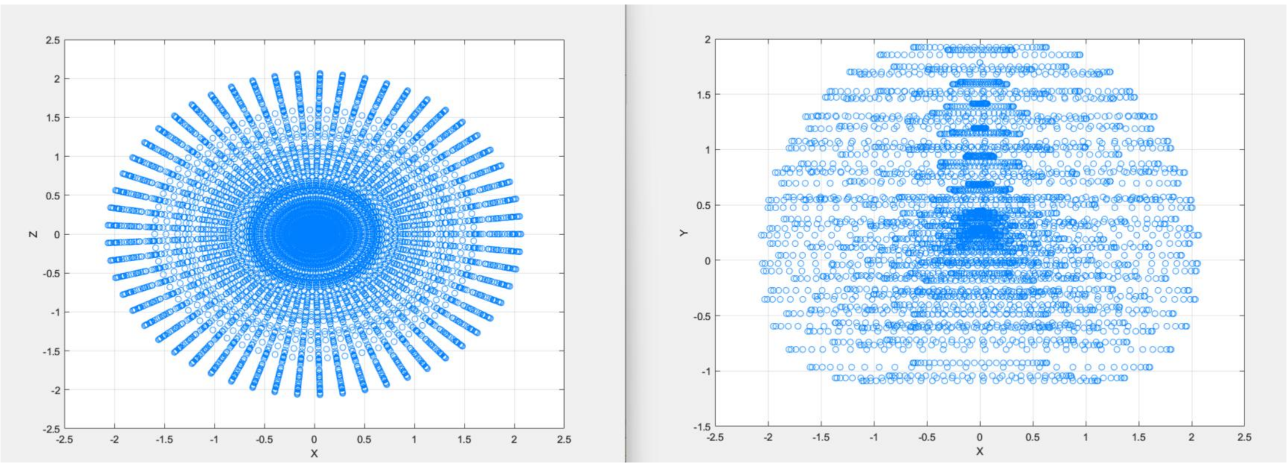


Figure 32 Workspace (in meters) of the 4DOF double parallelogram manipulator considering $d_1 = 0$. Top view (left), front view (right).

The transformation matrix T_e^0 express the displacement and the orientation of the end effector with respect to the frame {0}, which –however– is still attached to the mobile platform. Since the mobile platform might have positioning errors, it has been decided to calculate the position and the orientation of the end effector with respect to a frame located on the axis of the tube, in order to take into account the positioning errors along the z axis and around y. From the representation in Figure 33 of the frames {f} and {0}, we compute the transformation matrix T_0^f.

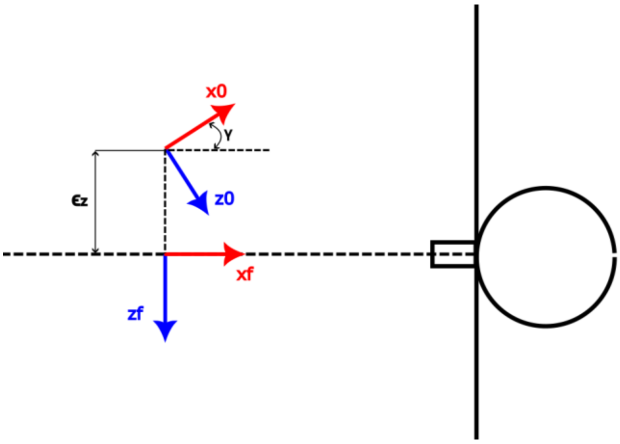


Figure 33 Positioning errors along z and around y.

$$T_0^f = \begin{bmatrix} \cos(\gamma) & 0 & \sin(\gamma) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\gamma) & 0 & \cos(\gamma) & \epsilon_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{26}$$

with γ representing the positioning error around y_f and $\epsilon_z$representing the positioning error along z_f.

The transformation matrix T_e^f= $T_0^f T_1^0 T_2^1\ T_3^2\ T_4^3 T_e^4$:

$T_e^f\,(d_1, q_1, q_2, q_3) =$

$$\begin{bmatrix} \cos\,(\gamma+q_2) & 0 & \sin\,(\gamma+q_2) & d_1 s_\gamma + \cos(\gamma+q_2)\cdot(b\ c_3 + c\ c_{34} + \frac{d}{2}\ \cos(\beta)) \\ 0 & 1 & 0 & b\ \mathrm{s}_3 + c\ \mathrm{s}_{34} + \frac{d}{2}\sin(\beta) \\ -\sin\,(\gamma+q_2) & 0 & \cos\,(\gamma+q_2) & \epsilon_z + d_1 c_\gamma + \sin\,(\gamma+q_2)\cdot(-\frac{d}{2}\cos(\beta) - b\ \mathrm{c}_3\ - c\ \mathrm{c}_{34}) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(27)

The transformation matrix (27) express the orentation and the displacement of the end effector with respect to a frame {f} with the origin along the axis of the tap, taking into account the positioning errors ϵ_z and γ. Note that if $\epsilon_z = 0$ and $\gamma = 0$, if follows that $T_e^f\ = T_e^0$.

4.2 Inverse Kinematics

To solve the inverse kinematic problem, an analytical solution is presented. As previously stated, joint 1 and joint 2 are used to correct positioning errors around y_f and along z_f, while joints 3 and 4 allow the manipulator to extend to reach distances along x_f and y_f. In order to solve the positioning errors, we start by expressing d_1 and q_2 so that the origin of frame {e} is positioned on the x_f axis and its orientation is the same as frame f.

The orientation between frame {f} and frame {e} are equivalent when the rotation matrix R_e^f coincides with the identity matrix :

$R_e^f\,(q_2) = I_3$,

$$\begin{bmatrix} \cos(\gamma+q_2) & 0 & \sin(\gamma+q_2) \\ 0 & 1 & 0 \\ -\sin(\gamma+q_2) & 0 & \cos(\gamma+q_2) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The equality is reached with:

$$q_2 = -\gamma. \tag{28}$$

The trasformation matrix T_e^f become:

$$T_e^f(d_1, -\gamma, q_3, q_4) = \begin{bmatrix} 1 & 0 & 0 & d_1 s_\gamma + (b\, c_3 + c\, c_{34} + \frac{d}{2}\cos(\beta)) \\ 0 & 1 & 0 & b\,\mathrm{s}_3 + c\,\mathrm{s}_{34} + \frac{d}{2}\sin(\beta) \\ 0 & 0 & 1 & \epsilon_z + d_1 c_\gamma \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The variable d_1 shall compensate the positioning error along the z_f axis. To collocate the end effector of the manipulator along the axis x_f, the displacement vector t_e^f:

$t_e^f(d_1)[3] = 0,$

$\epsilon_z + d_1\cos(\gamma) = 0,$

$$d_1 = -\frac{\epsilon_z}{\cos(\gamma)}. \tag{29}$$

The trasformation matrix T_e^f become:

$$T_e^f\left(-\frac{\epsilon_z}{\cos(\gamma)}, -\gamma, q_3, q_4\right) = \begin{bmatrix} 1 & 0 & 0 & -\frac{\epsilon_z}{\cos(\gamma)}\sin(\gamma) + (b\, c_3 + c\, c_{34} + \frac{d}{2}\cos(\beta)\,) \\ 0 & 1 & 0 & b\,\mathrm{s}_3 + c\,\mathrm{s}_{34} + \frac{d}{2}\sin(\beta) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The terms $-\frac{\epsilon_z}{\cos(\gamma)}\sin(\gamma)$ express the displacement caused by the joint d_1 between the base of the manipulator and the frame 0 attached on the mobile platform when both of the mentioned errors occurs. However, once the frame {e} has the correct orentation and the correct position along z_f, we can compute the values of q_3 and q_4 needed to reach the tap with respect to the base of the manipulator.

$$x_e = b\cos(q_3) + c\cos(q_3 + q_4) + \frac{d}{2}\cos(\beta) \tag{30}$$

$$y_e = b\sin(q_3) + c\sin(q_3 + q_4) + \frac{d}{2}\sin(\beta) \tag{31}$$

From the (30) and (31), we compute the (32) and (33):

$x_e - \frac{d}{2}\cos(\beta) = b\cos(q_3) + c\cos(q_3 + q_4),$

$y_e - \frac{d}{2}\sin(\beta) = \text{ b} \sin(q_3) + c \sin(q_3 + q_4)$,

$$\left(x_e - \frac{d}{2}\cos(\beta)\right)^2 + \left(y_e - \frac{d}{2}\sin(\beta)\right)^2 = (b \text{ c}_3 + c\, c_{34})^2 + (b\, s_3 + c\, s_{34})^2 \text{ ,}$$

$$x_e^2 + y_e^2 + \frac{d^2}{4} - 2x_e \frac{d}{2}\cos(\beta) - 2y_e \frac{d}{2}\sin(\beta) = b^2 + c^2 + 2bc\cos(q_4),$$

$$\cos(q_4) = \frac{(x_e^2 + y_e^2 + \frac{d^2}{4} - 2x_e \frac{d}{2}\cos(\beta) - 2y_e \frac{d}{2}\sin(\beta) - b^2 - c^2)}{2\, b\, c}:$$

$$q_4 = \pm\cos^{-1}\left(\frac{(x_e^2 + y_e^2 + \frac{d^2}{4} - 2x_e \frac{d}{2}\cos(\beta) - 2y_e \frac{d}{2}\sin(\beta) - b^2 - c^2)}{2\, b\, c}\right) \quad (32)$$

$$q_3 = \tan^{-1}\left(\frac{y_e - \frac{d}{2}\sin(\beta)}{x_e - \frac{d}{2}\cos(\beta)}\right) - \tan^{-1}\left(\frac{c\sin(q_4)}{b + c\cos(q_4)}\right) \quad (33)$$

The relations (32) and (33) admit two solutions, however, the allowed values for the designed configuration are $q_3 = (-12.1°, 168°), q_4 = (-125° - q_3, 55° - q_3)$.

In Table 9 are resumed the inverse kinematic expressions for the joint's variables d_1, q_2, q_3, q_4.

Table 7 Joints variable, inverse kinematics expression, range of values

d_1	$d_1(\epsilon_z, \gamma) = -\frac{\epsilon_z}{\cos(\gamma)}$	$d_1 \in [-0.16m, 0.16m]$
q_2	$q_2(\gamma) = -\gamma,\ q_2(\gamma) = -\gamma + \pi$	$q_2 \in [0{,}360°]$
q_3	$q_3 = \tan^{-1}\left(\frac{y_e - \frac{d}{2}\sin(\beta)}{x_e - \frac{d}{2}\cos(\beta)}\right) - \tan^{-1}\left(\frac{c\sin(q_4)}{b + c\cos(q_4)}\right)$	$q_3 \in (-12.1°, 168°)$
q_4	$q_4 = \pm\cos^{-1}\left(\frac{(x_e^2 + y_e^2 + \frac{d^2}{4} - 2x_e \frac{d}{2}\cos(\beta) - 2y_e \frac{d}{2}\sin(\beta) - b^2 - c^2)}{2\, b\, c}\right)$	$q_4 \in (-125° - q_3, 55° - q_3)$

In the next lines, an example of the inverse kinematic problem is proposed. Given the positioning of the mobile platform as depicted in Figure 34, we want to compute the joints variables d_1, q_2, q_3, q_4, in order to reach two taps with an height $y = 0.7\,m$ with respect to the mobile platform. In the proposed scenario there is both a positioning error around y equal to $\gamma = -15°$ and a positioning error along z equal to $\epsilon_z = -0.15m$.

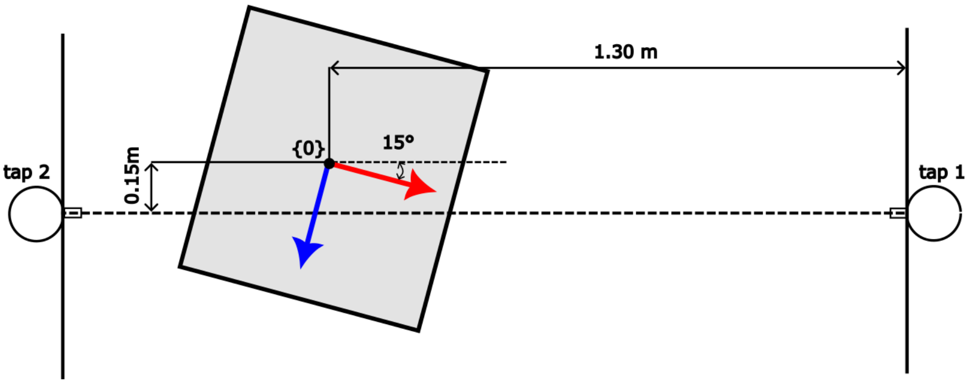


Figure 34 Positioning of the mobile platform with γ and ϵ_z.

We start by fixing the positioning errors. From the relations (28) and (29):

$$d1 = -\frac{\epsilon_z}{\cos(\gamma)} = \frac{0.15m}{\cos(-15°)} = 0.155m$$

$q_2 = -\gamma = 15°$ to reach the tap 1 and $q_2 = -\gamma + \pi = 195°$ for tap 2.

With d_1 and q_2 the base of the manipulator has both the correct position and the correct orientation.

The distances $L_i = (x, y)$ between the base of the manipulator and the tap i are:

$$L_1 = \left(\left(1.30 + \epsilon_z \frac{\sin(\gamma)}{\cos(\gamma)}\right) m, 0.70m\right) = (1.34m, 0.70m),$$

$$L_2 = \big((2 - 1.34)m, 0.70m\big) = (0.66m, 0.70m).$$

From the relations (32),(33) we compute the values of q_{3_i} and q_{4_i} for the tap i. Between the two solutions of q_{3_i} and q_{4_i}, we select the solution into the ranges expressed in Table 9. It follows that:

$q_{3_1} = 61°$

$q_{4_1} = -89°$

$q_{3_2} = 94°$

$q_{4_2} = -129°$

Table 8 Joints variables to reach tap 1 and tap 2.

	Tap 1	Tap 2
d_1	$0.155m$	$0.155m$
q_2	$15°$	$195°$
q_3	$61°$	$94°$
q_4	$-89°$	$-129°$

4.3 Dexterity analysis

To analyze the dexterity of the 4DOF manipulator, the coefficient k is introduced as:

$$k = \frac{\sigma_{max}}{\sigma_{min}}, \tag{34}$$

where σ_{max}, σ_{min} are the maximum and minimum singular value of the Jacobian matrix J, respectively. The coefficient k has values in $[1, +\infty]$. To convert these values to the range $[0,1]$, the $\frac{1}{k} = \frac{\sigma_{min}}{\sigma_{max}}$ is considered, where '0' means a low level of dexterity (singular configuration), while '1' means an high level of dexterity. The Jacobian of the manipulator:

$$J = \begin{bmatrix} 0 & -s_2\,(b\,c_3 + c\,c_{34} + \frac{d}{2}c_\beta) & -c_2\,(b\,s_3 + c\,s_{34} + \frac{d}{2}s_\beta) & -c_2\,(c\,s_{34} + \frac{d}{2}s_\beta) \\ 0 & 0 & b\,c_3 + c\,c_{34} + \frac{d}{2}c_\beta & c\,c_{34} + \frac{d}{2}c_\beta \\ 1 & -c_2\,(b\,c_3 + c\,c_{34} + \frac{d}{2}c_\beta) & s_2\,(b\,s_3 + c\,s_{34} + \frac{d}{2}s_\beta) & s_2\,(c\,s_{34} + \frac{d}{2}s_\beta) \\ 0 & 0 & s_2 & s_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & c_2 & c_2 \end{bmatrix}$$

(35)

Since we are interested in studying the dexterity in the positioning of the manipulator, only the translational part of the Jacobian is considered:

$$J_p = \begin{bmatrix} -s_2\left(b\,c_3 + c\,c_{34} + \frac{d}{2}c_\beta\right) & -c_2\left(b\,s_3 + c\,s_{34} + \frac{d}{2}s_\beta\right) & -c_2\left(c\,s_{34} + \frac{d}{2}s_\beta\right) \\ 0 & b\,c_3 + c\,c_{34} + \frac{d}{2}c_\beta & c\,c_{34} + \frac{d}{2}c_\beta \\ -c_2\left(b\,c_3 + c\,c_{34} + \frac{d}{2}c_\beta\right) & s_2\left(b\,s_3 + c\,s_{34} + \frac{d}{2}s_\beta\right) & s_2\left(c\,s_{34} + \frac{d}{2}s_\beta\right) \end{bmatrix} \tag{36}$$

Using Matlab, the value of $\frac{1}{k}$ is iteratively computed within the manipulator workspace. The results are shown in the colormap in Figure 35, where the white-yellow positions indicate values close to 1, while the darkest colors indicate values close to 0.

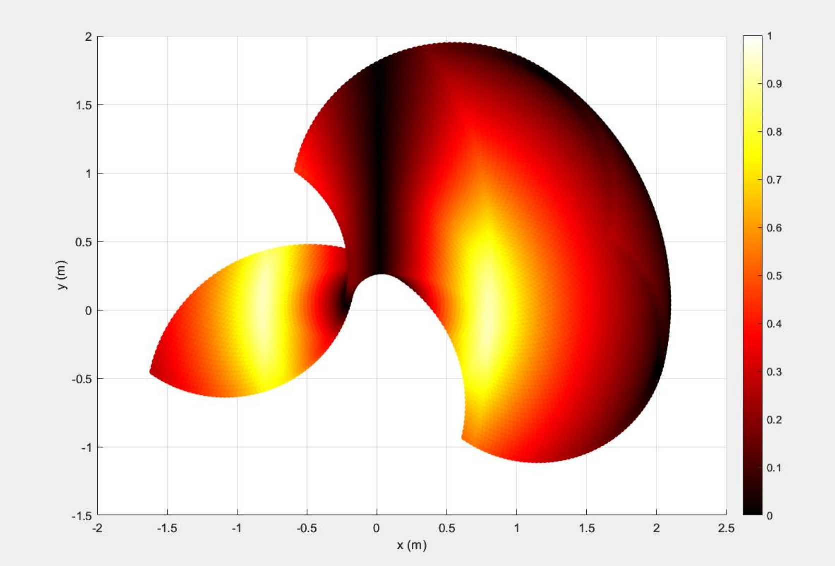


Figure 35 Dexterity of the manipulator expressed through the coefficient $\frac{1}{k}$.

Chapter 4

5 Mechanical Design

This chapter addresses the mechanical design of the robotic arm with a double parallelogram configuration. Building on the calculations from Chapters 2 and 3, the goal is to design the manipulator by selecting commercial components and custom-designing specific parts that can be 3D printed. This phase was conducted using SolidWorks.

To simplify the drafting of the chapter, we will divide the structure of the manipulator into 5 macro areas, as illustrated in Figure 36:

1) Joint 2, which contains the active joint J2,
2) Joint 3, which consists of the active joint J3 and the passive joint J3′,
3) Joint 4, which includes the active joint J4 and the passive joints J4′ and J4′′,
4) Joint 5, which contains the passive joints J5 and J5′,
5) Links, which comprise links b, b′, c, and c′.

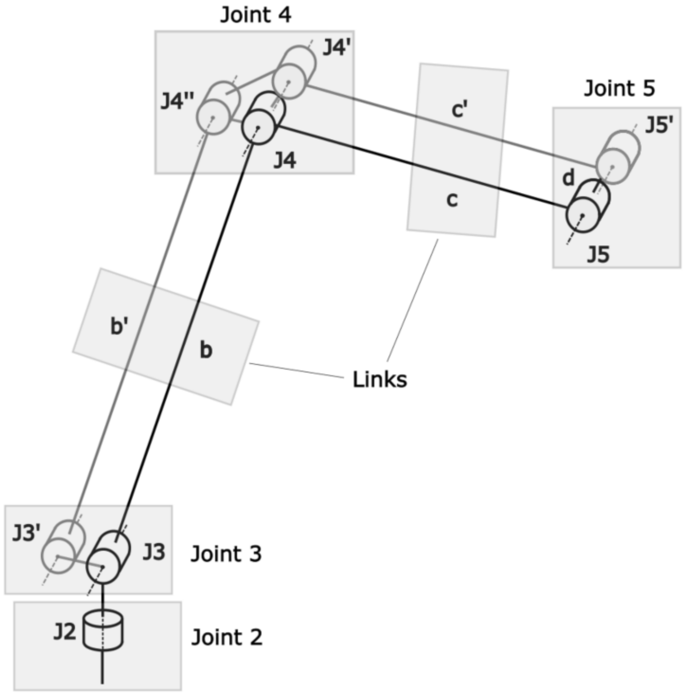
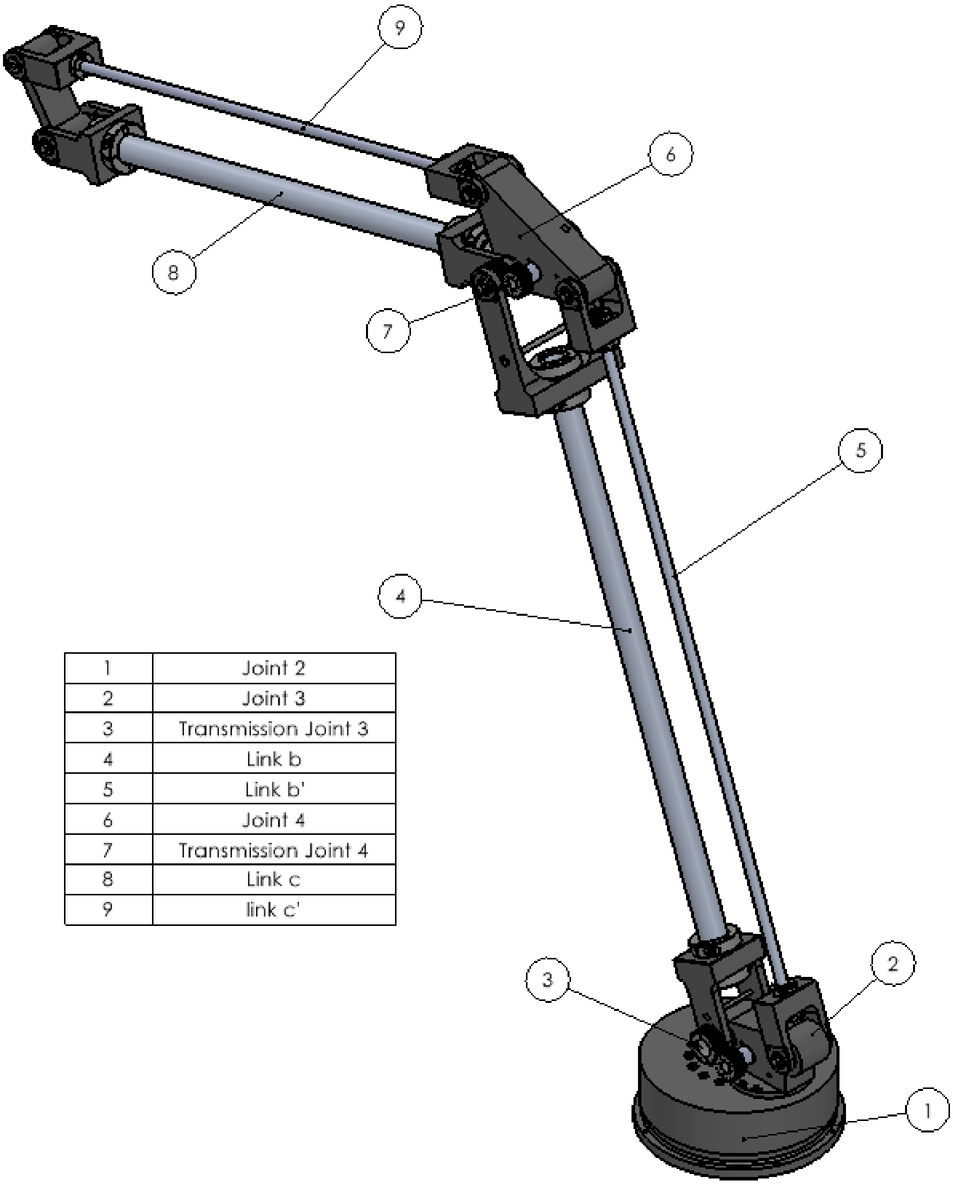


Figure 36 Division of the manipulator in macro areas.

For the construction of the hinges, pins and bearings will be used. Components for the joints 3 and 4 will be designed to serve both as housing for the motors and as support for the pins, while the bearings will be mounted on the links. For joints 3 and 4, a transmission with gears is employed. Each link is designed with two forks and a tube.



1	Joint 2
2	Joint 3
3	Transmission Joint 3
4	Link b
5	Link b'
6	Joint 4
7	Transmission Joint 4
8	Link c
9	link c'

Figure 37 3D view of the designed manipulator.

5.1 Motors

For the project, three actuators are required to drive the active joints of the robotic arm. The selection is between two models of electric motors for robotic applications available from the company 'ROBOTIS.' The actuators belong to the DINAMIXEL X series, specifically the models XH540W150-T and XH540-W270-T. While the two models share the same geometry, they differ in specifications, which are detailed in Table 10.

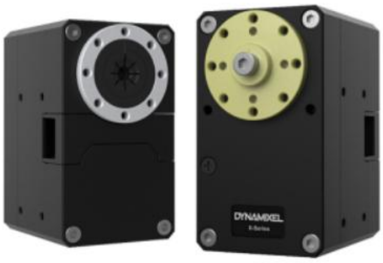


Figure 38 Dynamixel XH540-W270-T.

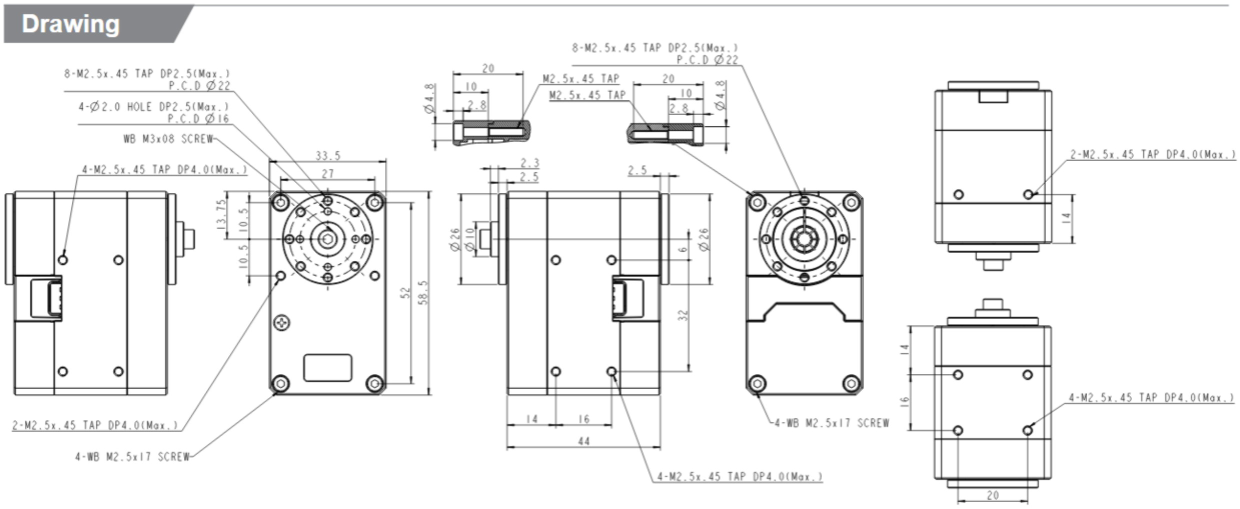


Figure 39 Drawing of the Dynamixel XH540-W150-T and XH540-W270-T.

Table 9 Specifications of Dynamixel XH540-W150-T and XH540-W270-T.

Properties	XH540-W150-T	XH540-W270-T
MCU	ARM CORTEX-M3 (72 [MHz], 32Bit)	ARM CORTEX-M3 (72 [MHz], 32Bit)
Weight	165 g	165 g
Gear ratio	152.3 : 1	272.5 :1
Stall Torque	7.1 Nm	9.9 Nm
Input voltage	12 V	12V
No load speed	70 rpm	39 rpm
Radial load	40 N (10 mm away from the horn)	40 N (10 mm away from the horn)
Axial load	20 N	20 N

Since the dimensions of the two models are identical, we will not specify which model to assign to a particular joint in the initial phase of the design. Given the required dimensions and loads, it can be assumed that attaching the links directly to the motor may not provide sufficient radial load capacity, as 40N might be inadequate. Therefore, a transmission system will be considered for actuation.

5.2 Joint 2

For the design of the Joint $J2$, a slewing ring with a toothed outer ring made of alluminium is selected. The slewing ring facilitates relative rotation between a fixed and a movable component, on which the robotic arm will be mounted, and it will be driven by a gear connected to the motor. Therefore, the initial focus will be on the transmission for Joint 2. The dimensions of the slewing ring are provided in Table 11 based on the drawing supplied by the vendor (Igus), shown in Figure 40. The mechanical properties are listed in Table 12.

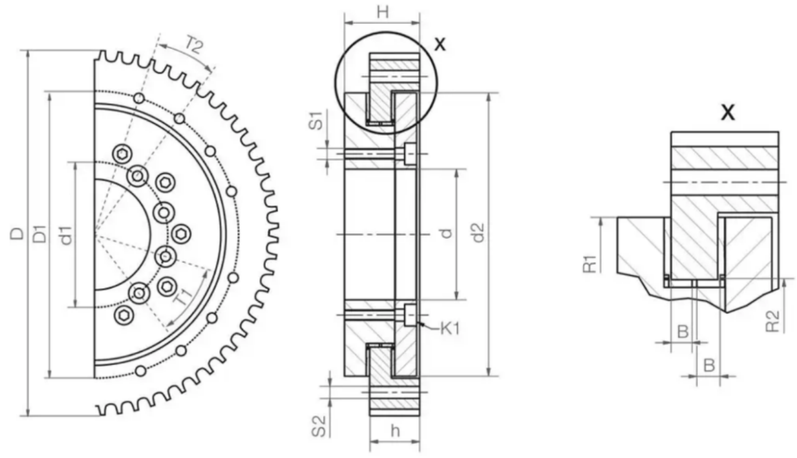


Figure 40 Drawing of the slewing ring.

Table 10 Dimensions of the slewing ring.

R1	60mm	**K1**	ISO 4762 M6
d1	65mm	**T1**	8x45°
	160mm	**T2**	16x22.5°
D1	135mm	**R2**	46.5mm
S1	M6	**S2**	6.6mm
d	50mm	**d2**	120mm
h	23mm	**Module**	2
B	4.5mm	**N of teeth**	78

Table 11 Specifications of the slewing ring.

Weight	1227.8g
Axial load, static	40000N
Axial load, dynamic	10000N
Radial load, static	8000N
Radial load, dynamic	2500N
Rotating speed dry running	200 rpm
Max. permissible tilt moment	600 Nm

For the driving gear, a straight-toothed gear has been sized from the Misumi website, which specializes in customizable commercial components. The purchased gear will require machining to create holes for the screws needed to attach to the motor, in particular 8x $\phi 3$, P.C.D $\phi 22$.

In Table 13 are shown the dimensions and the specifications of the gear with a module of 2.

Table 12 Specifications and dimensions of the driving gear for Joint 2.

N of teeth	16
Material	EN 1.1191
B	20mm
d reference diameter	32mm
Tip diameter	36mm
Root diameter	27mm
Shaft bore	16mm
Allowable transmission force	31.73 Nm

The gear ratio for Joint 2 can be computed as $i_2 = \frac{78}{16} = 4.87$.

For the construction of the joint, two components have been designed: one that houses the motor and the fixed part of the slewing ring; and a cap that will be connected to the slewing ring's crown. This configuration enables the cap to rotate with the crown, thereby allowing the manipulator mounted on it to rotate as well. As can be seen in Figures 40 and 41, the components have a circular shape and feature holes that allow for the connections mentioned above, enabling fastening via screws. The six 5.5 mm diameter holes visible on the cap are used to connect the third joint of the manipulator, which will be discussed later. Additional holes have been added on both the cap (15 mm diameter) and the fixed part (20 mm diameter) with the purpose of creating channels to route the motor cables to the base.

Detail A in Figure 41 illustrates the hole for the motor housing. The rectangular opening has been made 1 mm longer on both sides (with respect to the motors dimensions). The motor is secured using screws that pass through the slots with a radius of 1.5 mm.

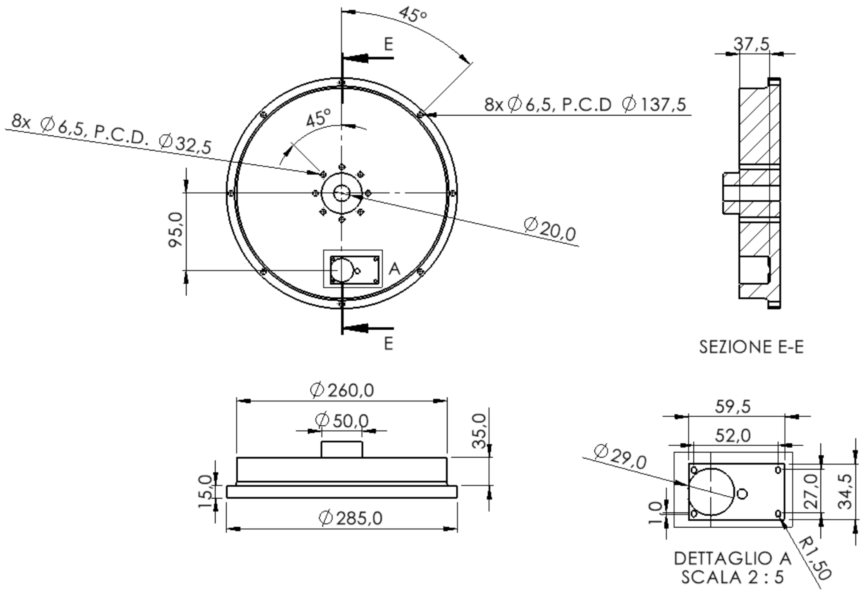


Figure 41 Joint 2, fixed component

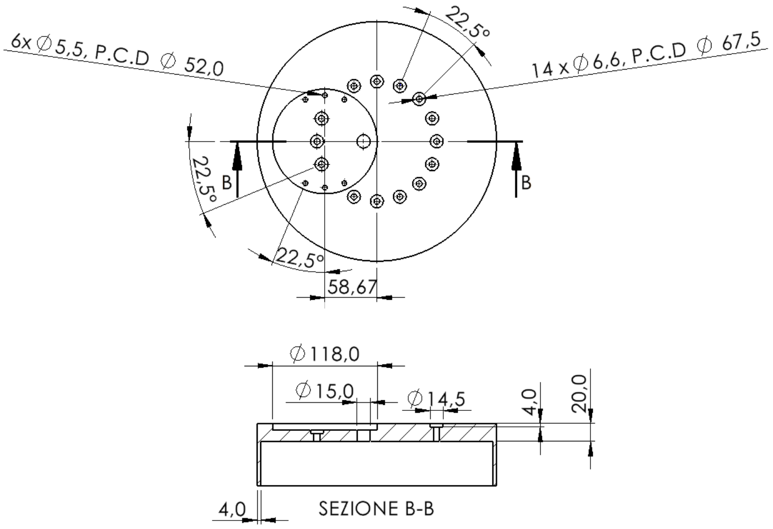


Figure 42 Joint 2, cap

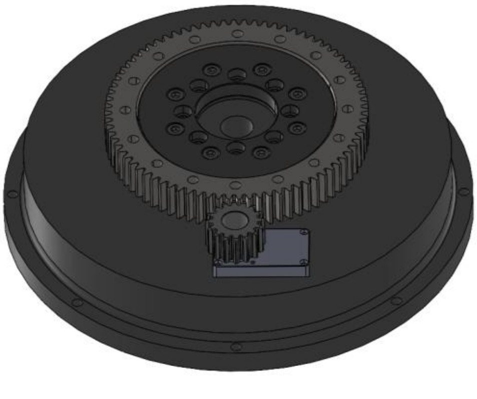


Figure 43 3D view of the fixed component with motor, slewing ring, and gear wheel.

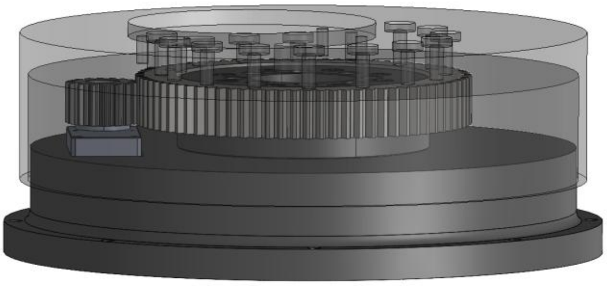


Figure 44 Joint 2 assembled with the cap shown in transparency.

5.3 Transmission for active joints J3 and J4

For the active joints J3 and J4, a transmission system has been designed using two straight-toothed gears: a driving gear to be connected to the motor, and a driven gear to be attached to the fork in links b and c. The transmission is identical for both joint 3 and joint 4 (The transmission could be eventually modified in the balancing phase). The gears are made of steel and have been selected from the Misumi catalog. The commercial components will require machining to create the holes necessary for connecting them to the motor and the fork. The gears have a module of 2, with their specifications and dimensions provided in Table 14.

Table 13 Dimensions and specifications of the gears, driving and driven.

	Driving gear	**Driven gear**
N of teeth	20	24
Material	EN 1.1191	EN 1.1191
B	20mm	20mm
d reference diameter	40mm	48mm
Tip diameter	44mm	52mm
Root diameter	35mm	43mm

Shaft bore	17mm	26mm
Allowable transmission force	44.59 Nm	57.96 Nm

The gear ratio for the transmissions of joints J3 and J4: $i_3 = i_4 = \frac{24}{20} = 1.2$.

The center distance between the gears is 44mm. In Figure 45 is depicted the drawing of the transmission for joint 3 and 4, including the required holes to be machined.

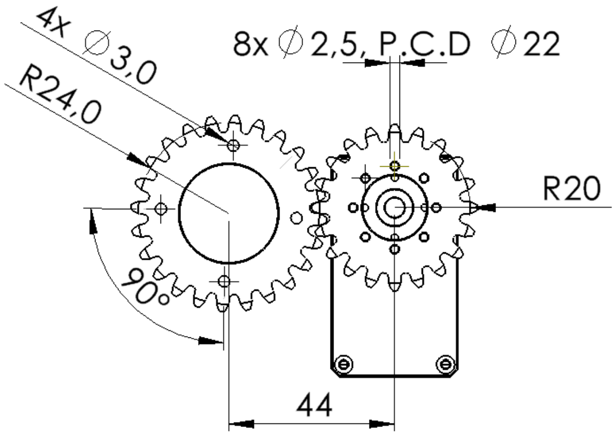


Figure 45 Drawing of the transmission of Joints 3 and 4.

In Figures 46 and 47 are shown the transmissions for joint 3 and joint 4 assembled. A circular component has been designed to align the driven wheel attached to the fork with the driving wheel and is positioned between the engine and the driving wheel.

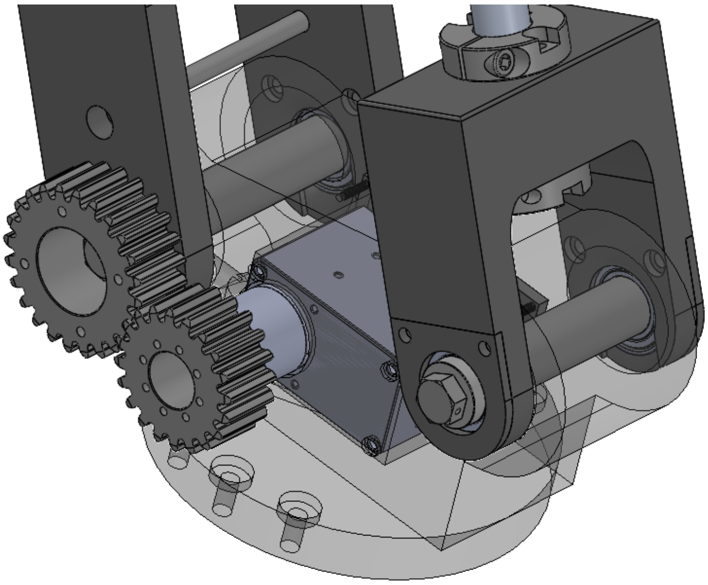


Figure 46 3D view of the transmission of joint 3.

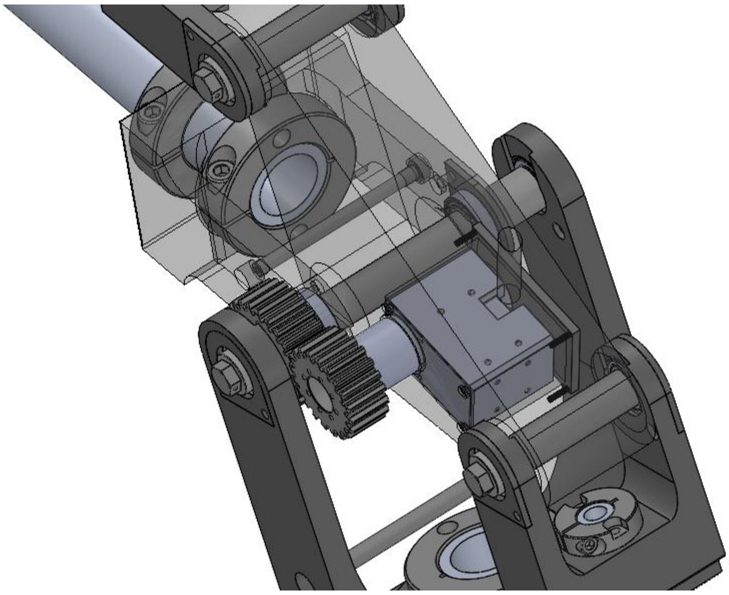


Figure 47 3D view of the transmission of joint 4.

5.4 Links

For the construction of links b, b', c, and c', a combination of commercial tubes and custom-designed 3D-printed forks was selected. The forks will be equipped with bearings to enable rotation around the joints, which are fitted with pins. The links connected to the active joints (links b and c) will be made using tubes with an outer diameter of 40mm, while the links connected to the passive joints (b' and c') will have a diameter of 16mm, as their primary function will be to maintain the orientation of the double-parallelogram system.

The chosen bearing is a deep groove ball bearing from SKF model 61903-2Z, which dimension and specifications are shown in Table 15.

Table 14 Specifications and dimensions of bearings.

Bore diameter	17 mm
Outside diameter	30 mm
Width	7 mm
Basic dynamic load rating	4.62 kN
Basic static load rating	2.55 kN
Reference speed	50000 r/min
Weight	0.0154 kg

Before the designing phase, the method for connecting the forks to the tubes was selected. Steel set collars, provided by the Misumi website, were chosen for this purpose. These collars will be attached to the fork to secure the tubes within the inner ring. Two different dimensions are selected, one for the tube $\phi 40mm$ and the other for the tube $\phi 16\ mm$.

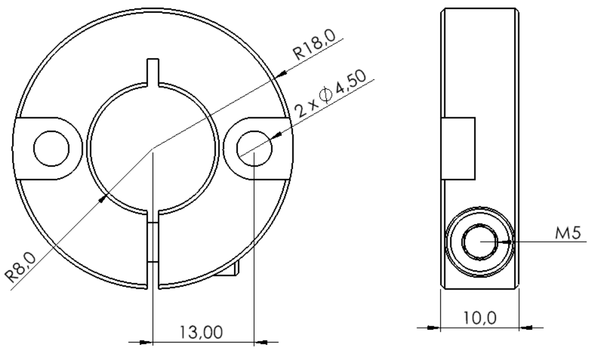


Figure 48 Set collar for tube $\phi16$.

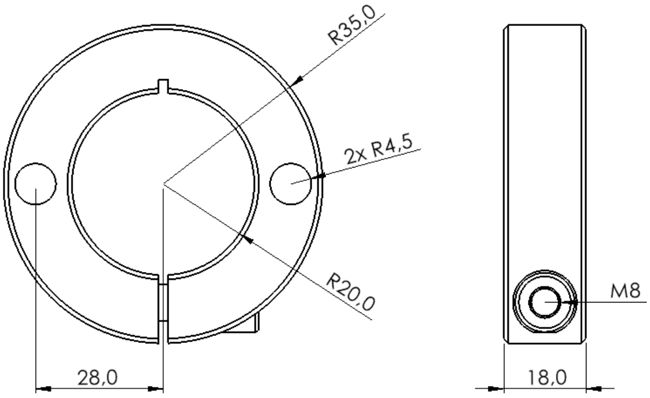


Figure 49 set collar for tube $\phi40$.

Five different types of forks have been designed:

1) Forks designed to connect links b′ and c′ to passive joints.
2) Fork to connect link b to active joint J3.
3) Fork to connect link c to active joint J4.
4) Fork to connect link b to active joint J4 without affecting the joint's actuation.
5) Fork to connect link c to passive joint J5.

For simplicity, the forks are named based on the letter of the link they belong to and the number of the joint they will be connected to. For example, fork b-3 connects the link b to joint J3.

Figure 50 presents a 3D view of the fork b-3. Before presenting the drawings and details of each fork, the general characteristics common to all of them will be explained. All the forks are equipped with two retaining rings for the tubes to improve their clamping. The bearings have been axially constrained using covers designed to be fastened with screws. All forks that will be paired with actuated joints have been equipped with stop pins, both to increase their strength and to create mechanical end-stops. Only the two forks responsible for actuation have the holes for mounting the driven gear wheel (Fork b-3 and fork c-4).

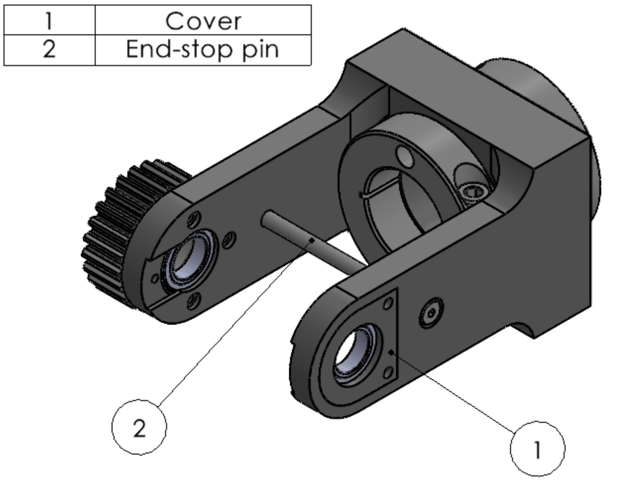


Figure 50 3D views of fork b-3.

5.4.1 Link b

For link b, two different forks have been designed: one that connects the tube to joint 3, where the gear wheel for actuation will be mounted, and the other that connects the link to joint 4, which will only allow passive rotation. The drawings are shown in Figures 51 and 52.

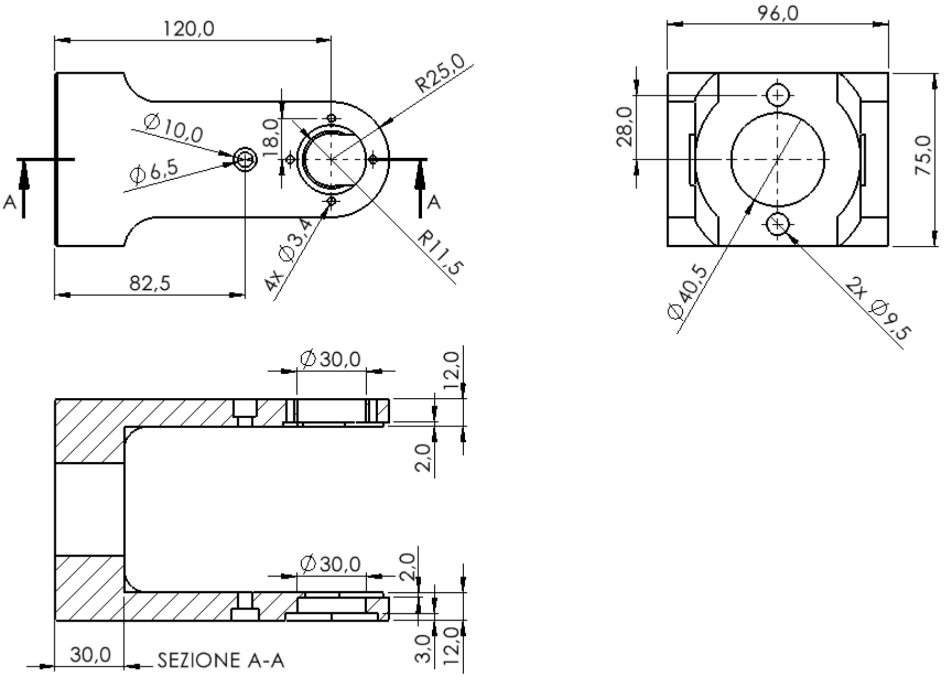


Figure 51 Fork b-3.

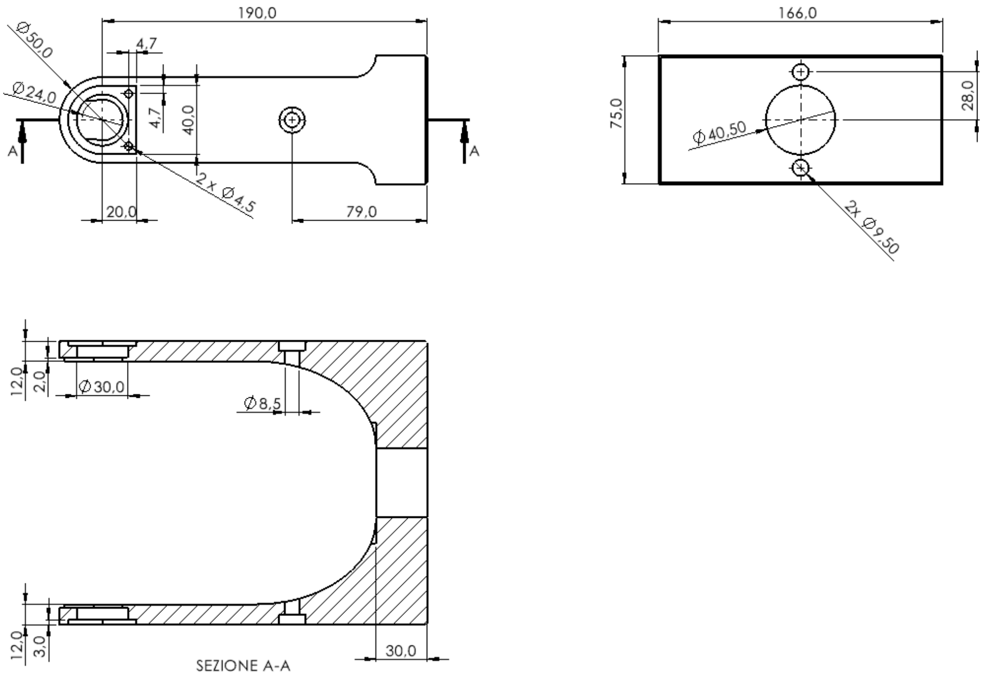


Figure 52 Fork b-4.

Based on the sizing calculations performed in Chapter 2, the length of link b was selected to be equal to 1140mm. We will now proceed to calculate the required tube length. The general relation:

$$l_t = l - l_{f1} - l_{f2} + t_{f1} + t_{f_2} + 2\,B, \tag{37}$$

where l_t is the length of the tube, l the link length, l_f is the length of the fork starting from the axis of the bearing for fork 1 and fork 2, t_f is the thickness of the fork (the distance between the two collars), and B is the collars width.

The length of the tube for link b:

$lt = 1140 - 120 - 190 + 30 + 30 + 2(18) = 926mm.$

5.4.2 Link c

Two different forks have also been designed for link c, with the drawings shown in Figures 53 and 54. We calculated the length of the tube for link c, knowing that the total length of link c shall be equal to 900mm. From the relation (37), $l_t = 771\ mm.$

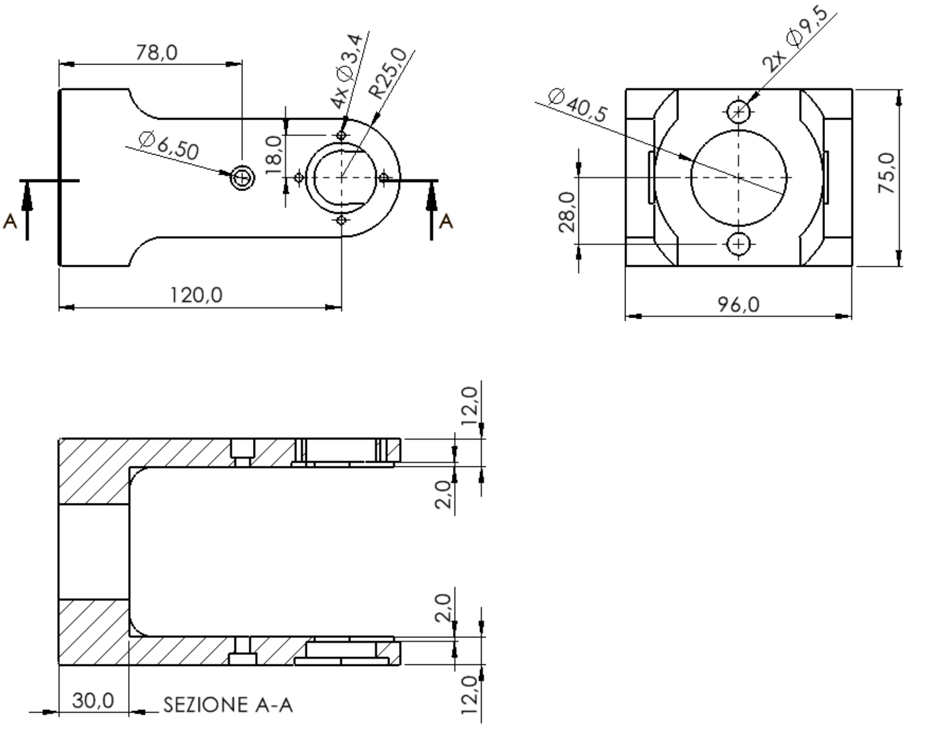


Figure 53 Fork c-4.

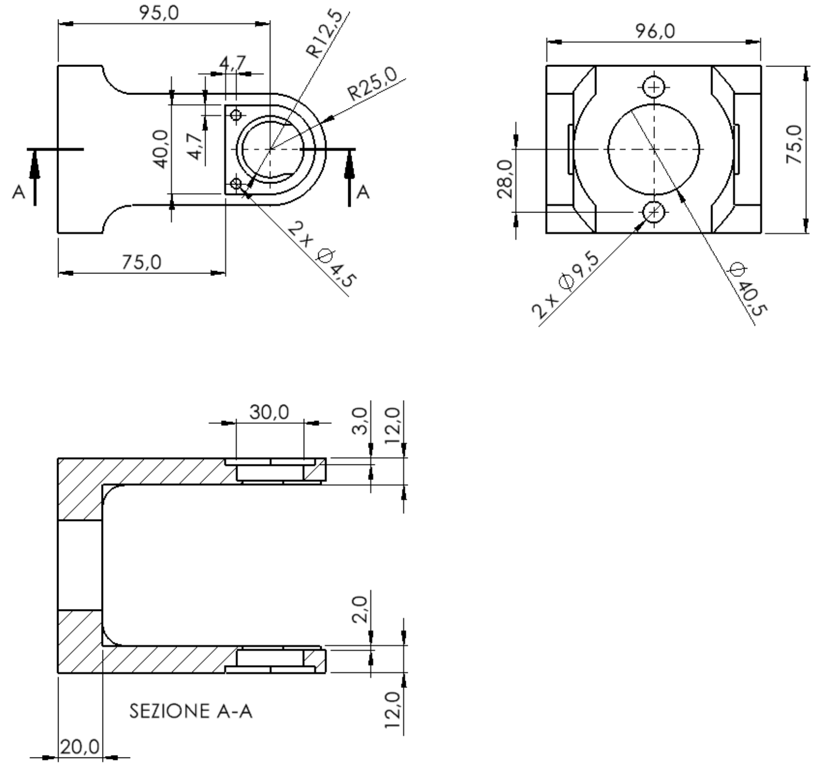


Figure 54 Fork c-5.

5.4.3 Link b′ - c′

A single type of fork has been designed for links b' and c', to be connected at both ends of the tube.

From equation (37), we then calculate the tube lengths, which are $l_t = 1010mm$ and $l_t = 770mm$ for links b' and c', respectively.

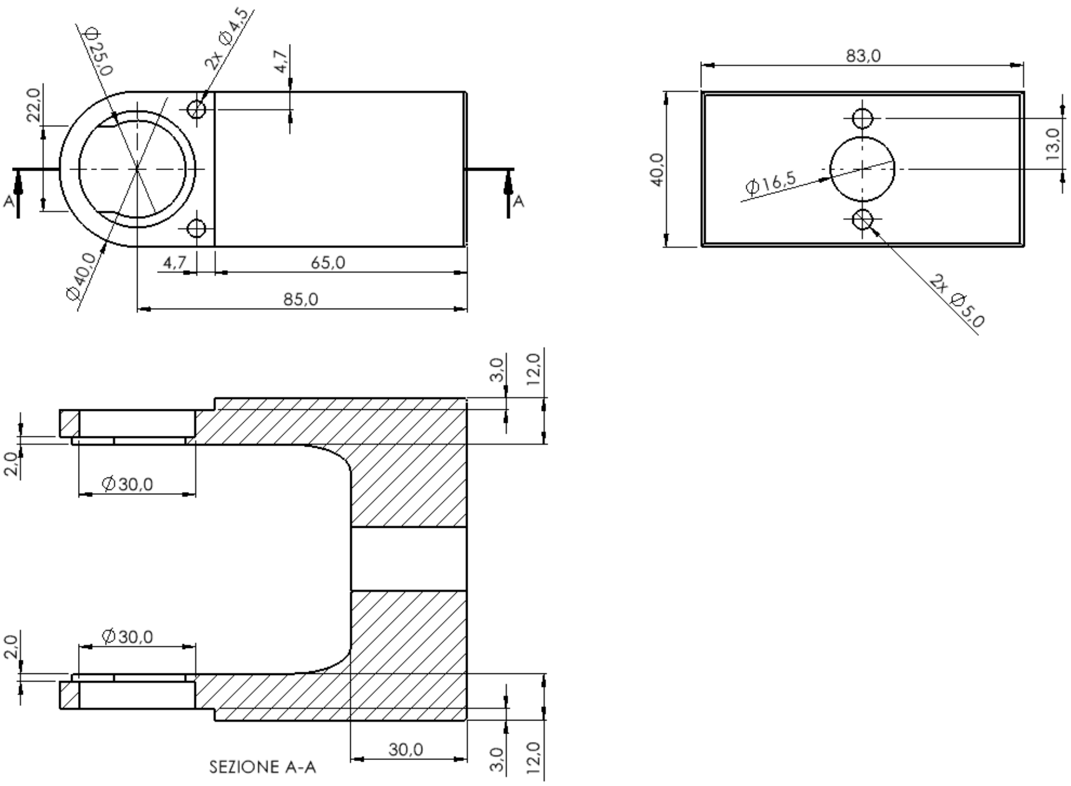


Figure 55 Drawing of the forks for passive joints.

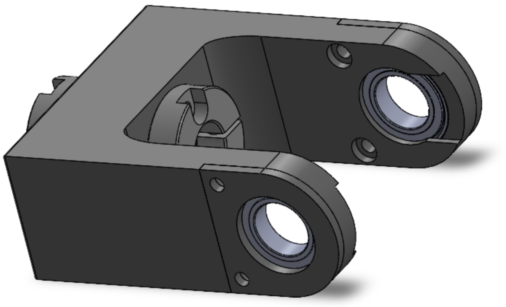
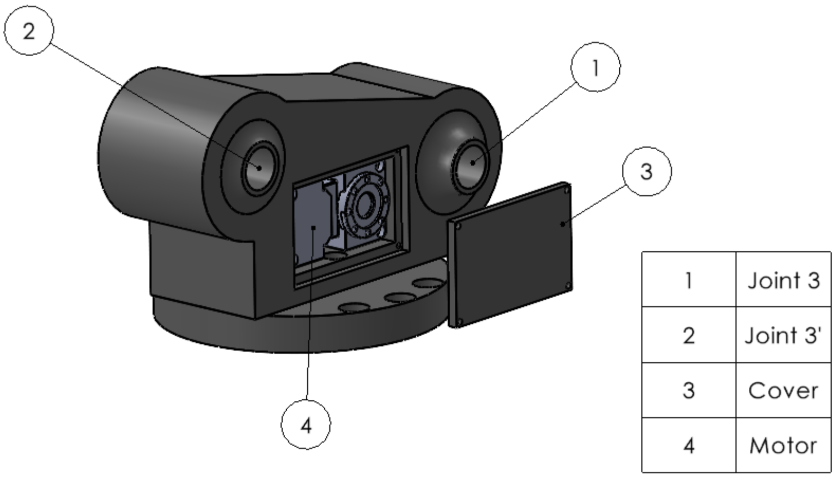


Figure 56 3D view of the forks for passive joints.

5.5 Fixed components for pin housing

Once the links and transmissions have been defined, the next step is to design the components that house the motor and support the hinge pins. Two components have been designed in SolidWorks: one for joints J3 and J3', and another for joints J4, J4', and J4''. The 3D view in Figure 57 illustrates the main features of these component: the holes to accommodate the pins, the housing for the motor, and the cover to close it. Holes have been included in the components for cable routing.



1	Joint 3
2	Joint 3'
3	Cover
4	Motor

Figure 57 Main features of the fixed component for pin housing.

The pins were sized using the Misumi website, and all have a diameter of 17 mm. The sectional views of the hinges are shown in Figures 58 and 59. In the hinge with multiplicity 2 (active Joint 4), a spacer has been inserted to separate the inner rings of the two forks.

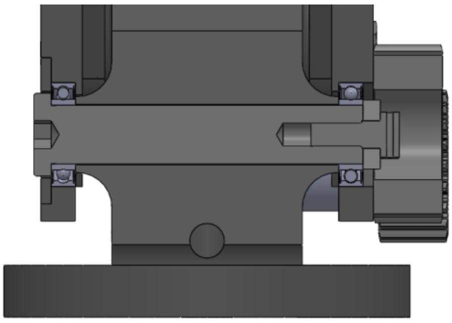


Figure 58 Joint 3 sectional view.

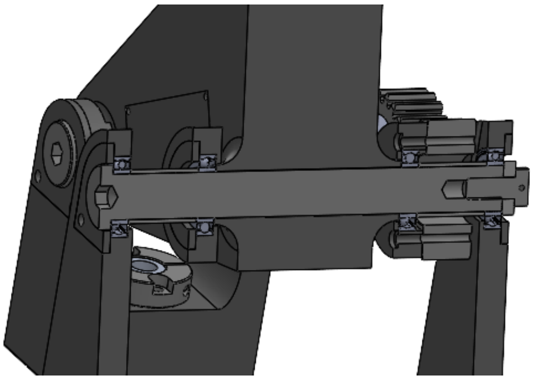


Figure 59 Joint 4 sectional view.

5.5.1 Joint 3

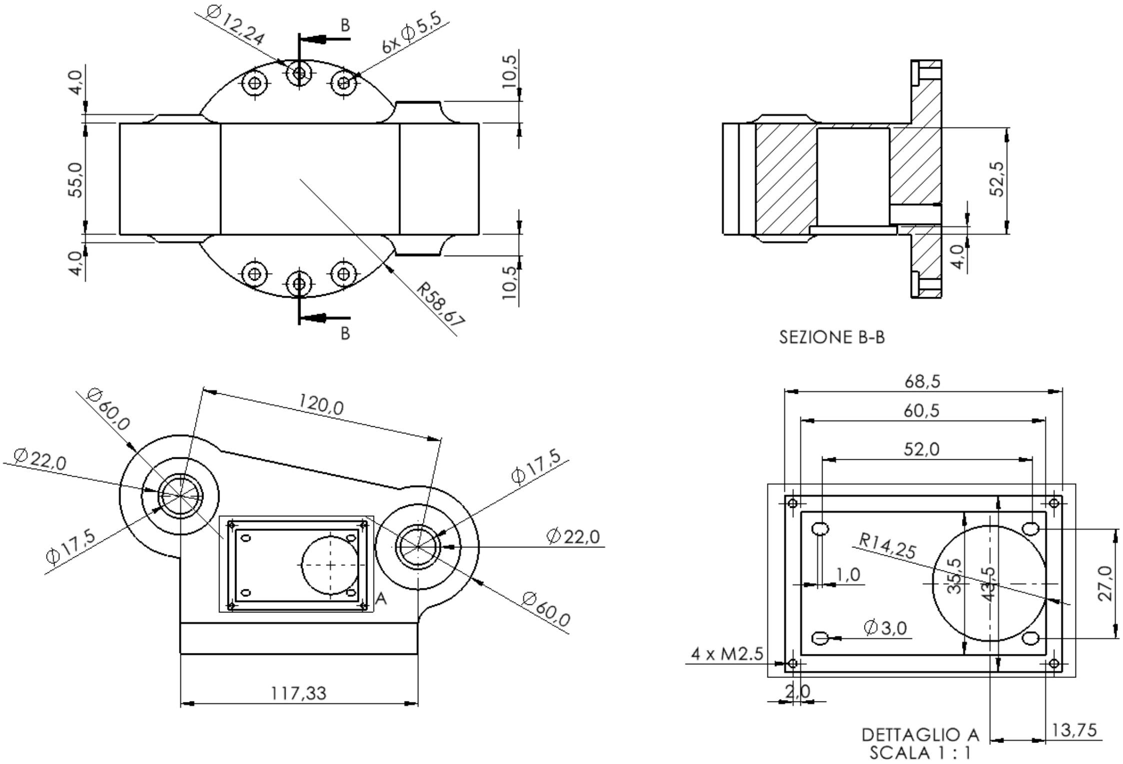


Figure 60 Joint 3 drawing.

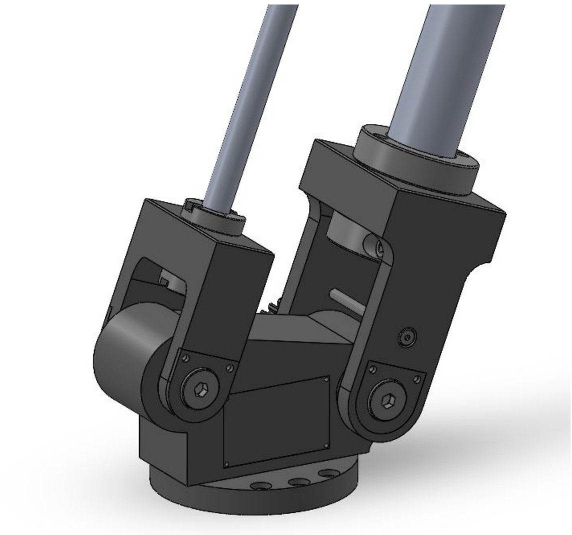


Figure 61 3D view of joints J3 and J3′ assembled.

5.5.2 Joint 4

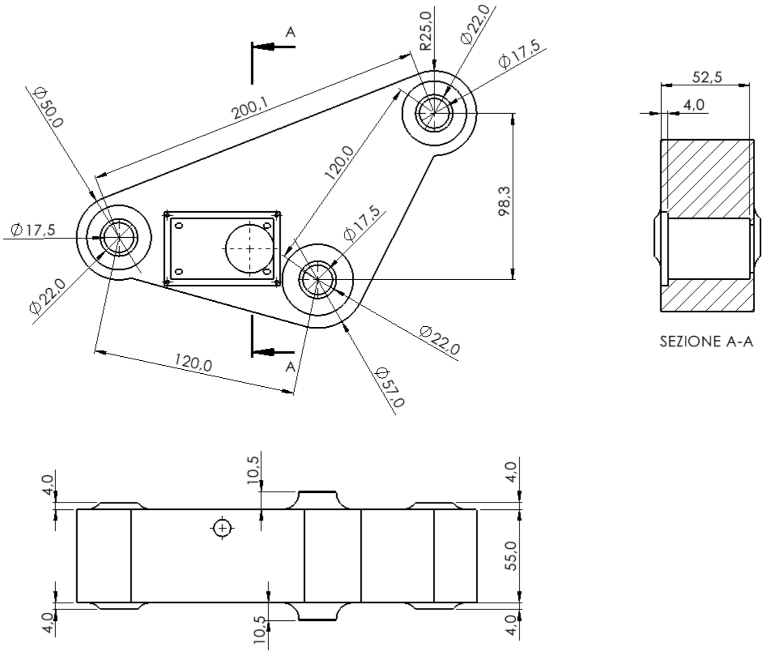


Figure 62 Joint 4 drawing.

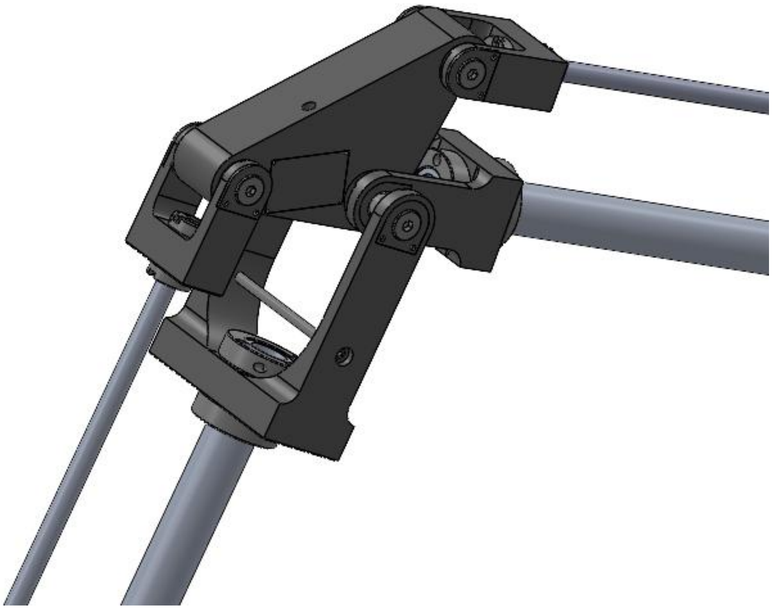


Figure 63 3D view of J4, J4’ and J4’’ assembled.

5.5.3 Joint 5

For the assembly of Joint 5, a component has been designed to passively connect links c and c' using pins, similar to the other passive joints. This component is intended to support the end effector. However, since the end effector has not yet been designed, no mounting holes have been included at this stage. The necessary holes can be added to the component in a subsequent phase.

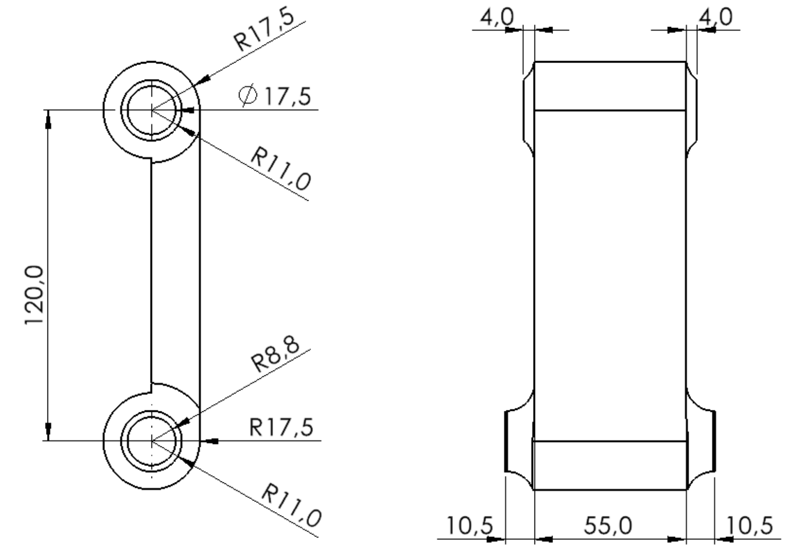


Figure 64 Drawing of Joint 5.

5.6 Operational space analysis with mechanical constraints

The table with the inverse kinematics solutions for angles q_3 and q_4, related to the operational space angles, is presented again. The aim is to verify that the designed manipulator can effectively reach the listed values, specifically ensuring that the end stops positioned on the forks allow full coverage of the operational space.

Table 15 Inverse kinematics solutions of the operational space from Chapter 2.

x	y	q_3	q_4	$q_3 + q_4$
0.5m	0.3m	78.7°	-153.1°	-74.4°
1.5m	1.1m	60.1°	-56.1°	4°
0.5m	1.1m	112.4°	-112.6°	-0.2°
1.5m	0.3m	46.9°	-87.1°	-40.2°

Using SolidWorks, we will mate the surfaces of the end stops with the surfaces they may come into contact with, in order to analyze the maximum and minimum values of q_3 and q_4 that the manipulator can reach. These values will then be compared with the maximum and minimum values listed in the table.

For angle q_3, the mechanical constraint is imposed by both the fork b-3 and the fork b'-3, which represents the maximum and the minimum values of q_3 respectively. The mechanical constraint for the minimum value of q_3 allows to reach an angle equal to 40°, thus the value of $q_3 = 46.9°$ to reach the point (1.5m,0.3m) can be satisfied.

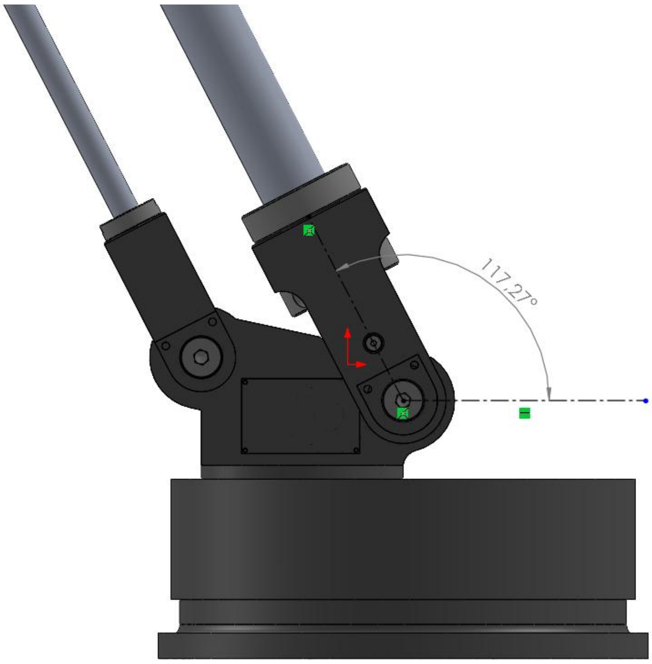


Figure 65 Maximum reachable angle $q_{3.}$

The maximum value of q_3 reachable by the manipulator is $q_3 = 117.27°$, which satisfies the angle of 112.4° calculated to reach the corner in (0.5m,1,1m).

The second mechanical constraint is imposed by the fork of link c, represented in Figure 65, that affects the maximum angle $q_3 + q_4$. In the operational space, the value required to reach the point (1.5m,1.1m) is equal to 4°. The manipulator is able to reach the maximum angle $q_3 + q_4 = 15.22°$.

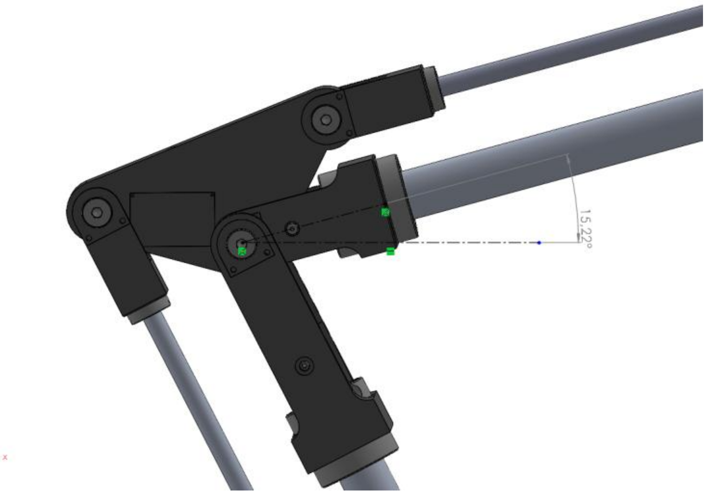


Figure 66 Maximum reachable angle $q_3 + q_4$.

The third constraint is due to a potential collision between the fork b-4 and the fork c-4 (Figure 66). The minimum q_4 angle is reached when the fork of link c rests against the end stop of the fork of link b. The minimum achievable angle is -158°, which exceeds the value listed in the table equal to

-153.1°.

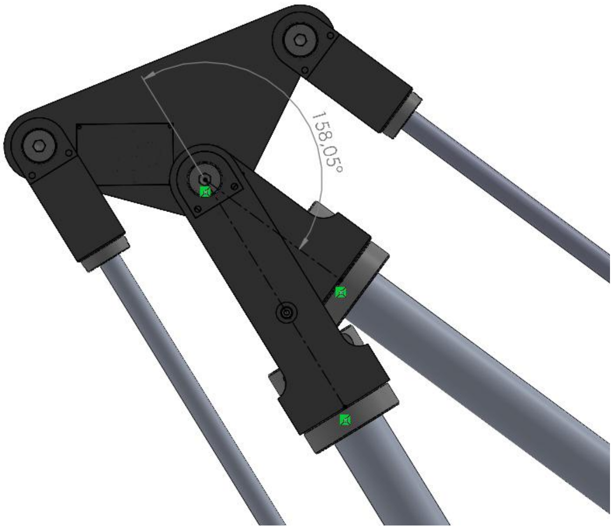


Figure 67 Maximum reachable angle q_4.

According to these calculations, we expect the manipulator to be capable of reaching all positions within the operational space. The constraints are expressed by the relation (38), (39) and (40).

$$40° \leq q_3 \leq 117.27°, \tag{38}$$

$$q_4 \geq -158°, \tag{39}$$

$$q_3 + q_4 \leq 15.22° \,. \tag{40}$$

The workspace of the designed manipulator with mechanical constraints is shown in Figure 68:

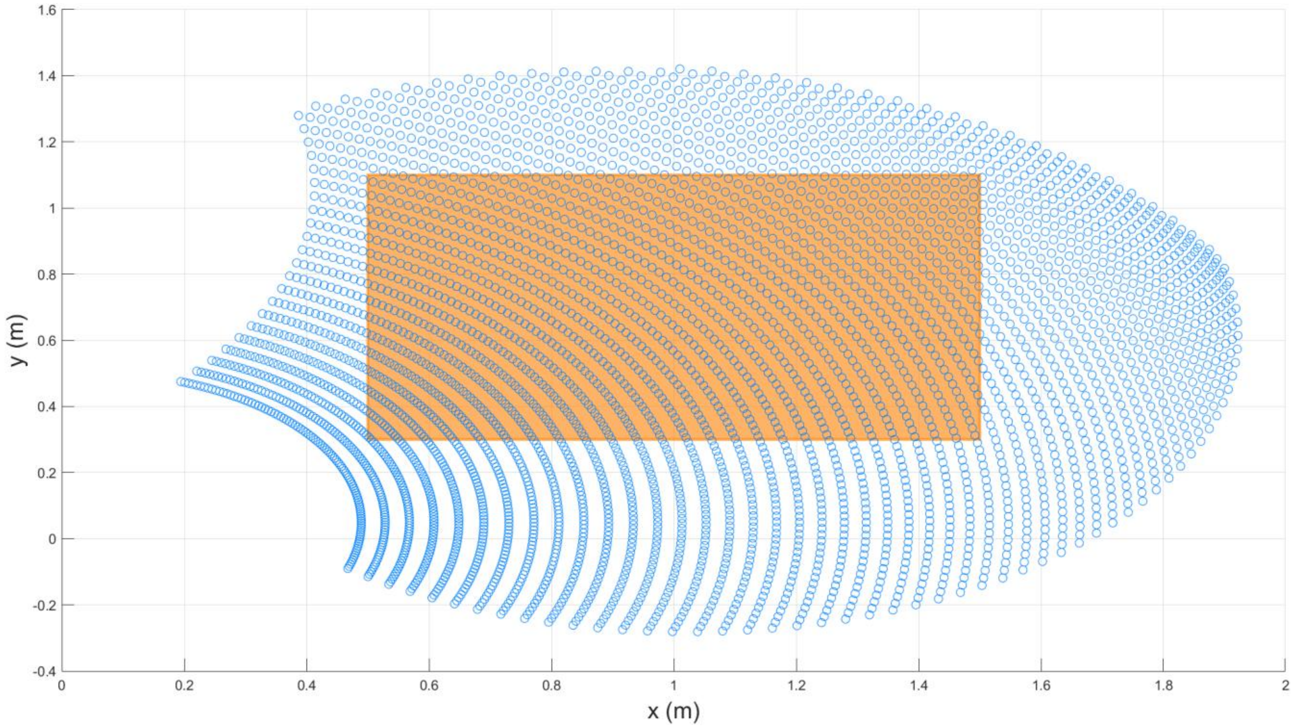


Figure 68 Workspace and operational space of the designed manipulator, including mechanical constraints.

Chapter 5

6 Control and Simulation

The project also involves the simulation of the designed robotic arm to test the motion of the actuated joints. In Simulink, three subsystems have been created: the plant, which represents the mechanical model of the manipulator; the controller which contains the control logic; and a sensor simulator, which emulates all the signals required by the controller logic, such as a detector for the coordinates of the taps. The motors that will be employed allow for multiple control modes, such as current, voltage, position and velocity. Two types of controllers are designed to test two simulation scenarios, one in the joint space and one in the operational space.

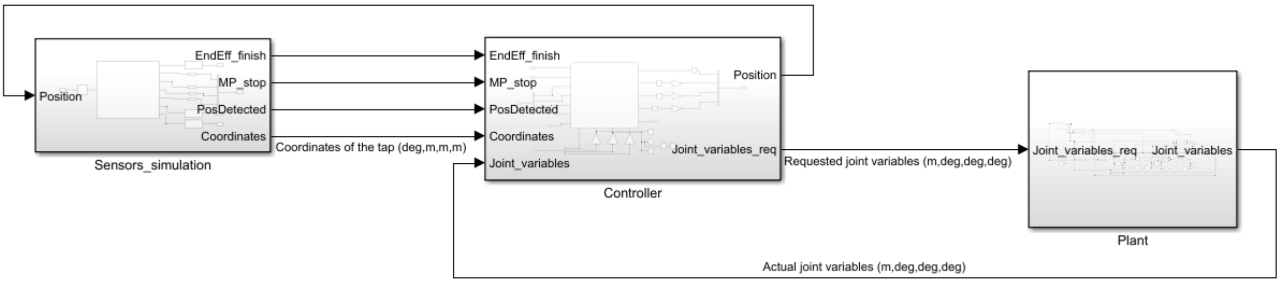


Figure 69 Control subsystems, Sensors, Controller and plant.

Table 16 List of signals

Signal	Type	Unit	Description
EndEff_finish	Boolean	/	Simulates a sensor that returns a signal equal to true when the end effector finishes the sampling tasks, false otherwise.
Mp_stop	Boolean	/	Simulates a sensor that returns a signal equal to true when the mobile platform reaches the position in front of the tap and remains in a stationary position, false otherwise.

PosDetected	Boolean	/	Simulates a sensor that returns a signal equal to true if the coordinates of the tap are detected correctly, false otherwise.
Coordinates	Double	[m,deg]	Simulates a sensor that provides the coordinates of the tap with respect to the base of the manipulator. The coordinates are stored in a vector $[r_z, x, y, z]$.
Joint_variables_req	Double	[m,deg]	The controller computes the required joints variable through the inverse kinematics problem and store it in a vector $[d_1, q_2, q_3, q_4]$.
Joint_variables	Double	[m,deg]	The sensors provide the actual position of the motors. The values are stored in a vector $[d_1, q_2, q_3, q_4]$.
Position	Boolean	/	The variable is true when the position is reached, false otherwise.

Two control models are designed and tested: a control in the joints space and a control in the operational space.

6.1 Plant

To simulate the mechanical model of the manipulator, we export the SolidWorks model in Simulink using Simscape Multibody link. The design presented in Chapter 4 has been simplified by removing components that are not relevant for the simulation, such as pins, screws, covers, gears, and motors. This allows for a simpler Simulink model, focusing only on joints and links. In addition, the prismatic joint J1 is simulated by designing a rectangle that stands for the mobile platform and by allowing the translation of the base of the robotic arm on the rectangle.

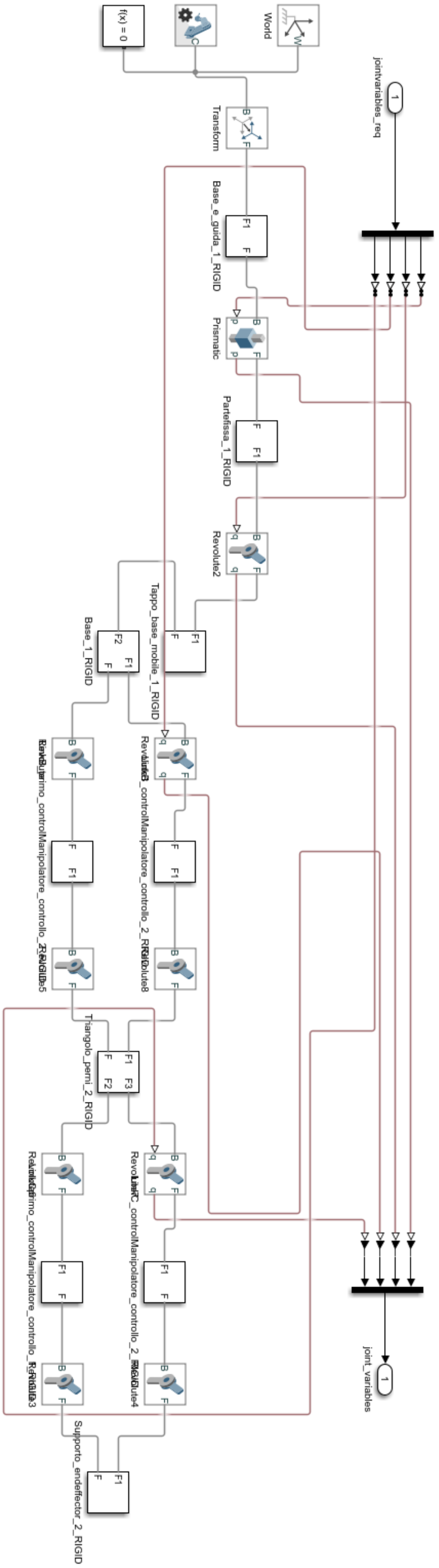


Figure 70 Simulink model of the plant, Simscape multibody link.

In Figure 70, the Simulink model of the plant is represented. The Simscape model is automatically generated by SolidWorks by exporting the designed links and joints. The joints are generated based on the mates exported from the CAD model and are represented in Simulink through the blocks "Revolute Joint" or "Prismatic Joint". The blocks allow to set both the actuation through a position input and the sensing on position. Thus, the active joints receive the control signal from "*Joint_variables_req*" and return the sensing positions as an output. According to the actuators datasheets, actuators may be controlled through a constant goal position enabling the automatic computation of the trajectory by selecting a velocity profile.

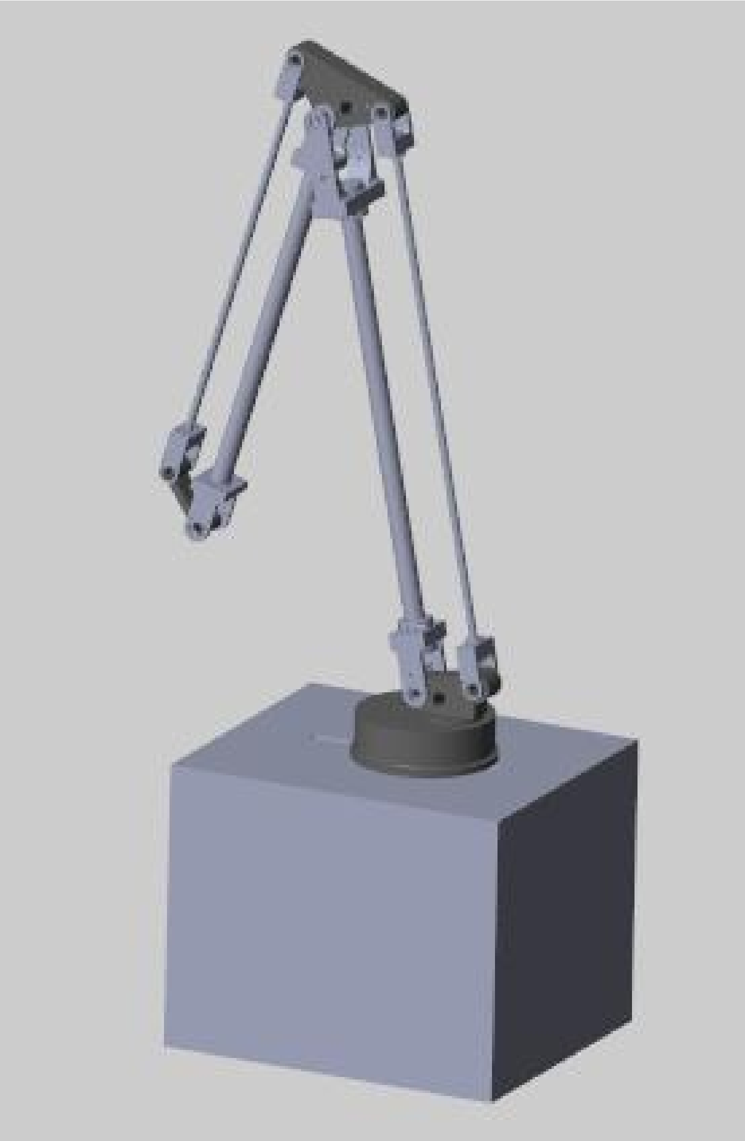


Figure 71 3D Model in Simscape Multibody link.

6.2 Controller

Both the controllers are designed only for simulation and testing purposes. The control of the manipulator applied in the real scenario shall be designed more in detail, ensuring low velocities near the taps. The models presented in this chapter do not path a trapezoidal velocity profile, while the trajectory on both the controllers is not suitable for a rigorous control mode.

6.2.1 Control in the joints space: controller

The controller solves the inverse kinematics problem and returns as an output a constant target position of each joint. Each joint variable is controlled independently from the others, and no trajectory of the end-effector is pathed.

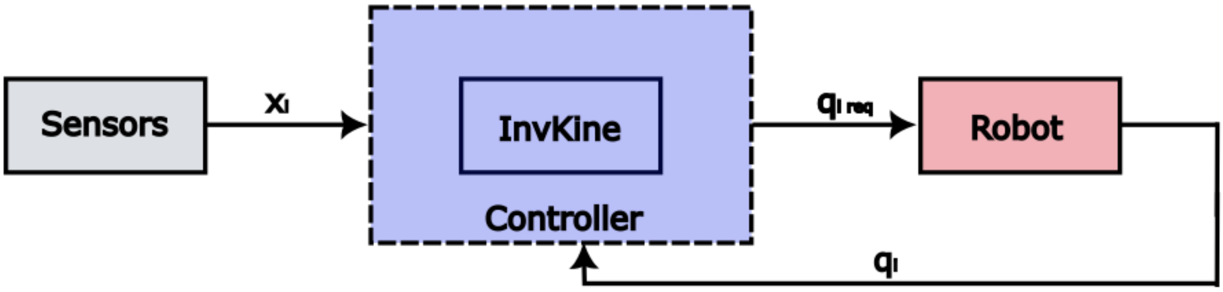


Figure 72 Simplified scheme of the control.

The control logic is designed using *Stateflow*, through a chart that allows to follow the task of sampling. *Stateflow* is a graphical tool in Matlab and Simulink used for designing and simulating state machines and control logic.

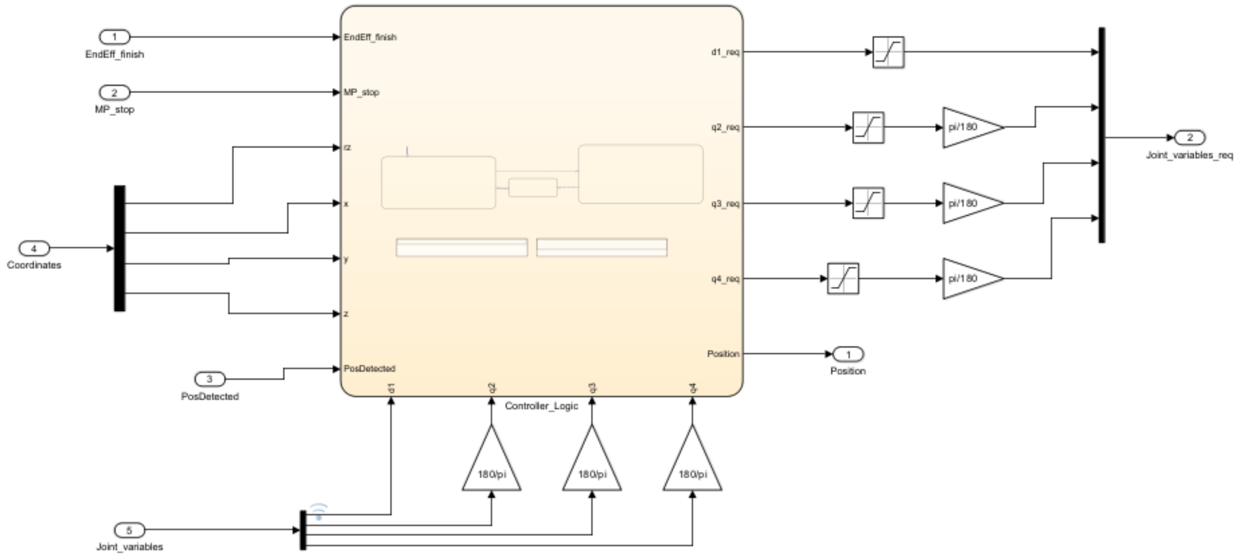
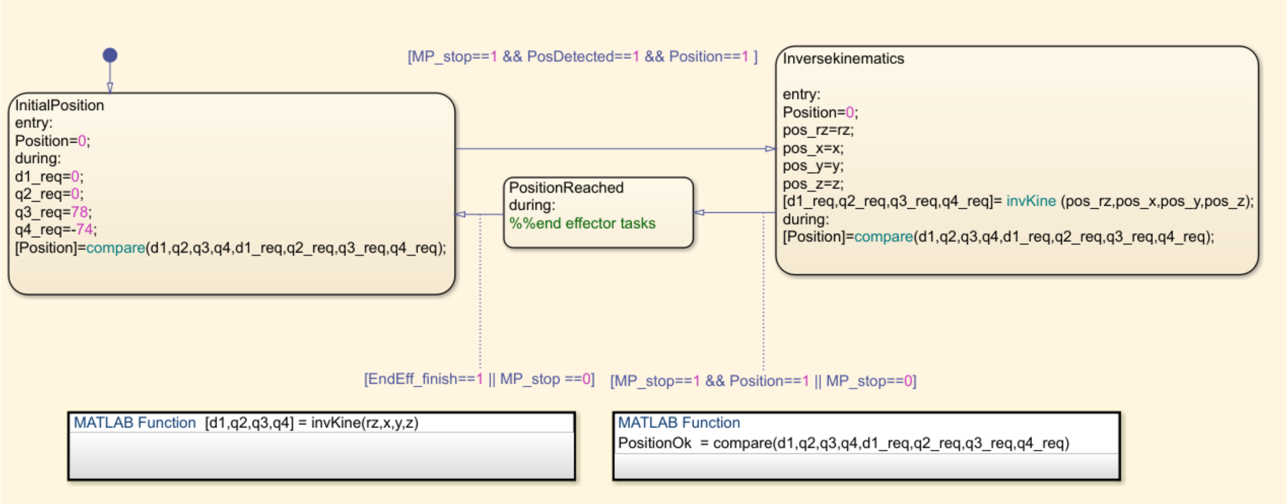


Figure 73 Controller subsystem in Simulink.

The inputs of the controller are all the signals provided from the other components of the mobile robot, such as the mobile platform, the end effector and the camera for distances detection. These signals are required to perform realistic logic for the controller of the robotic arm. In particular, we expected a signal that contains the information about the correct sampling task performed by the end effector and a signal containing data about whether the mobile platform is moving or has reached its position. The camera shall detect the coordinates of the tap with respect to the base and input them to the controller.

The controller solves the inverse kinematics problem obtaining two outputs: the solution of the inverse kinematics and a Boolean signal notifying whether the end effector has reached its goal position. The saturation blocks include the mechanical constraints of the manipulator.

The logic of the controller is shown in Figure 74.



```
InitialPosition
entry:
Position=0;
during:
d1_req=0;
q2_req=0;
q3_req=78;
q4_req=-74;
[Position]=compare(d1,q2,q3,q4,d1_req,q2_req,q3_req,q4_req);
```

```
Inversekinematics

entry:
Position=0;
pos_rz=rz;
pos_x=x;
pos_y=y;
pos_z=z;
[d1_req,q2_req,q3_req,q4_req]= invKine (pos_rz,pos_x,pos_y,pos_z);
during:
[Position]=compare(d1,q2,q3,q4,d1_req,q2_req,q3_req,q4_req);
```

```
PositionReached
during:
%%end effector tasks
```

```
[MP_stop==1 && PosDetected==1 && Position==1 ]
[EndEff_finish==1 || MP_stop ==0]
[MP_stop==1 && Position==1 || MP_stop==0]
```

```
MATLAB Function  [d1,q2,q3,q4] = invKine(rz,x,y,z)
```

```
MATLAB Function
PositionOk  = compare(d1,q2,q3,q4,d1_req,q2_req,q3_req,q4_req)
```

Figure 74 Stateflow, control logic.

We define 3 states where the target joints variables are computed:

- *InitialPosition*: the state containing the variables to reach a rest position;
- *Inverse kinematics*: the state solving the inverse kinematics solution to reach the tap;
- *Position reached*: the state waiting for the end effector to accomplish the extraction task. .

The rest position is chosen to reduce the footprint. In particular, the joints variables extend the end-effector in the nearest position of the operational space of $(x_e, y_e) = (0.5m, 0.3m)$. Since the orientation of the 4th motor does not depend on the variable of q_3, the variable defined in the chart as q_4 represents the angle $q_3 + q_4$ from the kinematics equations 30-31.

The control algorithm follows the logic explained below:

1) From the rest position the controller waits that the mobile platform is stopped, and the coordinates of the tap are correctly detected. If the conditions are satisfied, we proceed with step 2.
2) The controller changes the state and starts to compute the inverse kinematics problem for the coordinates received as an input from the camera. The solutions are computed through the inverse kinematics equations in Chapter 4. Then, the requested joints variables are sent to the motors. A function defined as "*compare*" compares the requested variables to the instant joints variables and returns a Boolean variable "*Position*" = true if all the joint variables differs only within a certain threshold, and hence the goal position is reached with a good approximation. The controller then switches to the state "*PositionReached*", in which the end-effector can start the extraction from the barrels.

3) Once the end effector finishes the extraction task and the variable "EnfEff_finish" becomes true, the state comes back to the "InitialPosition". When the rest position is reached and the new coordinates are detected, the controller continues in loop with the aforementioned computations.

6.2.2 Control in the operational space: Controller

For the control in the operational space, we tried to map a trajectory by controlling the final position of the end effector instead of the joints position. The output of the controller is a variable joint position value. The control scheme is shown in Figure 75.

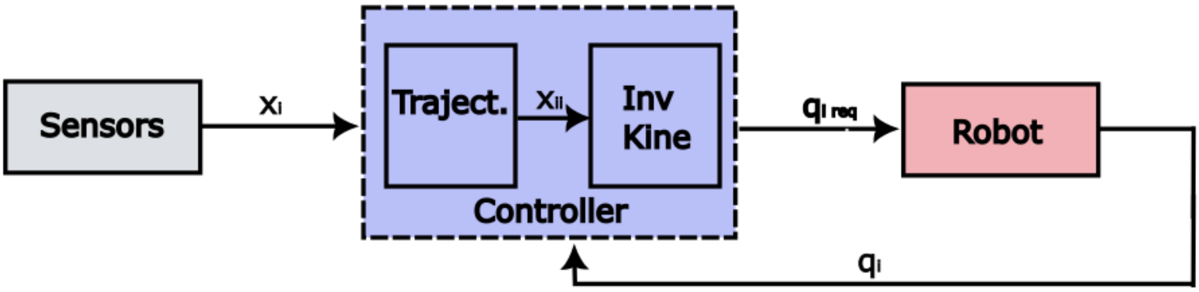


Figure 75 Simplified model of the control.

The controller is designed using Stateflow and has a more complex structure with respect to the controller in the joint space. The simplified scheme is depicted in Figure 76.

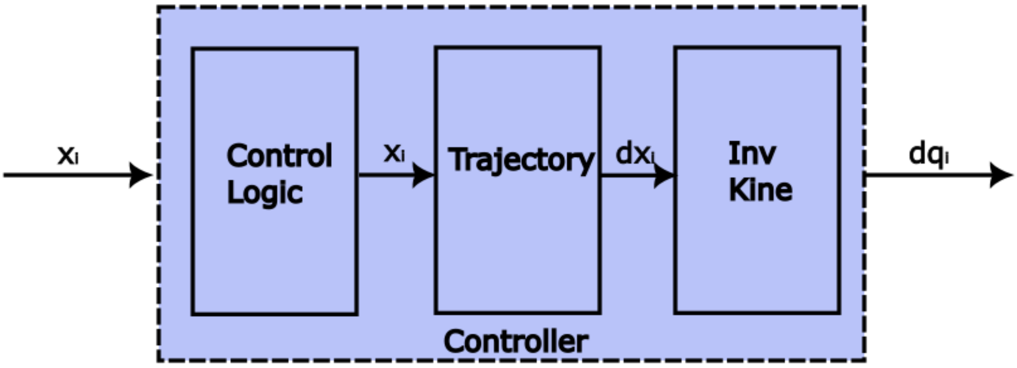


Figure 76 Simplified scheme of the controller.

The controller receives as an input information about the position of the tap. The control logic defines when and how the end effector shall reach a position, based on a logic that will be presented later into this Section. The trajectory block receives the constant target position and sends an instant position of the end effector. Then, the inverse kinematics is performed to give an instant solution of the positions.

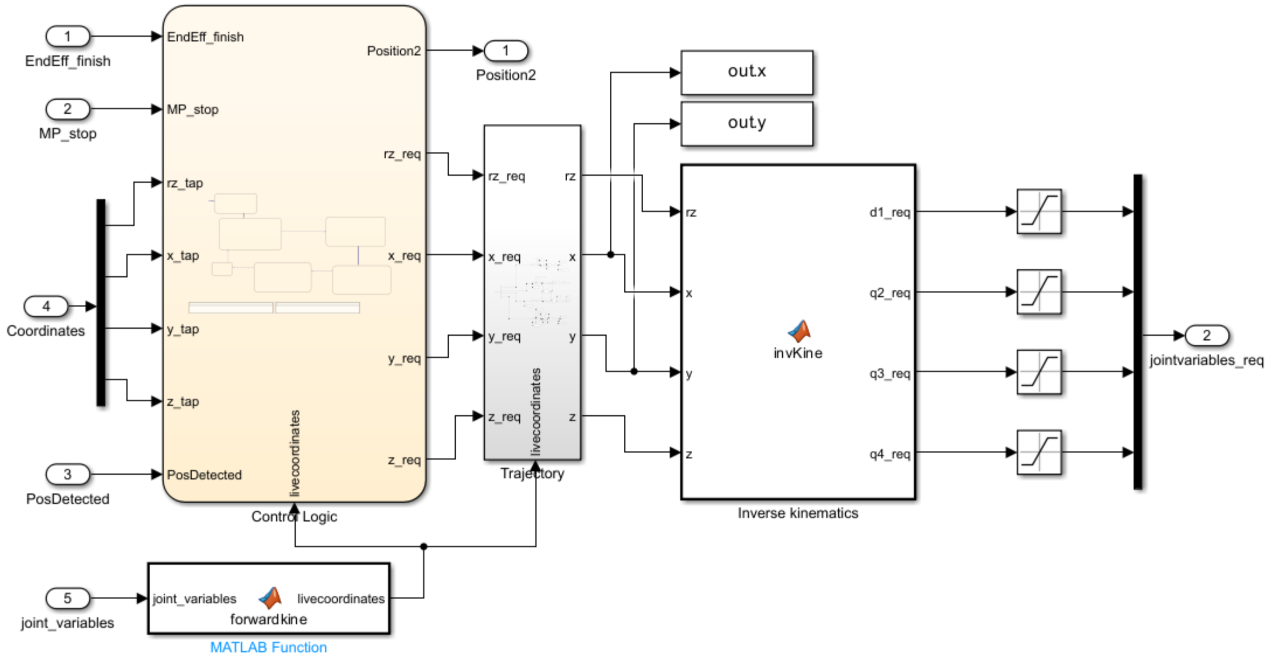
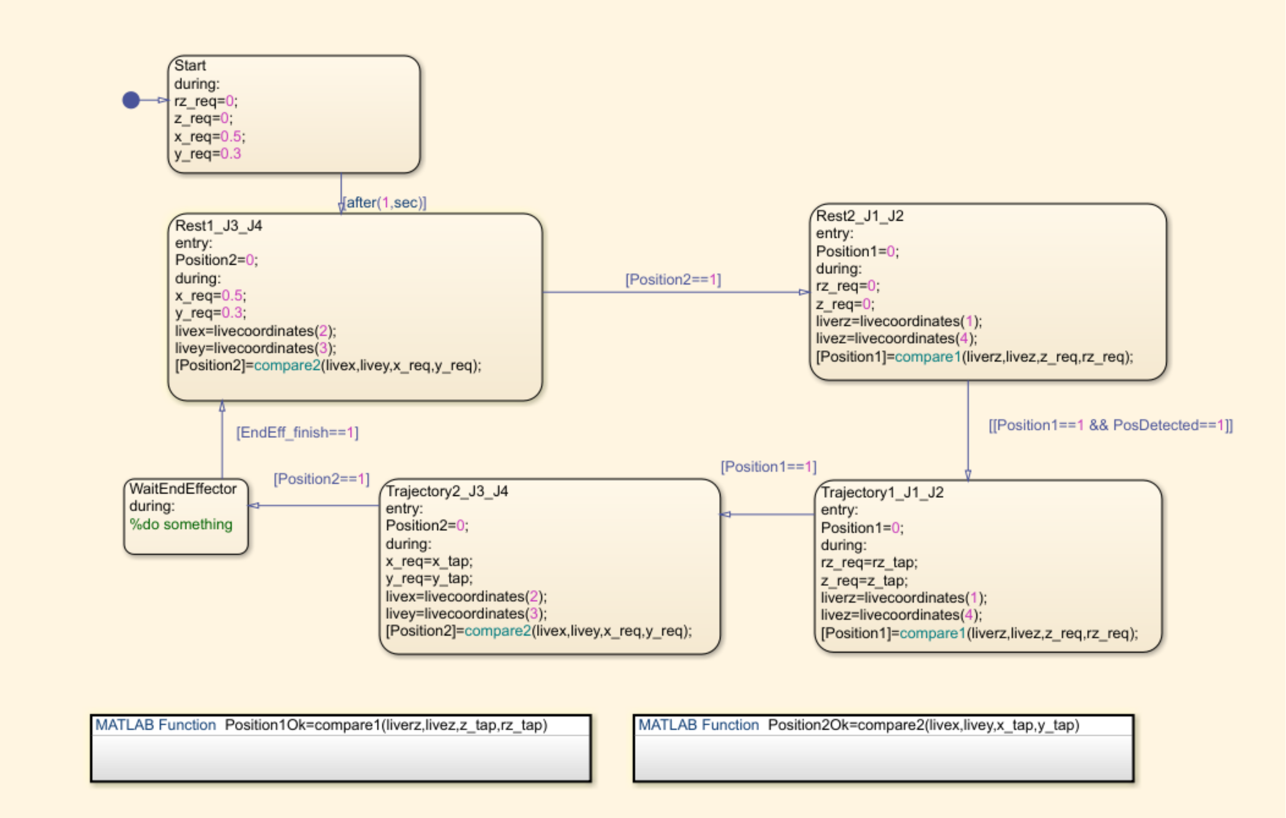


Figure 77 Subsystem of the controller.

In this control mode, the positions of the joints change with an order expressed by the states in Stateflow. There are four states:

- **Rest1_J3_J4:** as a first operation, the manipulator shall reach the rest position considering joint J3 and J4. The rest position of joints 3 and 4 guarantees the less bulky extension.
- **Rest2_J1_J2:** Once the target extension is reached, the end effector reaches the entire rest position with the control of joints J1 and J2.
- **Trajectory1_J1_J2:** If the position of the tap is detected, the states give the positions considering first joints J1 and J2.
- **Trajectory2_J3_J4:** The extension is controlled by joints J3 and J4 and the end effector is ready to perform the extraction task.

The logic ensures that when the joints J1 and J2 change in position, the extension of the robotic arm is minimized with respect to the bulk.



```
Start
during:
rz_req=0;
z_req=0;
x_req=0.5;
y_req=0.3
```

```
Rest1_J3_J4
entry:
Position2=0;
during:
x_req=0.5;
y_req=0.3;
livex=livecoordinates(2);
livey=livecoordinates(3);
[Position2]=compare2(livex,livey,x_req,y_req);
```

```
Rest2_J1_J2
entry:
Position1=0;
during:
rz_req=0;
z_req=0;
liverz=livecoordinates(1);
livez=livecoordinates(4);
[Position1]=compare1(liverz,livez,z_req,rz_req);
```

```
WaitEndEffector
during:
%do something
```

```
Trajectory2_J3_J4
entry:
Position2=0;
during:
x_req=x_tap;
y_req=y_tap;
livex=livecoordinates(2);
livey=livecoordinates(3);
[Position2]=compare2(livex,livey,x_req,y_req);
```

```
Trajectory1_J1_J2
entry:
Position1=0;
during:
rz_req=rz_tap;
z_req=z_tap;
liverz=livecoordinates(1);
livez=livecoordinates(4);
[Position1]=compare1(liverz,livez,z_req,rz_req);
```

```
MATLAB Function  Position1Ok=compare1(liverz,livez,z_tap,rz_tap)
```

```
MATLAB Function  Position2Ok=compare2(livex,livey,x_tap,y_tap)
```

Figure 78 Stateflow, control logic.

The trajectory block transforms the constant target position in a time dependent variable position. The position of the end-effector changes linearly and the velocity is represented by a square wave.

- Joints J1 and J2: the position of the end effector changes linearly in J1 and J2 directions and with no dependance from each other's. The constant velocities are settled through Simulink.
- Joints J3 and J4: **these joints** are responsible for the manipulator extension. We define x_i, y_i as the actual position x and y of the end effector and x_{req}, y_{req} as the coordinates x and y of the tap. The velocity of the end effector along the x and y directions: $v_x = \frac{dx}{L}$ and $v_y = \frac{dy}{L}$ where $dx = x_{req} - x_i$, $dy = y_{req} - y_i$, and $L = \sqrt{dx^2 + dy^2}$. The velocities of the end effector guarantee that the trajectory passes through line L between points (x_i, y_i) and (x_{tap}, y_{tap}). In this manner the trajectory is controlled through the end effector position.

Then, the variable positions input the inverse kinematics block, which computes the respective joints positions.

6.3 Simulation

To simulate the controller and the plant in the Simulink environment, a sensor simulator system is designed. The logic is chosen to simulate a real behavior of the sampling task. The model illustrated in Figure 80 is designed through Stateflow and operates with the following behavior:

- The mobile platform stops in front of the barrels;
- After a time *t*, the camera detects the position of one tap;
- Once the controller gives the information about the correct achievement of the position, the end effector performs the extraction task;
- Once the end effector finishes the task, and the manipulator reaches the rest position, the operations are performed in loop.

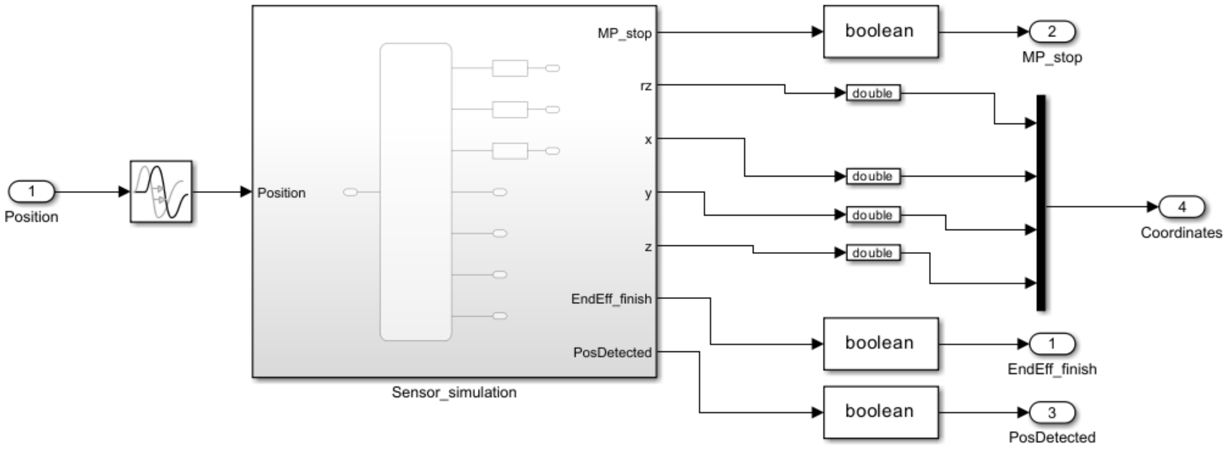
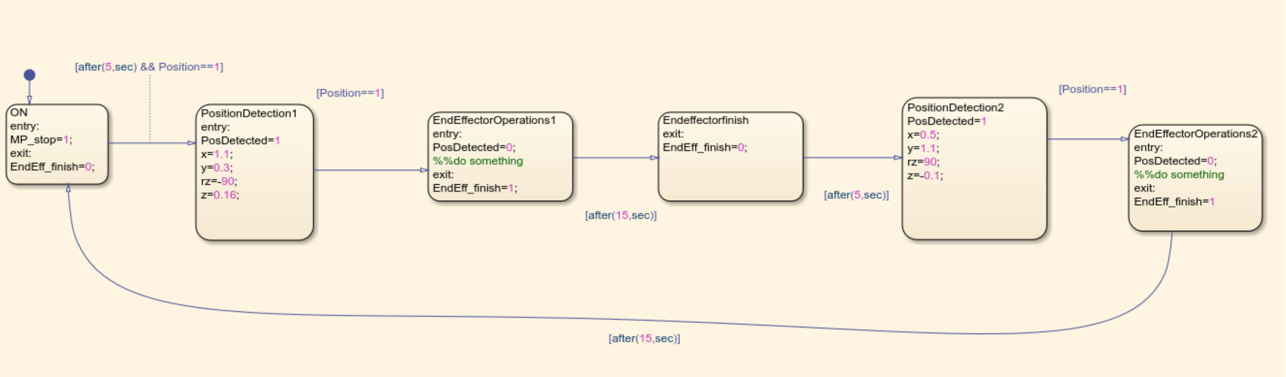


Figure 79 Subsystem of the Sensors simulation.



```
ON
entry:
MP_stop=1;
exit:
EndEff_finish=0;
```

```
PositionDetection1
entry:
PosDetected=1
x=1.1;
y=0.3;
rz=-90;
z=0.16;
```

```
EndEffectorOperations1
entry:
PosDetected=0;
%%do something
exit:
EndEff_finish=1;
```

```
Endeffectorfinish
exit:
EndEff_finish=0;
```

```
PositionDetection2
PosDetected=1
x=0.5;
y=1.1;
rz=90;
z=-0.1;
```

```
EndEffectorOperations2
entry:
PosDetected=0;
%%do something
exit:
EndEff_finish=1
```

Figure 80 Stateflow, logic of the signals generator.

In the simulation are tested the cases of $[x, y, z, ez] = [1.1,0.3,0.16, -90°]$ and $[x, y, z, ez] = [0.5,1.1,0.1,90°]$, for both the control in the joint space and the control in the operational space.

6.3.1 Control in joint space

To simulate the rectangle trajectory profile, the block in Figure 81 is inserted between a desired position and the rotative joint in the plant the subsystem, in order to choose the velocity of the trajectory only for a simulation purpose. For the test is chosen a velocity equal to $\left|\frac{pi}{6}\right| \frac{rad}{s}$ for each rotative joint and equal to $|0.2| \frac{m}{s}$ for the prismatic joint J1.

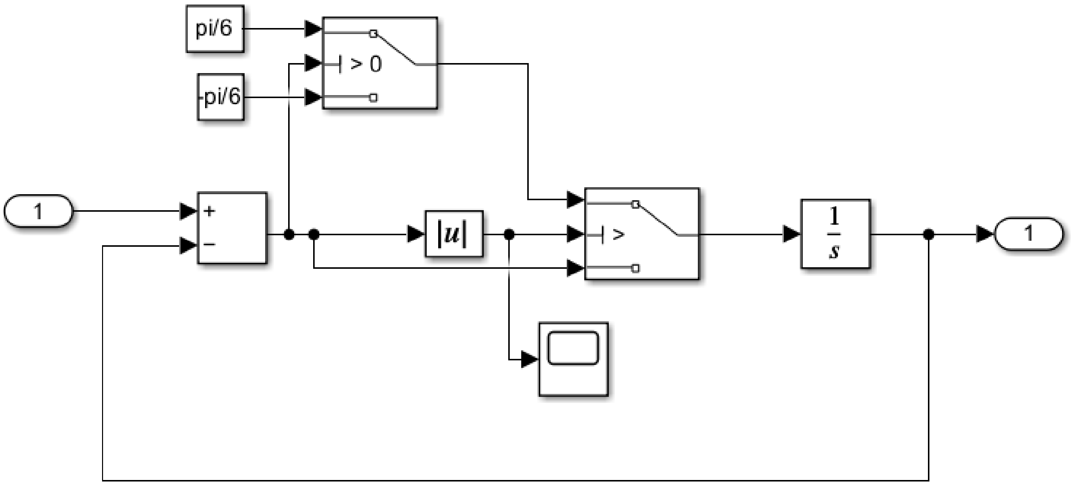


Figure 81 Rectangle Profile Trajectory simulation.

The taps are simulated through two spherical solids, each located in the simulated position. The frames of the simulation are illustrated in Figures 82-83-84.

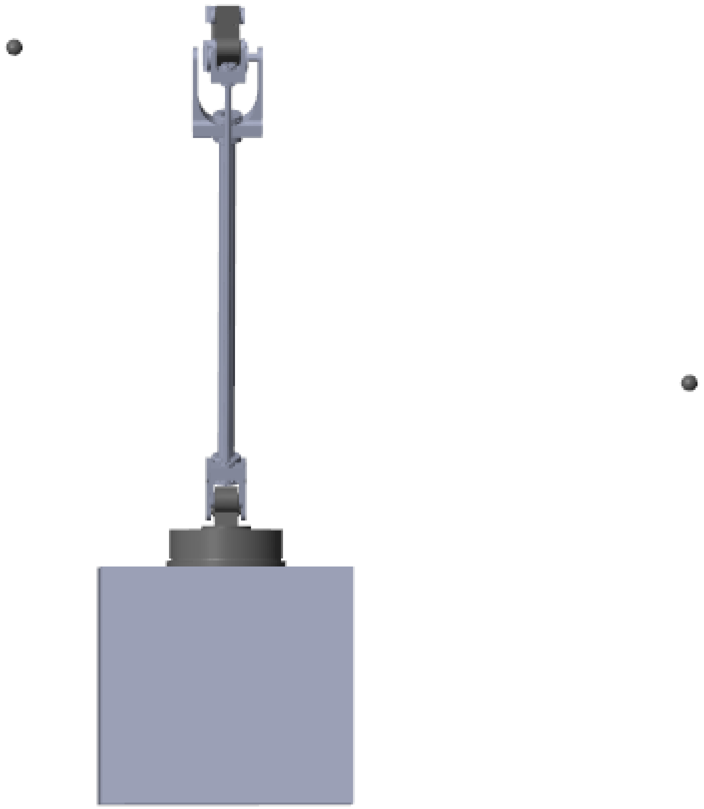


Figure 82 Rest position.

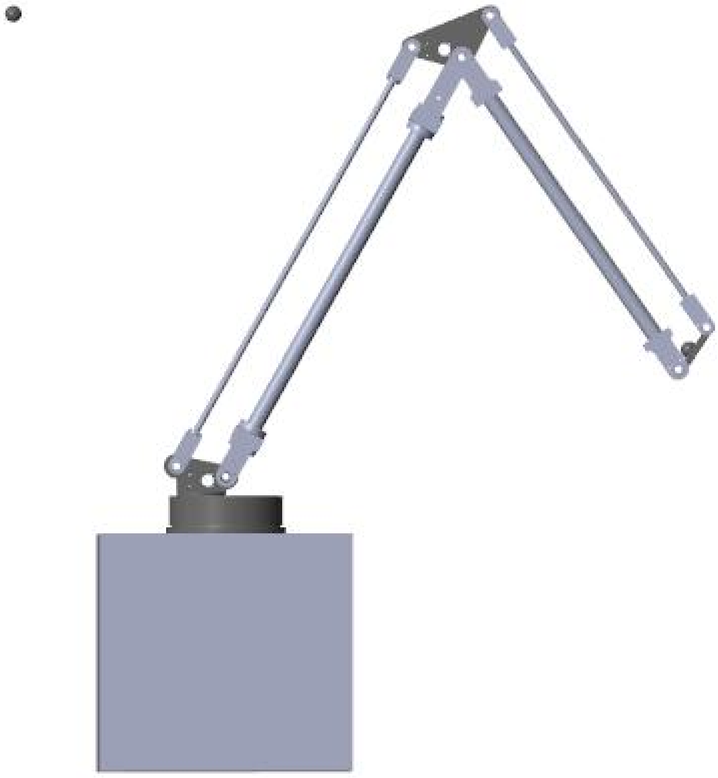


Figure 83 Right tap reached [1.1,0.3,0.16, −90°]]

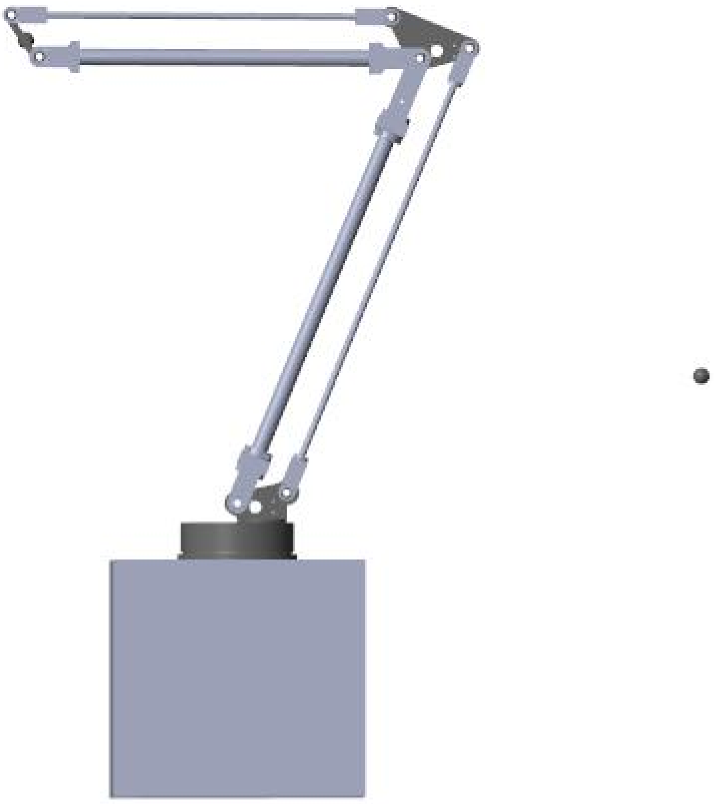
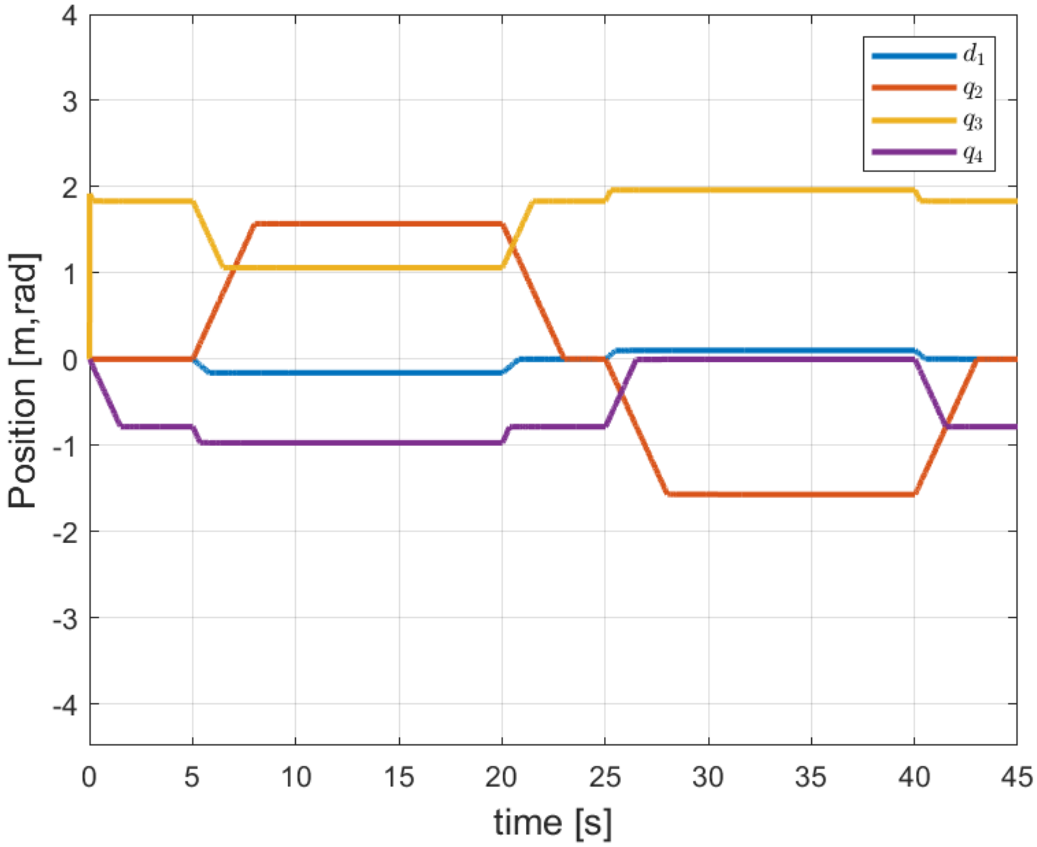


Figure 84 Left tap reached [0.5,1.1,0.1,90°].

The manipulator reaches both the two simulated positions correctly. In Figure 85 are shown the positions and the velocities of each joint during the simulation time with measure units [m,rad,rad,rad].


Position [m,rad]
time [s]
4
3
2
1
0
-1
-2
-3
-4
0
5
10
15
20
25
30
35
40
45
d_1
q_2
q_3
q_4

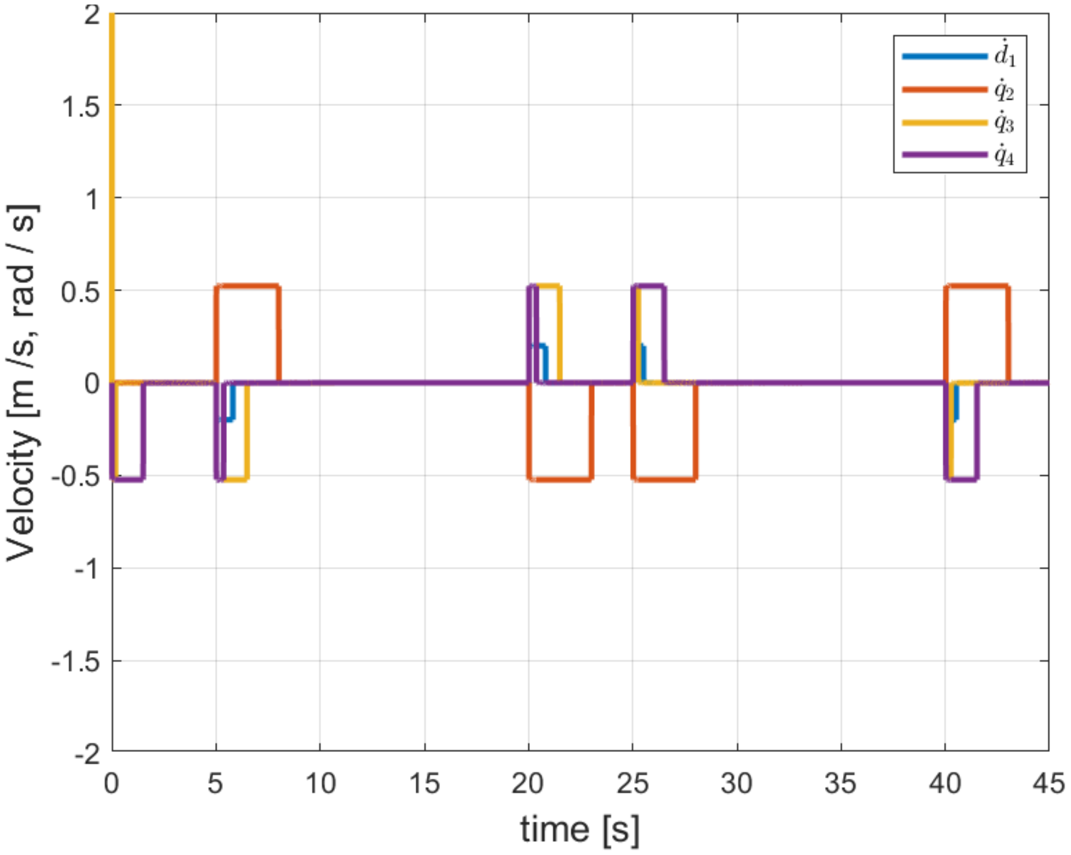


Figure 85 Position-velocity.

6.3.2 Control in the operational space.

The controller sets a trajectory for the end effector regarding the extension of the arm. In particular, we expect the trajectory in the x-y plot to change linearly on the line from the actual coordinates (x, y) to the target position (x_f, y_f) . We plot through Simulink the curve (x(t),y(t)), that illustrates the trajectory of the end effector from the point (0.5m,0.3m) to the point (1.5m,1.1m).

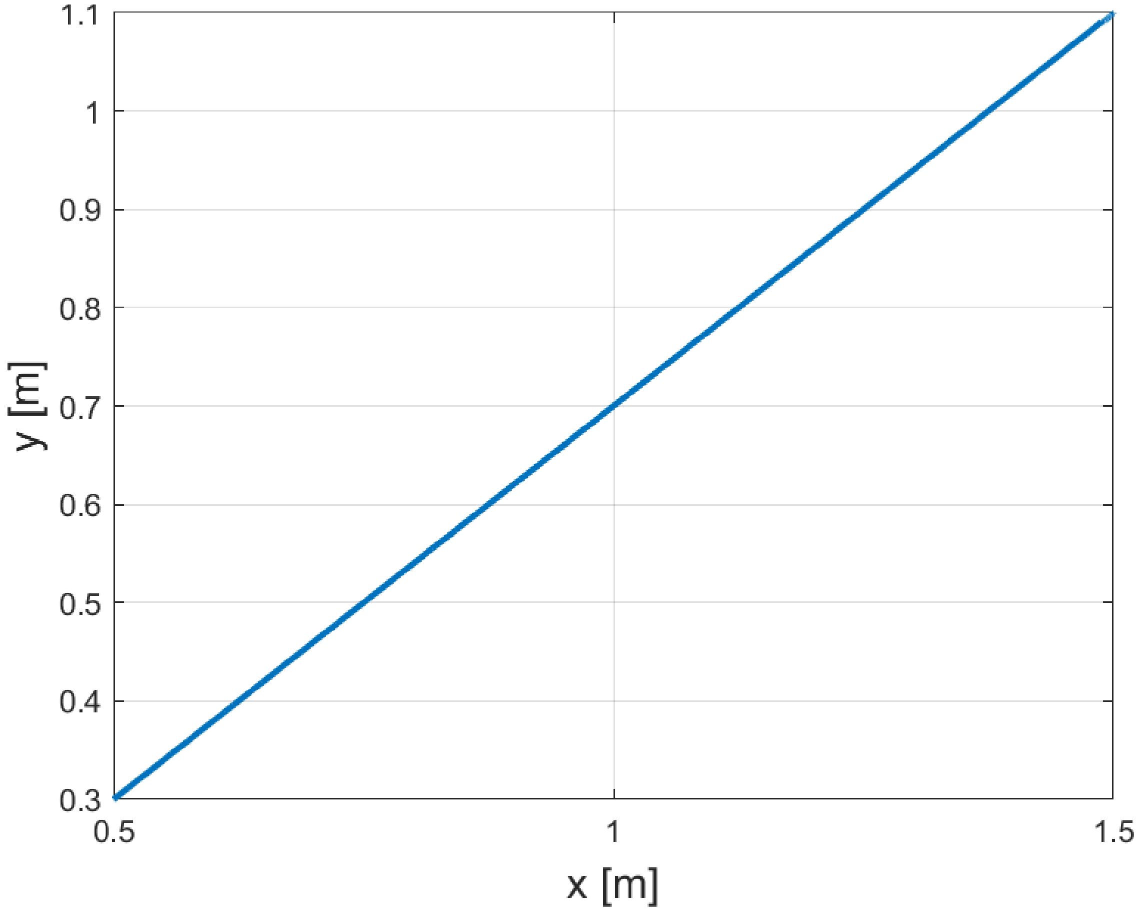


Figure 86 Simulated trajectory, from (0.5m,0.3m) to (1.5m,1.1m)

As expected, the simulation shows a trajectory that coincides with the diagonal $L = \sqrt{dx^2 + dy^2}$.

The controller controls the trajectory, thus we simulate the model with the same signal as in Figure 80. The joint positions are plotted in Figure 87.

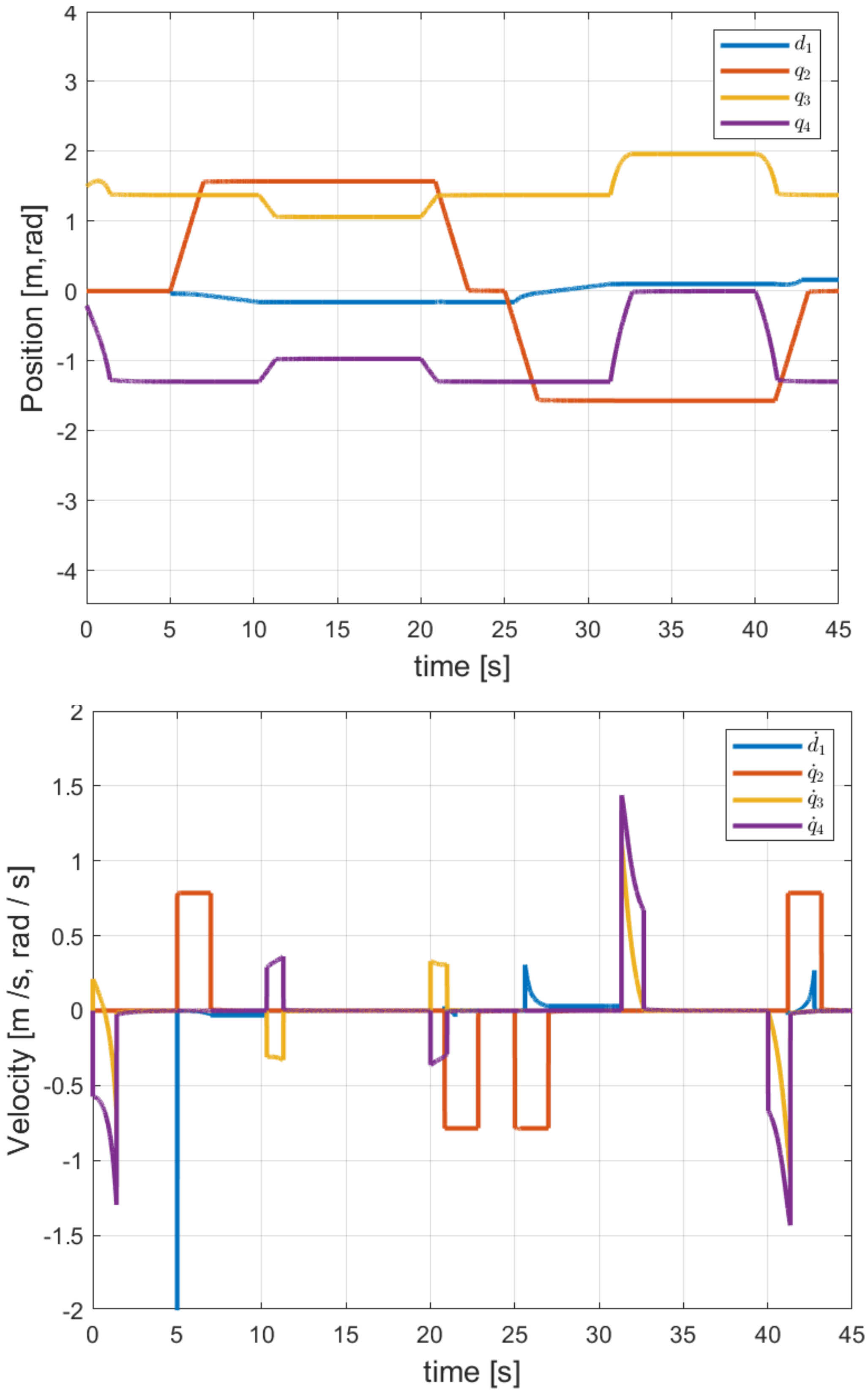


Figure 87 Position-Velocity

6.4 Comparison

Both the control in the joint space and the control in the operational space follow the same logic designed on Stateflow. Nonetheless, while the first is controlled through the joint position, the second controls the coordinates of the end effector, defining a trajectory for the extension of the manipulator.

The **control in the joints space** has a simpler structure but does not consider the end effector trajectory. All the joints are actuated independently from the others. The controller returns as an output a constant value that is the input of the actuators.

The **control in the operational space** has a more complex structure. The trajectories for the joints J1 and J2 are expressed as a linear change of the coordinates of the end effector, and during their motion the manipulator stays in the less extended position of J3 and J4 in order to reduce the bulk. The extension of the arm is driven by the actuation of joints J3 and J4, while the coordinates change by following a liner path from the actual position to the final one. The output of the controller is a signal with the joints position that changes depending on the variable end effector position.

The control in the joints space should be preferred if there are no obstacles in the corridor, while the control in the operational space ensures the trajectory while preventing excessive arm extensions. Since it is specified that the control was done to simulate and test the model, both models must be refined by designing a well-specific trajectory allowing to control the speeds along the trajectory. The approach path of the end effector to the tap has been neglected, but it is expected that the control will provide a slow approach avoiding any obstacles near the tap.

Chapter 6

7 Balancing System

7.1 Gravity compensation

After providing an overview in Chapter 1 of gravity compensation methods in robotics, we aim to delve deeper into this topic in order to develop a balancing system for the designed manipulator. The focus will be on analyzing balancing mechanisms that use spring-and-pulley systems, spring-and-cam systems, and, if necessary, counterweights. The analysis of counterweights will be reserved for a later stage, as the primary objective is to minimize additional weight on the robot's structure.

We will now outline the general procedures for balancing a generic link with mass m relative to a rotative joint, using both spring-and-pulley and cam-and-pulley systems.

7.1.1 Balancing with a linear spring and a circular pulley

The design of the balancing system using a linear spring and a circular pulley consists in setting the parameters h and p, with reference to Figure 88, in order to compute the spring parameters that allows a point-to-point gravity compensation.

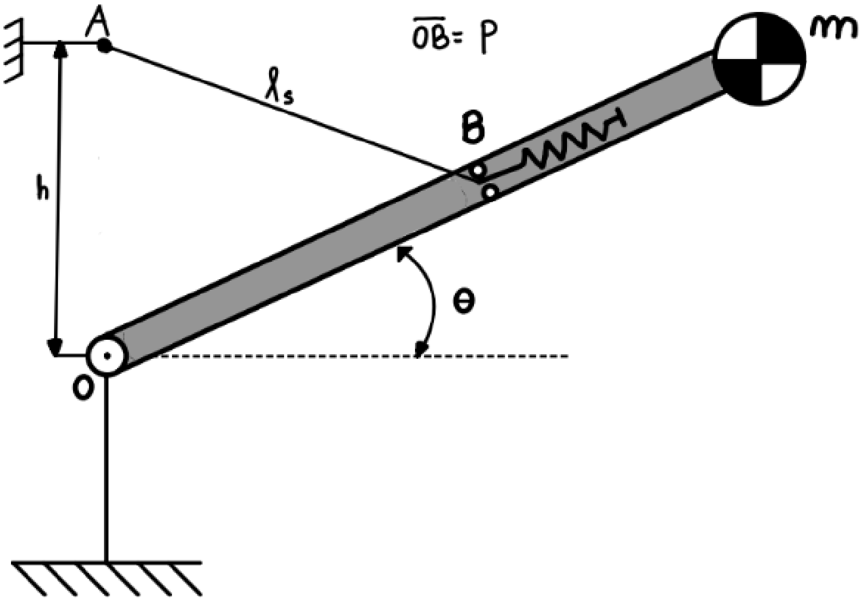


Figure 88 Gravity compensation with a linear spring and a circular pulley model.

We define:

$p\ = \overline{OB},$

$F_e =$ elastic force,

T_e = elastic torque,

k = spring constant,

ϕ = angle between l_s and p.

The torque generated by the gravitational force of mass m is expressed by:

$$T = -m\ g\ l_m \cos(\theta), \tag{41}$$

with $l_m =\ \overline{Om}.$

The general relations for the system in Figure 68:

$$F_e = k\ (l_s - l_0), \tag{42}$$

$$l_s(\theta) = \sqrt{(p\cos(\theta))^2 + (h - p\sin(\theta))^2}, \tag{43}$$

$$l_0 = h - p. \tag{44}$$

The point-to-point gravity compensation is achieved when $l_0 = 0$ [10]. Thus:

$$h = p. \tag{45}$$

The elastic torque:

$$T_e = \ p \ F_e \sin(\phi), \tag{46}$$

$$h \ cos(\theta) = l_s \sin(\phi). \tag{47}$$

Substituting the (47) in the equation (46), the elastic torque:

$$T_e(\theta) = \frac{p \ F_e \ h \cos(\theta)}{l_s} = \ p \ k \ h \cos(\theta) \tag{48}$$

The balancing condition:

$$T_e = -T,$$

$$p \ k \ h \cos(\theta) = \ m \ g \ l \cos(\theta). \tag{49}$$

The spring constant k:

$$k = \frac{m \ g \ l}{p \ h}. \tag{50}$$

7.1.2 Balancing with a linear spring and a cam

The project involves developing a balancing system using a linear spring connected via a belt to a cam attached to the joint, depicted in Figure 89. The design of the balancing system, therefore, focuses on determining the cam's configuration and selecting a spring that ensures consistent balancing throughout the entire workspace. The cam design is approached through an iterative graphical method, which can be practically implemented using software tools. The algorithm is outlined in the following lines and refers to the parameters illustrated in Figure 90.

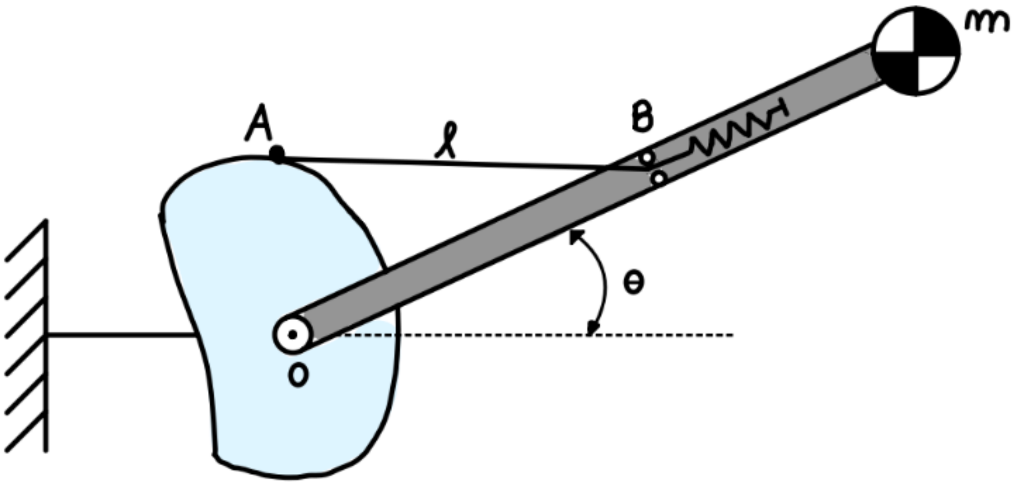


Figure 89 Gravity compensation with a linear spring and a cam model.

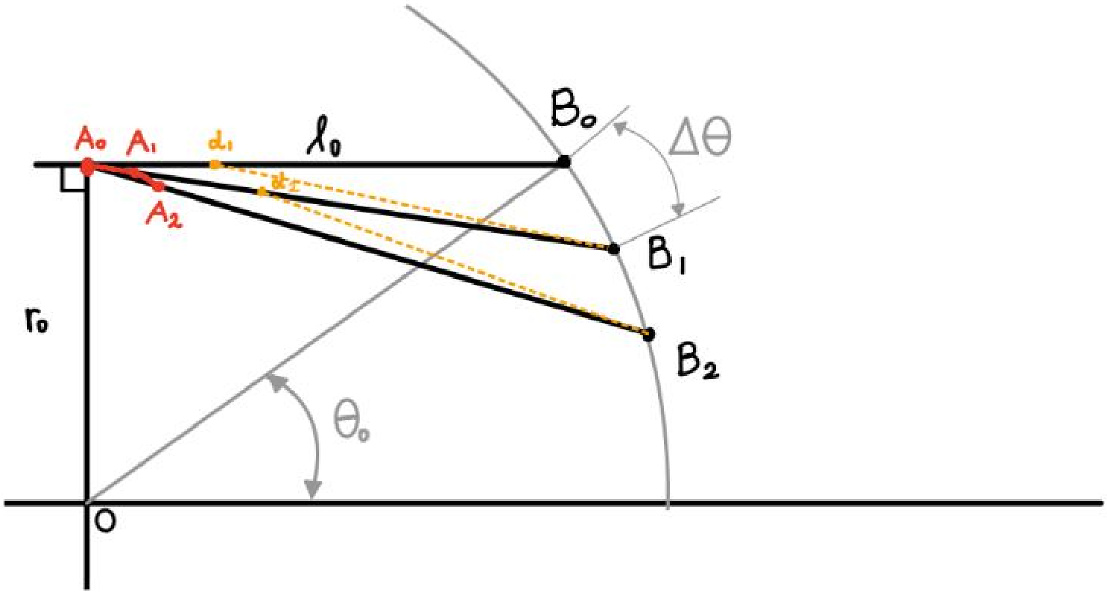


Figure 90 Graphical algorithm for cam design.

-Choose the length $l_b = \overline{OB}$.

- Set $\theta_0 = \theta_{max}$ and choose r_0. Find A_0 such that $\overline{A_0B_0} \perp \overline{OA_0}$.

- Evaluate $l_0 = \|p_{A_0} - P_{B\,0}\|$ and $F_0 = \frac{\tau(\theta_0)}{r_0}$.

- Choose the spring constant k and the spring natural length l_n such that:

$(l_0 - l_n)k + F_n = \tau(\theta)\backslash r_0$, where F_n is the preload at the natural length.

-Choose a small enough value for $\Delta\theta$.

1) Set $i = 1$,

$\theta_i = \theta_{i-1} - \Delta\theta$, place a point α_i on $\overline{A_{\iota-1}\, B_{\iota-1}}$: $p_{\alpha\, i} = p_{A_{i-1}} + \alpha_i(p_{A\, i-1} - p_{B\, i-1})$.

$l_i = l_{i-1} - (\|p_{B\, i-1} - p_{\alpha\, i}\| - \|p_{B\, i} - p_{\alpha\, i}\|)$.

$F_i = (l_i - l_0)k + F_0$ and $r_i = dist(line(B_i, \alpha_i), origin)$.

2) Find the value of a_i such that $F_i r_i = \tau(\theta_i)$ and set $A_i = \alpha_i$.

3) Set $i = i + 1$ and iterate until $\theta_i = \theta_{min}$.

Practically, the shape of the cam is designed by tracing a curve through the computed points of A_i. The cam design allows the link to be balanced without requiring the condition $h = p$ noticed in the balancing by using only the linear spring and the circular pulley, resulting in a solution that add space consuming component to fix the wire at an height of h. In contrast, the cam allows to design a component with a variable radius where the values of l_b and r_0 may be reasonably chosen at the beginning of the algorithm, offering a more compact balancing solution.

7.2 Torques analysis

Before designing the balancing system, we first aim to analyze the torques generated by the gravitational forces acting on the manipulator's components at joints J3 and J4 as a function of the angles q_3 and q_4. To simplify the analysis, the double parallelogram configuration is represented as an R-R arm.

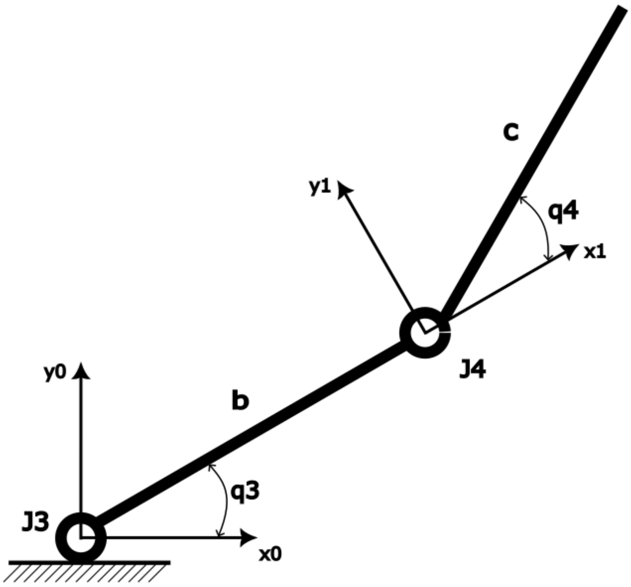


Figure 91 R-R manipulator semplification.

For simplification purposes, we consider links b and b′ as a single unit, as well as links c and c′. In SolidWorks, we updated all components by assigning the appropriate material to each. For the 3D-printed components, PLA is selected as printing material. Using the 'mass properties' function in SolidWorks, we can simulate the mass of the components and the coordinates of the center of mass relative to a reference point. We then calculate the torque around joint J3, $T_3(q_3, q_4)$, and the torque around joint J4, $T_4(q_4)$. To simulate the presence of the end effector, a cubic component with a weight of 2 kg is designed.

We place the manipulator with $q_3 = 90°$ and $q_3 + q_4 = 0°$. Using the mass properties feature, we then select links *c*, *c′*, and the cube simulating the end effector. In Figure 92, the selected elements whose mass we intend to calculate are highlighted in blue.

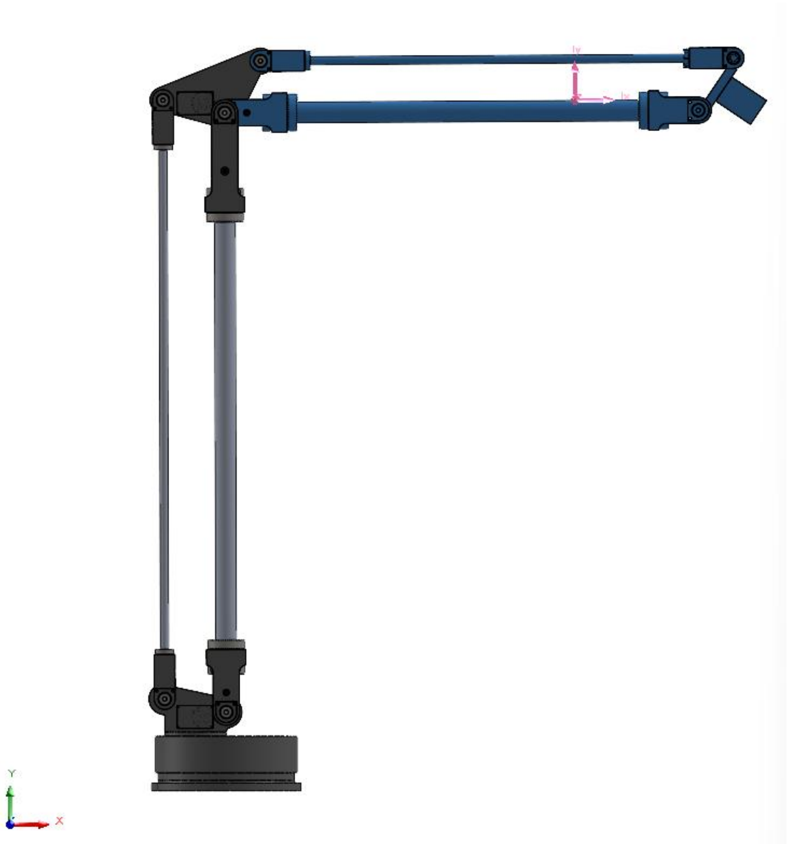


Figure 92 Mass m_c analysis.

The selected components generate a torque around joint J4, then we proceed by reading the mass m_c and the coordinates of the center of mass relative to J4 $G_4(x, y)$.

$$m_c = 6.35\ kg$$

$$G_4(x, y) = (668.3\ mm, 23.5\ mm)$$

Following the same approach, we calculate the mass of the remaining components that contribute to the torque around joint J3.

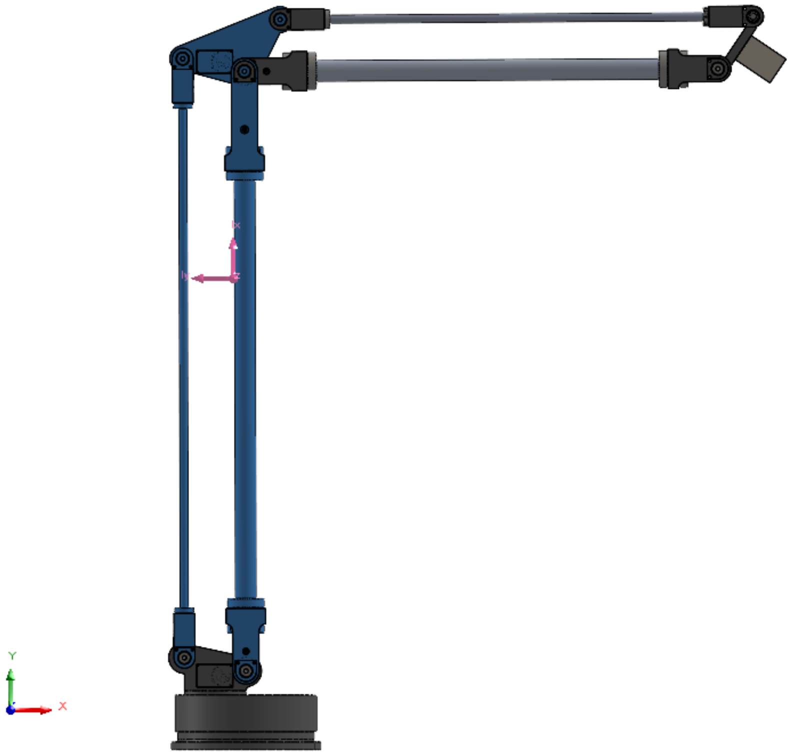


Figure 93 Mass m_b analysis.

The sum of the masses that create rotation around joint 3, denoted as m_b ,and the coordinates of the center of mass with respect to joint 3 G_3:

$$m_b = 6.52\, kg$$

$$G_3(x, y) = (747.2\, mm, -22.4\, mm)$$

7.2.1 Torque around J4

Using the data about the mass and the coordinates of the center of mass, the torque T_4 may be computed.

Since during the measurements in SolidWorks the manipulator is posed with an angle $q_3 + q_4 = 0$, the angle θ_4 expresses the angle of the line with length L_4 (that connects Joint J4 and the point G_4) and the x-axis.

$$\theta_4 = \tan^{-1}\left(\frac{23.5}{666.3}\right) = 2.01°, \tag{51}$$

$$L_4 = \sqrt{23.5^2 + 666.3^2} = 666.71\, mm = 0.667\, m, \tag{52}$$

$$T_4(q_3 + q_4) = m_c\, g\, L_4 \cos(q_3 + q_4 + \theta_4). \tag{53}$$

In Figure 94, the plot of the torque $T_4(q_3 + q_4)$ computed in Matlab is shown. The considered range of values of $q_3 + q_4$ is [-80°,10°], which includes the maximum and minimum values required to reach the positions in the operational space plus a threshold equal to ≈ 5°.

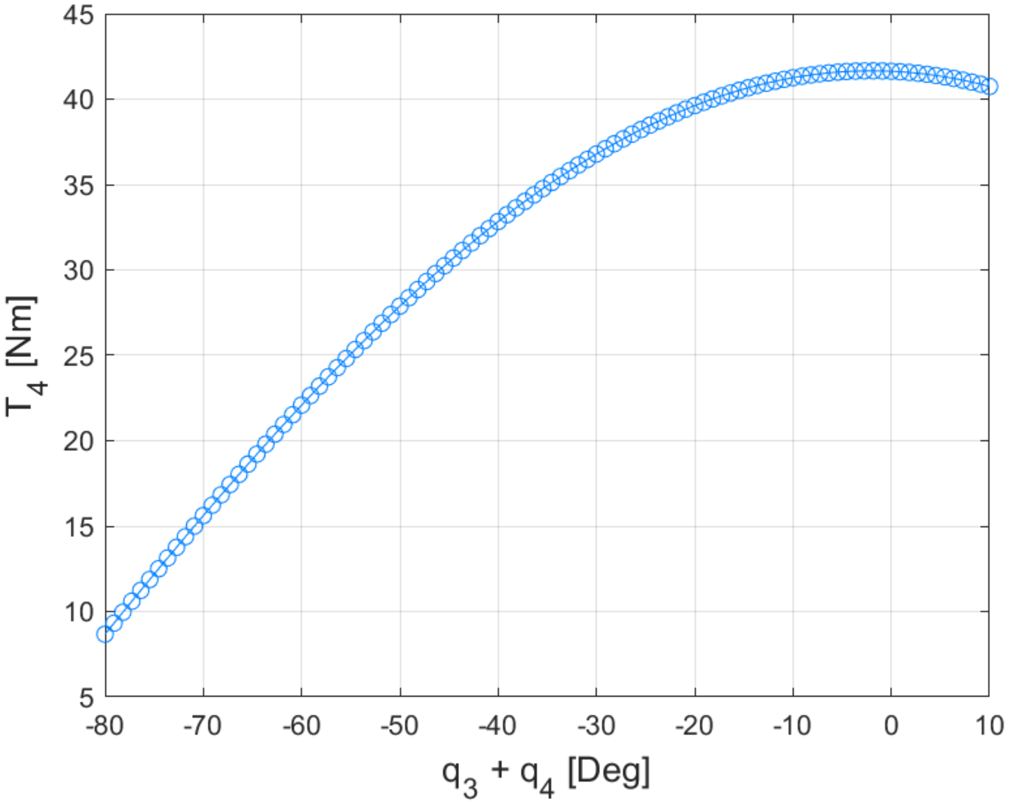


Figure 94 Plot of $T_4(q_3 + q_4)$.

7.2.2 Torque around J3

The same approach is followed to calculate the torque with respect to joint J3. In this case, during the measurements in SolidWorks, the manipulator was positioned with an angle $q_3 = 90°$. The next step is to calculate the angle θ_3 , which represents the angle of the line of length L_3 (connecting the center of mass point G_3 to Joint 3) with respect to the y-axis.

$$\theta_3 = \tan^{-1}\left(-\frac{22.4}{747.2}\right) = -1.72°, \tag{54}$$

$$L_3 = \sqrt{-22.4^2 + 747.2^2} = 747.53\text{mm} = 0.747 \text{ m}, \tag{55}$$

$$T_3(q_3, q_4) = \mathrm{m}_b \, \mathrm{g} \, \mathrm{L}_3 \cos(\mathrm{q}_3 + \theta_3) + \mathrm{m}_\mathrm{c} \, \mathrm{g} \, (\mathrm{b} \cos(\mathrm{q}_3) + \mathrm{L}_4 \cos(\mathrm{q}_3 + \mathrm{q}_4 + \theta_4)), \tag{56}$$

where b is the length of link b.

The torque calculated in (56) accounts for both the mass m_b and m_c, and depends on the variables q_3 and q_4. To illustrate the torque curve relative to Joint 3 as a function of q_3, Figure 95 presents four curves, each corresponding to a different value of $q_3 + \; q_4$. The selected values, shown in the Figure, were chosen to cover the two extremes necessary for the full operational space, along with two intermediate values. The range of q_3 was selected to include the maximum and minimum values required to reach the entire operational space, with an additional ≈5° threshold.

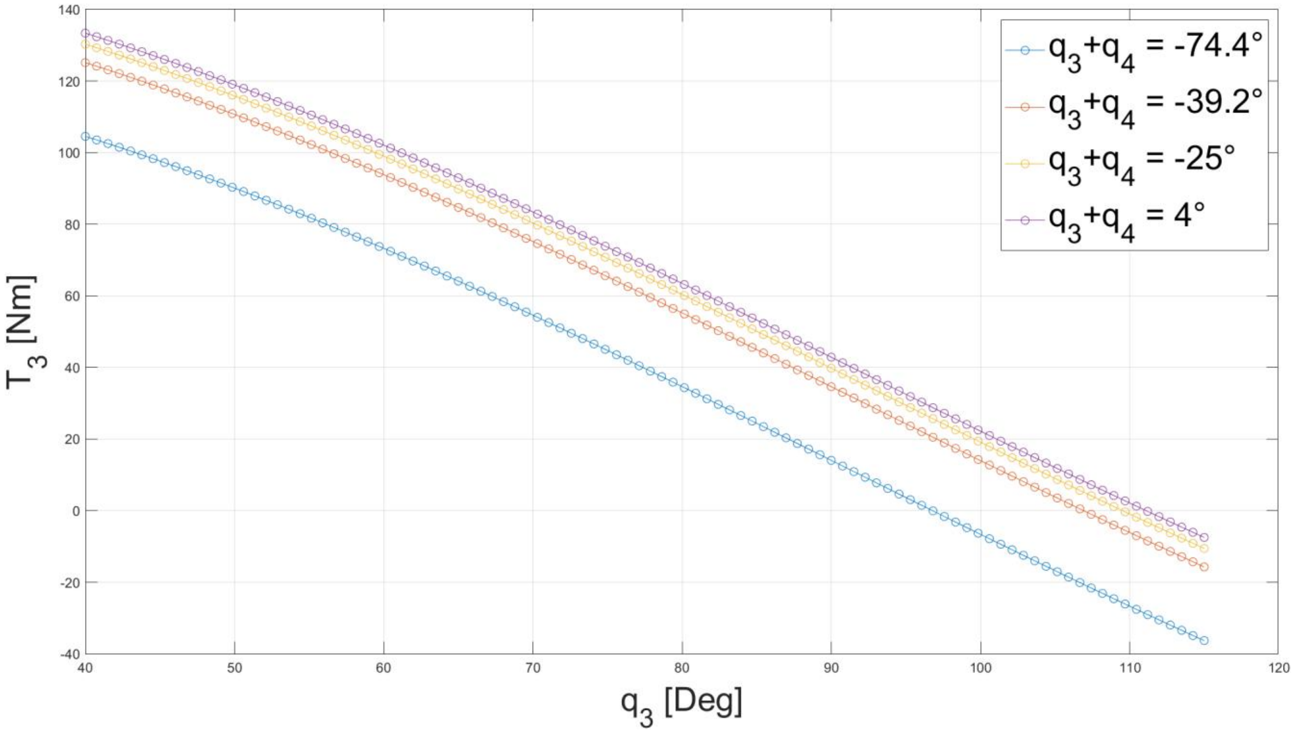


Figure 95 Plot of $T_3(q_3)$.

In Figure 96 are plotted the Torques $T_3(q_3)$ around J3 considering the maximum and the minimum contribution of the mass m_c. The maximum and minimum contribution of the mass m_c occur, respectively, when:

$$\cos(q_3 + q_4 + \theta_4) = 1, \tag{57}$$

$$\min(|\cos(q_3 + q_4 + \theta_4)|). \tag{58}$$

The maximum torque:

$$\mathrm{T}_{3_{max}}(q_3) = \mathrm{m}_b\, \mathrm{g}\, \mathrm{L}_3 \cos(\mathrm{q}_3 + \theta_3) + \mathrm{m}_\mathrm{c}\, \mathrm{g}\, (\mathrm{b} \cos(q_3) + \mathrm{L}_4). \tag{59}$$

The minimum torque $T_{3_{min}}$ is plotted by considering $q_3 + q_4 = -80°$, taking into account a threshold of ≈ 5° with respect to the minimum value to reach the operational space.

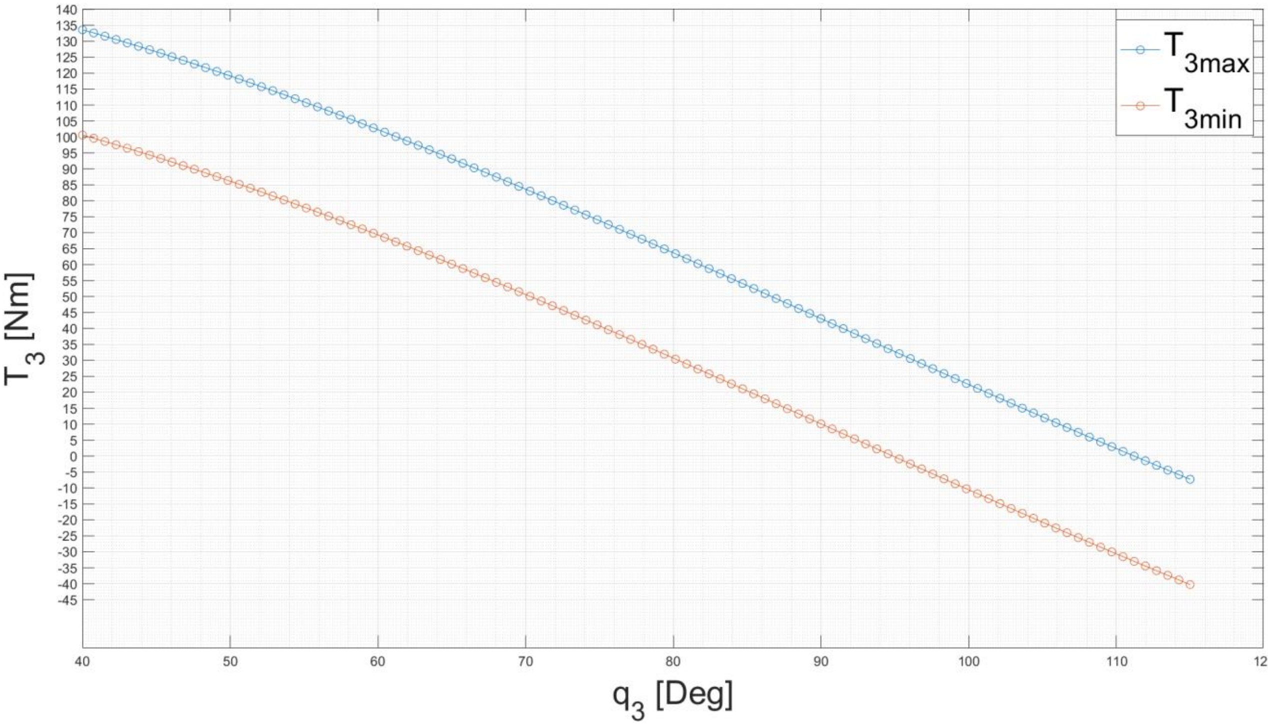


Figure 96 Plot $T_3(q_3)_{max}$, $T_3(q_3)_{min}$.

7.3 Balancing of the manipulator

7.3.1 Balancing system for $\boldsymbol{T_4}$ using a linear spring and a circular pulley

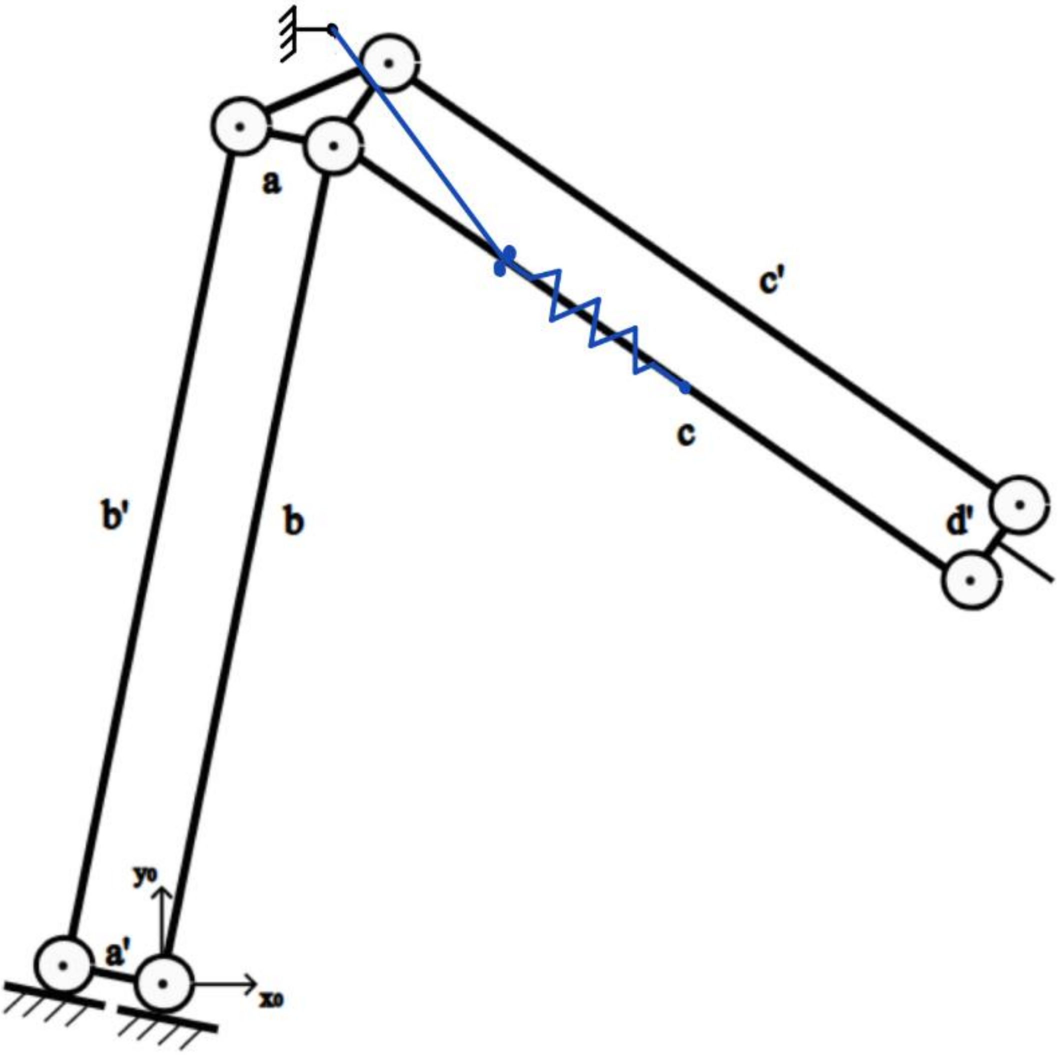


Figure 97 Balancing system for T_4 using a linear spring and a circular pulley.

Let us derive the general expression for T_e as a function of $q_3 + q_4$:

$$-T_e(q_3 + q_4) = T_4(q_3 + q_4),$$

$$p\,k\,h\cos(q_3 + q_4) = m_c\,L_4\cos(q_3 + q_4).$$

The spring constant:

$$k = \frac{m_c\, g\, L_4}{p\, h}.$$

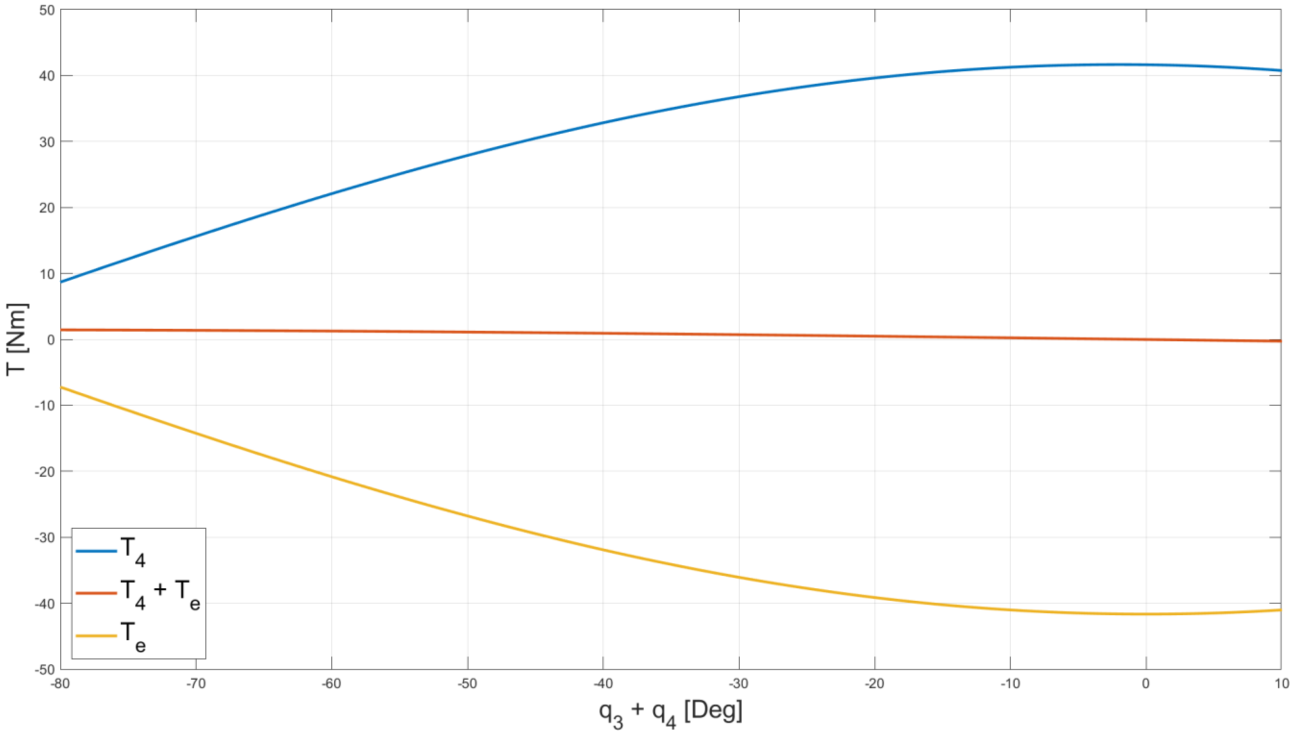


Figure 98 Resulting torque after gravity compensation.

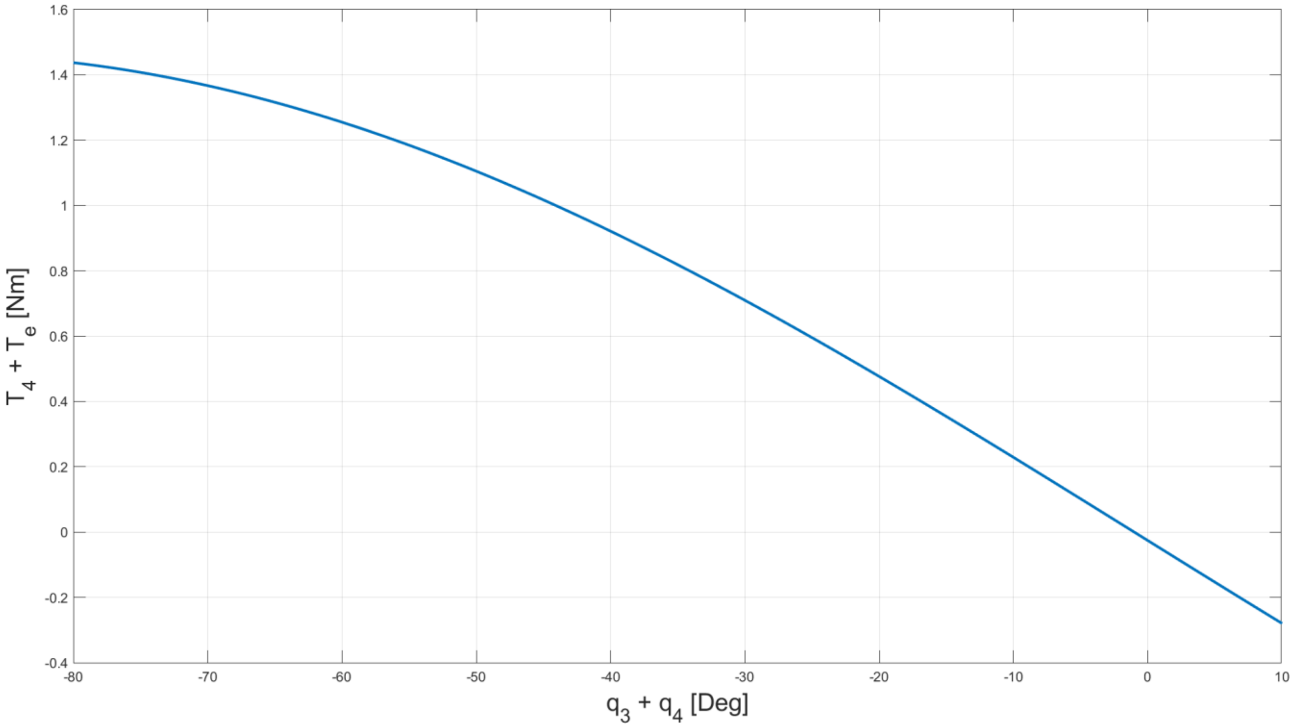


Figure 99 Resulting torque after gravity compensation.

The torque is not perfectly balanced due to the angle θ_4, that has been neglected in the equations. This balancing system, although functional, has two drawbacks: the spring and pulley system installed on the link increase its weight, and, additionally, the constraint of $p = h$ requires us to create a component of relatively high height h to which the spring is attached. Therefore, we proceed with the calculation of the value of k by choosing different values of h and p, with $h = p$. In Table 18 are listed values of k in function of $a = h$, and the respective values $l_{s_{max}}$ computed when $l_s(-80°)$.

Table 17 k , $l_{s_{max}}$ and F_{max} at different values of $h = p$.

$h = p = 0.2m$	$k = 1041 \frac{N}{m}$	$l_{s_{max}} = 0.3985\ m$	$F_{max} = 415\ N$
$h = p = 0.3m$	$k = 462.8\ \frac{N}{m}$	$l_{s_{max}} = 0.5977\ m$	$F_{max} = \ 276.6\ N$
$h = p = 0.4m$	$k = 260.35\ \frac{N}{m}$	$l_{s_{max}} = 0.7970m$	$F_{max} = \ 207.5\ N$
$h = p = 0.5m$	$k = 166.6\ \frac{N}{m}$	$l_{s_{max}} = 0.9962\ m$	$F_{max} = \ 166\ N$

Proceeding with the spring sizing, MISUMI's website provides a customizable tension spring, where it is possible to select parameters such as free length L_0, outer diameter, and wire thickness. For each dimension available in the catalog, reference values of spring constant k_r and maximum deflection factor mdf are provided. An iterative approach is employed to identify a spring that effectively balances the system, with a focus on minimizing both size and weight. Given the very high maximum allowable deformation l_s, only the first two cases listed in the table are considered. The iterative spring calculation method was applied across several spring specifications. However, the most effective calculation approach is detailed below to highlight the optimal result.

No configuration was found in the catalog that allows the system to be balanced with a single spring. Therefore, multiple springs in parallel will be used to achieve the desired balance. The goal is to minimize the number of springs used in parallel.

The formulas provided by MISUMI to compute the spring constant k and the maximum deflection:

$$k\left[\frac{N}{mm}\right] = k_r \frac{10}{L_0} \tag{60}$$

$$l_{s_{max}}[mm] = L_0\ mdf \tag{61}$$

Choosing a spring with outer diameter $D = 28mm$, a wire diameter of $2.6mm$, the parameters are $k_r = 7.45\ \frac{N}{mm}$ and $mdf = 138\%$. We start by considering the case $h = p = 0.3$, with $ls_{max} = 0.598\ m$.

We compute L_0 as:

$$L_0 = \frac{598}{1.38} = 433\ mm$$

$$k = 7.45 \cdot \frac{10}{433} = 172\frac{N}{m}.$$

By mounting 3 springs in parallel the total value of $k = 3 \cdot 172 = 516.16\frac{N}{m}$.

Through the (50) we compute the value of $h = p$ with the updated value $k = 516.16\frac{N}{m}$.

$h = p = 0.28m$.

The updated value of $l_{s_{max}} = 0.566\ m$.

In order to reduce the value of L_0, we repeat the steps from the beginning until the value of k is as close as possible to the required value. The algorithm is solved by a C code as follows:

1) We define $k_{req} = \frac{m_c\ g\ L_4}{p \cdot h}$ and $l_{s_{max}}$ with $p = h = 300mm$.
2) We compute L_0 through the (61) and k_{spring} with the (60) considering a product of 3 due to three springs in parallel.
3) We compute the updated value of $p = h = \sqrt{\frac{m_c g L_4}{k_{spring}}}$.
4) We repeat the computation from n°1 with the updated p until $k_{req} = k_{spring}$.

The optimal solution for the spring with outer diameter $D = 28mm$, a wire diameter of $2.6mm$ is: $L_0 =$ 387.5mm, $p = h = 268.4mm$, $k = 192.27\frac{N}{m}$, $k_{tot} = 3\,k = 576.81\frac{N}{m}$.

7.3.2 Balancing system for $\boldsymbol{T_4}$, cam design

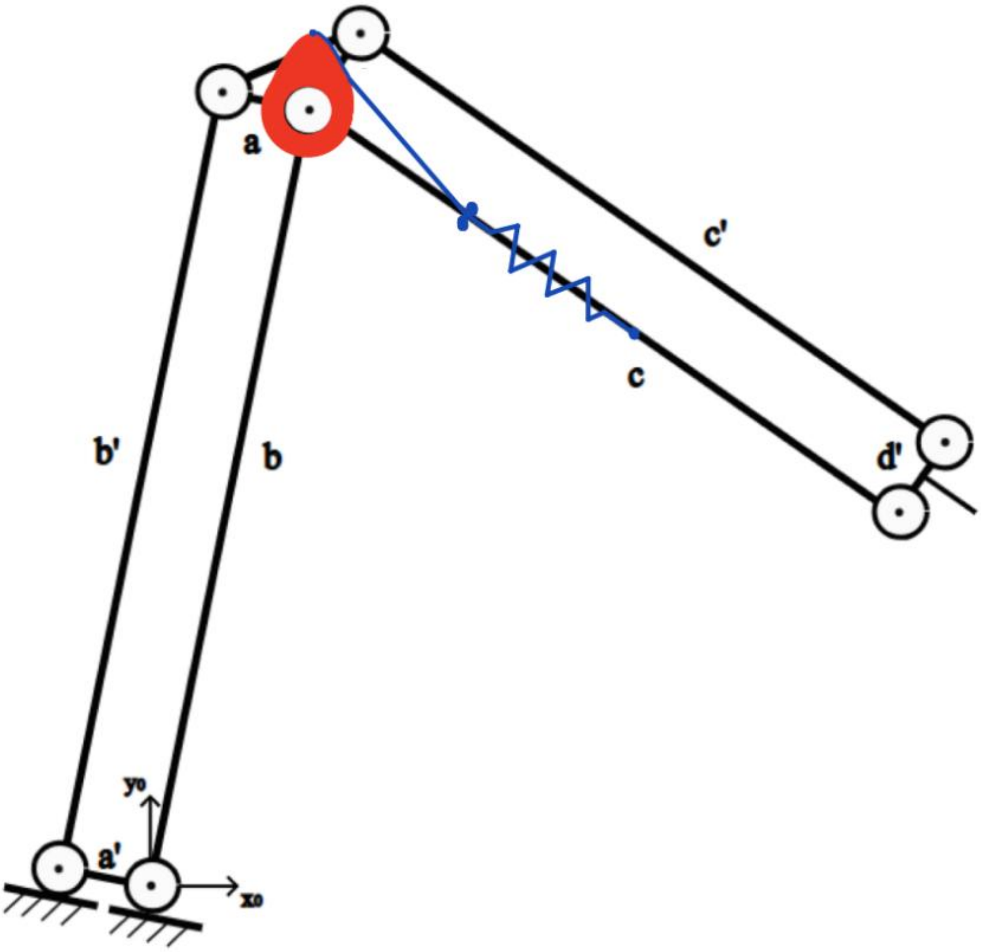


Figure 100 Balancing system for $\boldsymbol{T_4}$, with a linear spring and a cam.

In order to balance the torque $T_4(q_4)$, we implemented on Matlab the algorithm presented in section 7.1.2. The objective is to utilize a commercial linear spring that satisfies the parameters that guarantee point-to-point gravity compensation.

```
m = 6350/1000;  %kg
l = 0.6687; %m


th_min = deg2rad(-80); th_max = deg2rad(10);
th = linspace(th_min, th_max, 300);
```

```
tau = @(in) m*l*9.81*cos(in);

plot(rad2deg(th),tau(th),'LineWidth',1.5); grid on;
xlabel("$\theta$, deg","Interpreter","latex"); ylabel("$\tau$, Nm","Interpreter","latex");


lb = 0.5;
th0 = th_max;
r0 = 0.24;

pB0 = lb*[cos(th0), sin(th0)]';
pA0 = r0*[cos(th0+acos(r0/lb)), sin(th0+acos(r0/lb))]';
l0  = norm(pA0-pB0);

F0 = tau(th0)/r0;

Fn = 22.5;
ln = 0.25;
k  = (F0 - Fn)/(l0-ln);

dth = deg2rad(0.01);
th = th_min:dth:th_max;
thi = nan(size(th));
pAi = nan(2,length(th));
pBi = nan(2,length(th));
li = nan(size(th));
ai = nan(size(th));

for i = 1:length(th)
    if i == 1
        thi(i) = th0 - dth;
        ai(i) = 0;
        pAold = pA0;
        pBold = pB0;
        lold = l0;
    else
        thi(i) = thi(i-1) - dth;
        ai(i) = 0;
        pAold = pAi(:,i-1);

        pBold = pBi(:,i-1);

        lold = li(i-1);

    end


    pBi(:,i) = lb*[cos(thi(i)), sin(thi(i))]';

    pAi(:,i) = pAold + ai(i)*(pAold-pBold);


    fun = @(x)funParm(x,tau(thi(i)),pAold,pBi(:,i),pBold,lold,l0,F0,k);


    [ai(i),fval] = fzero(fun,ai(i));


    pAi(:,i) = pAold + ai(i)*(pAold-pBold);

    li(i) = lold - (norm(pBold - pAi(:,i)) - norm(pBi(:,i) - pAi(:,i)));


end


plot(pAi(1,1:end),pAi(2,1:end),'LineWidth',1.5);axis equal; grid on; hold on

plot([0, pAi(1,1), pBi(1,1),0], [0, pAi(2,1), pBi(2,1),0],'k','LineWidth',0.5);

plot([0, pAi(1,end), pBi(1,end),0], [0, pAi(2,end), pBi(2,end),0],'k','LineWidth',0.5);

dlmax=li(end)-ln

mdf=0.76;

dlspringmax=ln*mdf

kr= 19.65 ;

kspring=kr*10/(ln)



function out = funParm(x,taui,pAold,pB,pBold,lold,l0,F0,k)

    pAlpha = pAold + x*(pAold-pBold);

    li = lold - (norm(pBold - pAlpha) - norm(pB - pAlpha));

    Fsi = (li-l0)*k + F0;

    m = (pB(2) - pAlpha(2))/(pB(1) - pAlpha(1));

    q = pAlpha(2) - (pB(2) - pAlpha(2))/(pB(1) - pAlpha(1))*pAlpha(1);

    ri = abs(q)/sqrt(1+m^2);

    out = +taui - Fsi*ri;

end
```

We start by choosing a spring in the MISUMI catalogue, in order to express the parameters F_n, k_r and mdf which are respectively the preload at the natural length, the standard spring constant and the maximum deflection factor.

We express the maximum deflection of the spring in meters as $dlspring_{max} = l_n \cdot mdf$, and the spring constant of the commercial spring as $k_{spring} = kr \cdot \frac{10}{l_n}$.

The next step is to choose the parameters r_0, l_b, l_n through a trial-and-error approach in order to find the values such that:

$$k \cong k_{spring} \tag{62}$$

$$dl_{max} \leq dlspring_{max}. \tag{63}$$

In this manner, the required value of k computed by the algorithm may be obtained through a commercial spring purchased on MISUMI. By testing various springs, we identified a solution that avoids an overly bulky cam, employing a spring with $l_n = 0.25\ m, F_n = 22.5\ N, k_{spring} = 786 \frac{N}{m}$, $mdf = 0.76$. The values of l_b and r_0 are $l_b = 0.5\ m$ and $r_0 = 0.24\ m$. The cam geometry plotted by the algorithm is shown in the Figure 81 . The value of k obtained from the algorithm is $k = 786.86 \frac{N}{m}$.

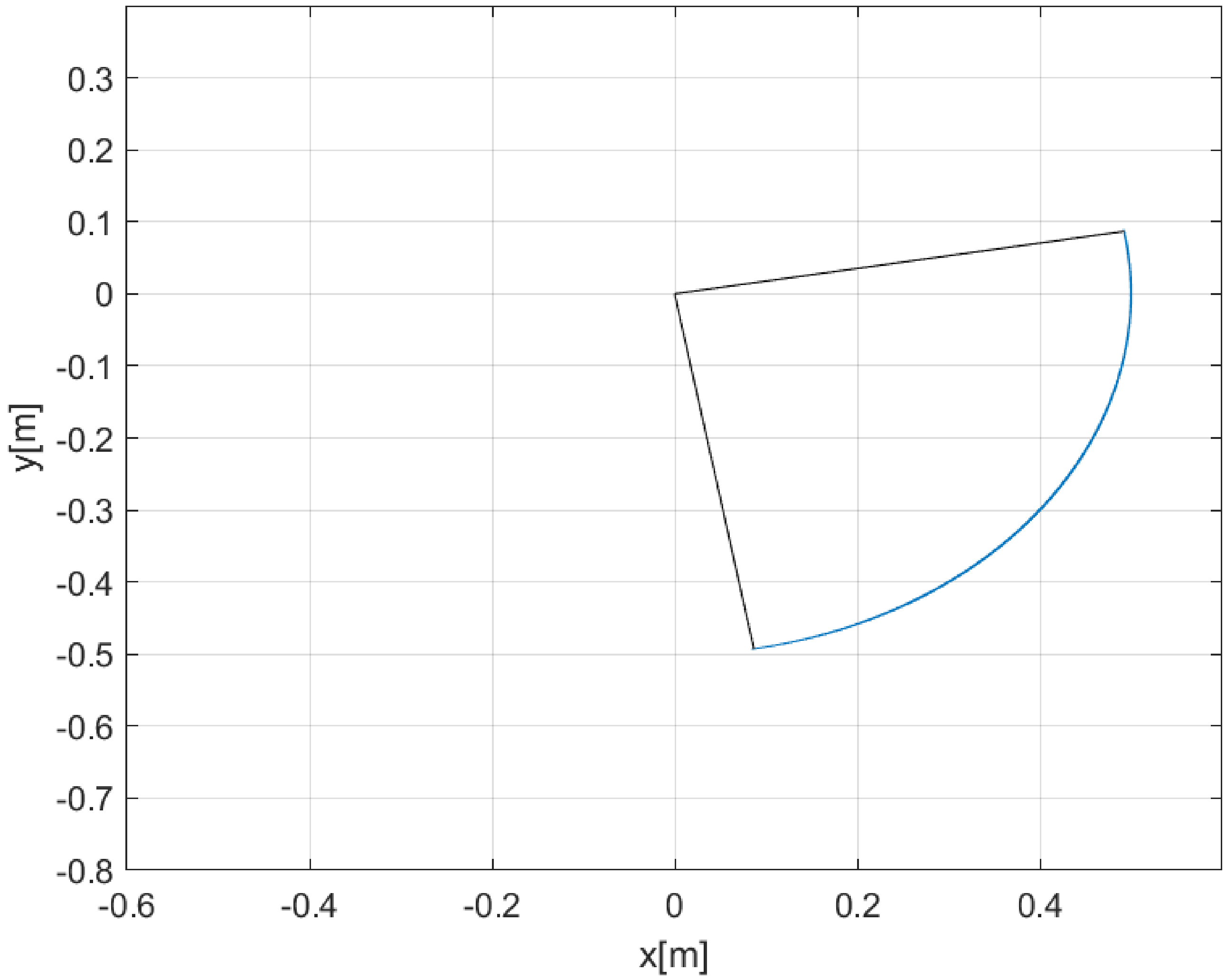


Figure 101 Plot of the cam design with the graphical algorithm.

7.3.3 Balancing system for $\boldsymbol{T_3}$ using a linear spring and a circular pulley

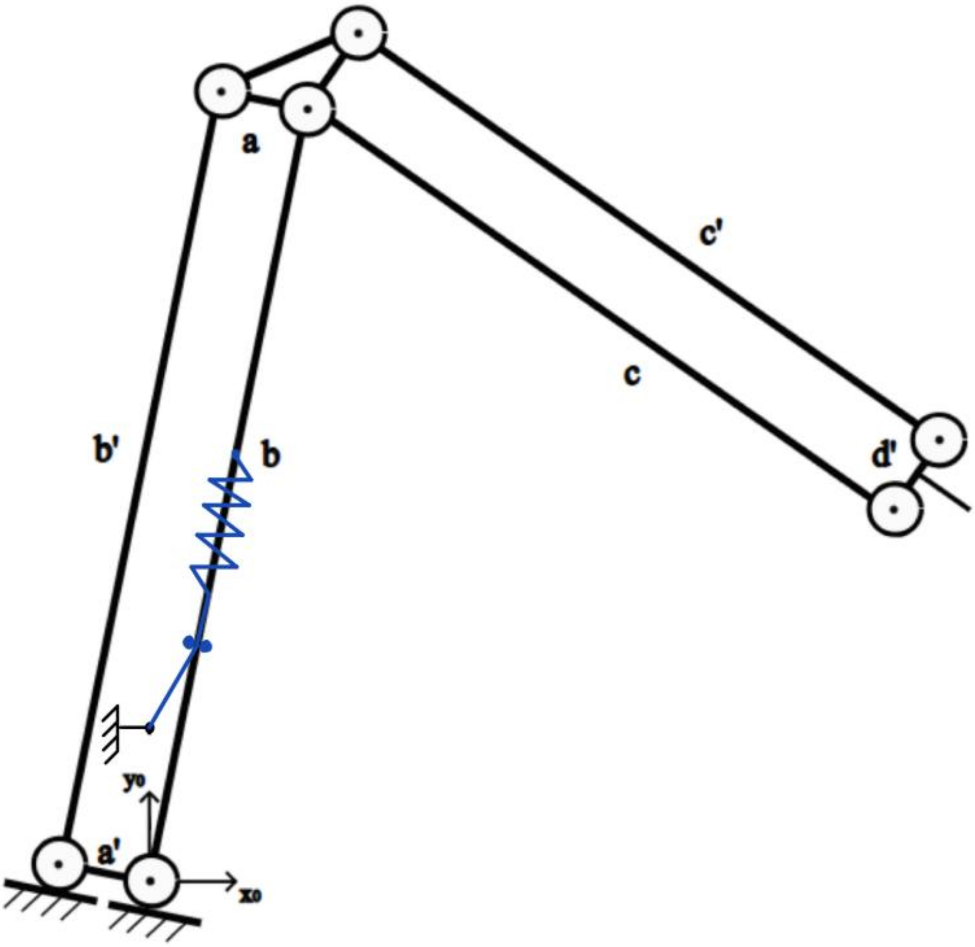


Figure 102 Balancing system for T_3 using a linear spring and a circular pulley.

Given that the torque T_3 depends on the two joint variables q_3 and q_4, it becomes challenging to achieve a static balance of link b for every possible position. As shown in Figure 95, the torque $T_3(q_3)$ does not cross 0 Nm when $q_3 = 90°$. Consequently, the balancing model with a linear spring and a circular pulley presented in chapter 7.1.1 may not ensure a point-to-point balance of the system. However, the following solution is proposed to partially compensate for torque $T_3(q_3, q_4)$.

Starting from equation (50), we choose a value of $p = h$, while the spring constant k is determined through a trial-and-error approach with the aim of closely approximating the variation of $T_3(q_3)$, thereby ensuring that the resulting $T_3 - T_e$ remains as steady as possible. The results are shown in Table 19.

Table 18 k , $l_{s_{max}}$ and F_{max} at different values of $h = p$.

$h = p = 0.25m$	$k = 1917\frac{N}{m}$	$l_{s_{max}} = 0.25m$
$h = p = 0.3m$	$k = 1331\frac{N}{m}$	$l_{s_{max}} = 0.3m$
$h = p = 0.35m$	$k = 978\frac{N}{m}$	$l_{s_{max}} = 0.35m$
$h = p = 0.4m$	$k = 749\ \frac{N}{m}$	$l_{s_{max}} = 0.4\ m$

The balancing of T_3 is plotted in Figure 83, that shows the resulting balanced torques $T_{3_{max}}, T_{3_{min}}$.

For the all the possible values of and q_4 within the operational space, $T_3(q_3)$ lies between the two curves shown in Figure 103. More precisely: 8.52 $Nm \leq T_3(q_3, q_4) \leq 43.38\ Nm$.

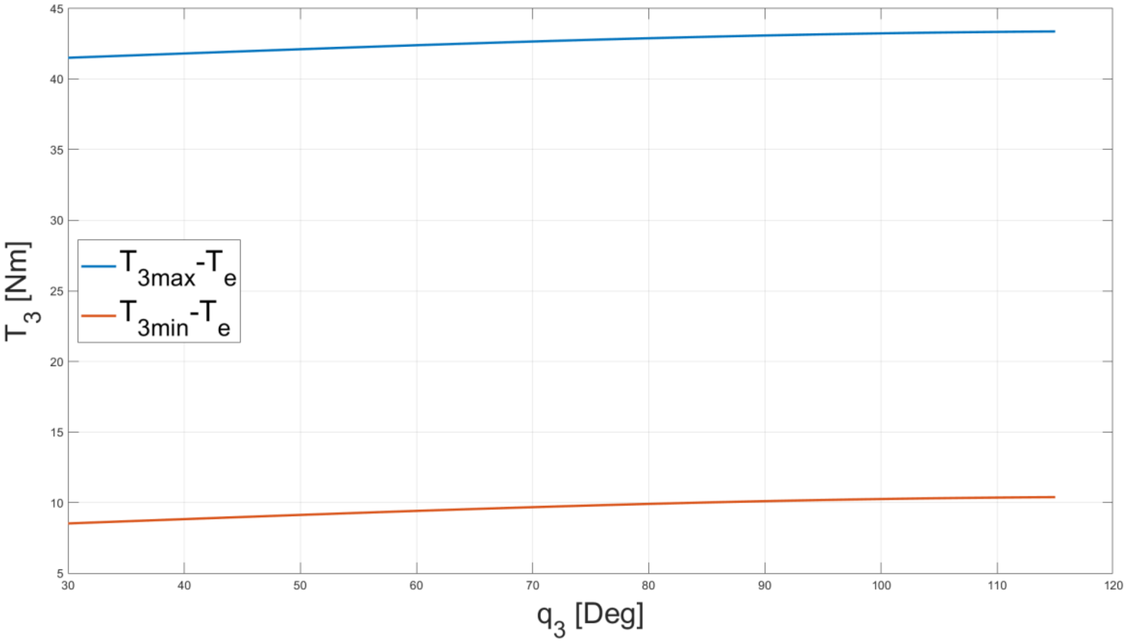


Figure 103 T_3 reduction after gravity compensation.

By consulting the Misumi catalogue, we choose the commercial linear spring WFSP25-2.9-528 to balance the torque $T_3(q_3, q_4)$. Through the algorithm presented in chapter 7.3.1, a suitable solution is found with the parallel of two springs with parameters: $L0 = 527.76\, mm,\ h = p = 401.1\, mm,$ $k = 372.3\, mm.$

7.3.4 Summary of gravity compensation

Each active joint is balanced though a linear spring and a circular pulley, whilst joint J4 is also balanced with a cam-belt system.

- **Torque $\boldsymbol{T_4(q_4)}$**: two solutions are presented. Balancing with the cam and the linear springs seems to be the best solution in terms of bulk reduction, due to the variable radius r_i of the cam. The features of the cam allow for the gravity compensation of link c to be balanced through a single linear commercial spring instead of 3 springs for the best solution using only the linear spring and the circular pulley. Ultimately, the cam design enables the minimization of the mass and of the dimensions of the overall balancing system.
- **Torque $\boldsymbol{T_3(q_3, q_4)}$**: the balancing of the torque T_3 depends on two joints variables, resulting more tricky to find a solution for the gravity compensation. The solution expressed with the linear spring and the circular pulley allows to reduce the torque. In particular, the maximum resulting torque $T_3 - T_e$ is equal to $43.38 \frac{N}{m}$. The transmission at joint J3 must guarantee a transmission ratio of at least 4.4. Another possible solution would be to shift the center of mass of the mass m_c to Joint 4 using counterweights, so that the torque T_3 becomes a function of only the angle q_3 . In Chapter 7.4, the manipulator's configuration was modified in such a way that both joints are actuated from the base, thereby making each torque dependent on a single joint variable.

7.4 Double parallelogram configuration with 2 active joints on the base.

Since the torque $T_3(q_3, q_4)$ may not be entirely balanced through the approach analyzed in chapter 7.3.3, a solution that adds modification to the designed manipulator is proposed. In Figure 104 is shown the double parallelogram configuration with the addition of links b''', e and e', that allows both the joints J1 and J2 to be actuated on the base. J1 allows the rotation of links b and b', while the link c is actuated by J2 through the articulated transmission composed by the links e, b''' and e'. Note that the angle between links e′ and c is constant and equal to π.

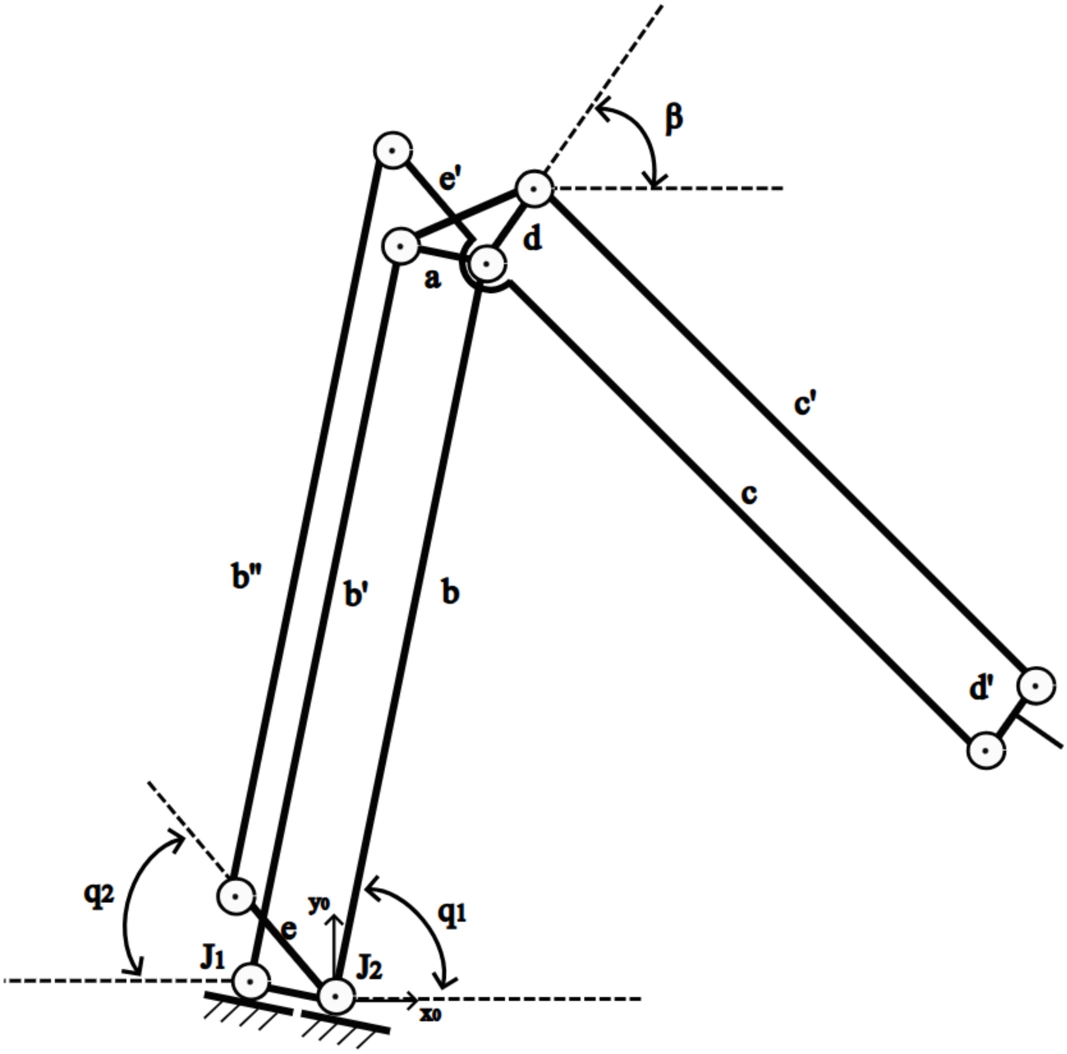


Figure 104 Manipulator configuration with 2 active joints on the base.

The displacement of the end effector with respect to frame {0} is expressed by the translation vector

$$t_e^0 = \begin{bmatrix} b\cos(q_1) + c\cos(-q_2) + \frac{d}{2}\cos(\beta) \\ b\sin(q_1) + c\sin(-q_2) + \frac{d}{2}\sin(\beta) \\ 0 \end{bmatrix}.$$

The kinematics model is equivalent to the one treated in Chapter 3, with the only difference been that the orientation of the link c with respect to frame {0} is defined by only the variable q_2 instead of the sum $q_1 + q_2$. The solutions of the inverse kinematics is unvaried for q_1, while the values of q_2 in Figure 105 are equivalent to $-(q_1 + q_2)$ computed in Chapter 3. The updated values necessary to reach the vertices of the workspace are listed in Table 20.

Table 19 Updated operational space boundaries.

x	y	q_1	q_2
0.5m	0.3m	78.7°	74.4°
1.5m	1.1m	60.1°	-4°
0.5m	1.1m	112.4°	0.2°
1.5m	0.3m	46.9°	40.2°

7.4.1 Static model

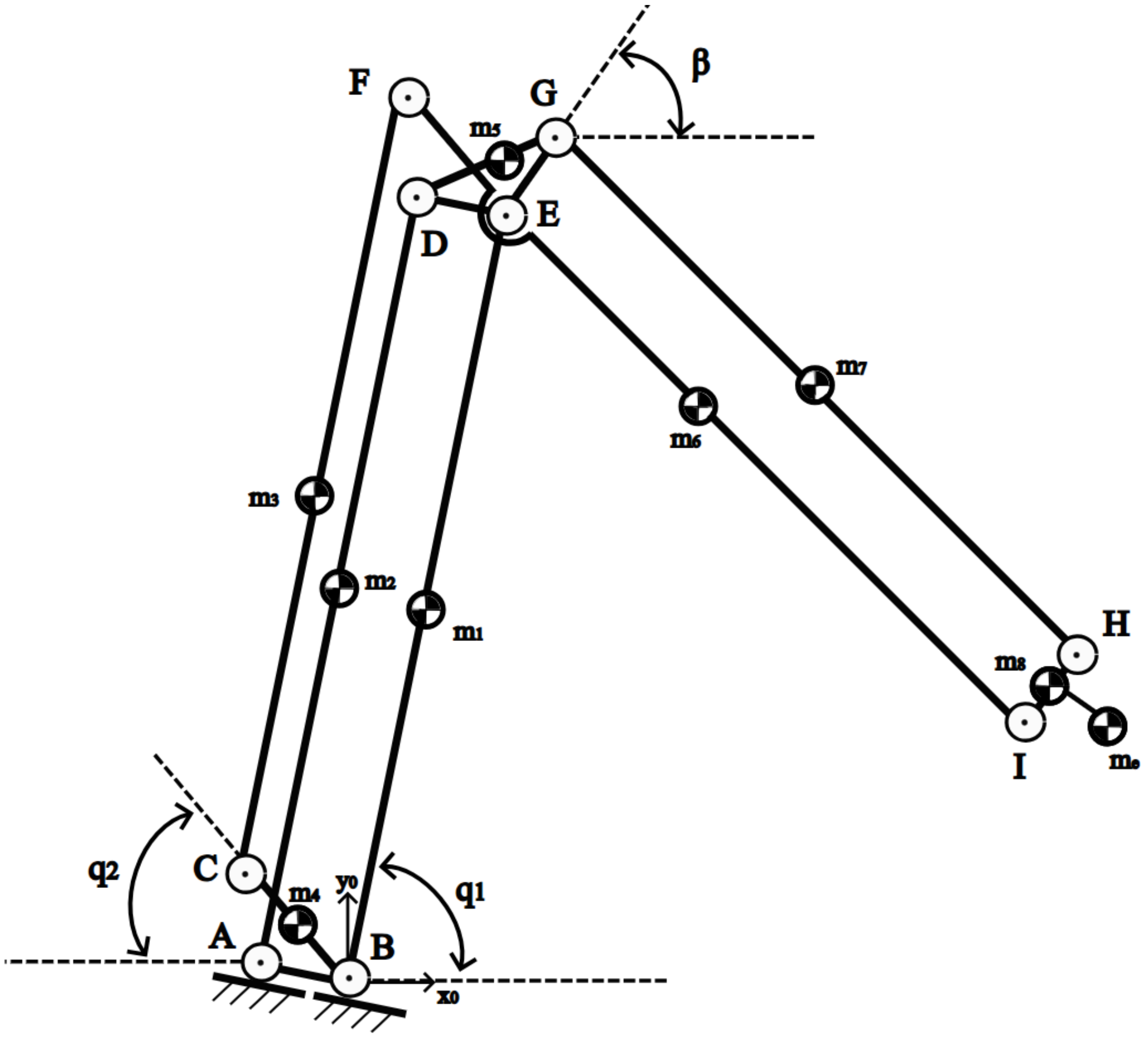


Figure 105 Static model of the manipulator with two active joints on the base.

We define p_i as the position of the center of mass of mass m_i with respect to frame {0}, and P_i as the position of Joint i, with $i = [A, B, C, D, E, F, G, H, I]$. Then we define:

$l_1 = \overline{Bp_1},$

$l_2 = \overline{Ap_2},$

$l_3 = \overline{Cp_3},$

$l_4 = \overline{Bp_4},$

$p_5 = \left[l_{5x}, l_{5y}\right]^T,$

$p_6 = \left[l_{6x}, l_{6y}\right]^{\mathrm{T}},$

$l_7 = \overline{Gp_7},$

$p_8 = \left[l_{8x}, l_{8y}\right]^{\mathrm{T}}.$

The centers of mass of the system are:

$p_{m1} = P_B + l_1\,[c_1, s_1]^{\mathrm{T}},$

$p_{m2} = P_A + l_2\,[c_1, s_1]^{\mathrm{T}},$

$p_{m3} = P_C + l_3[c_1, s_1]^{\mathrm{T}},$

$p_{m4} = P_B + l_4\,[c_2, s_2]^{\mathrm{T}},$

$p_{m5} = P_E + \left[l_{5x}, l_{5y}\right]^T,$

$p_{m6} = P_E + l_6[c_{-2}, s_{-2}]^T,$

$p_{m7} = P_G + l_7[c_{-2}, s_{-2}]^T,$

$p_{m8} = P_I + \frac{\overline{IH}}{2}\left[c_\beta, s_\beta\right]^T,$

$p_{ee} = P_I + \left[l_{eex}, l_{eey}\right]^T.$

The geometry of the system:

$\overline{BE} = \overline{AD} = \overline{CF} = b,\ \overline{AB} = \overline{ED} = \overline{EG} = \overline{HI} = d,\ \overline{EI} = \overline{GH} = c,\ \overline{BC} = \overline{EF} = e.$

The torque $\tau = [\tau_1, \tau_2]^T = \sum_i J_i^T\,(q_i)\,F_i$, with $F_i = [0, m_i\,g]^T$.

$$\tau = \begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix} = \begin{bmatrix} -\big(l_1m_1 + l_2m_2 + l_3m_3 + b(m_5 + m_6 + m_7 + m_8 + m_{ee})\big)g\cos(q_1) \\ -g\left((l_4m_4 + e\,m_5)\cos(q_2) + \left(l_6m_6 + l_7m_7 + \left(c + \frac{d}{2}\cos(\beta)\right)m_8 + (c + l_{eex})m_{ee}\right)\cos(-q_2)\right)\end{bmatrix}. \tag{64}$$

Each torque $\tau_1(q_1)$ and $\tau_2(q_2)$ is dependent on solely one joint variable, thus enabling a point-to-point balancing compensation for both torques. The double-parallelogram manipulator with the articulated transmission can be balanced through the systems proposed in chapter 7.1.1 and 7.1.2.

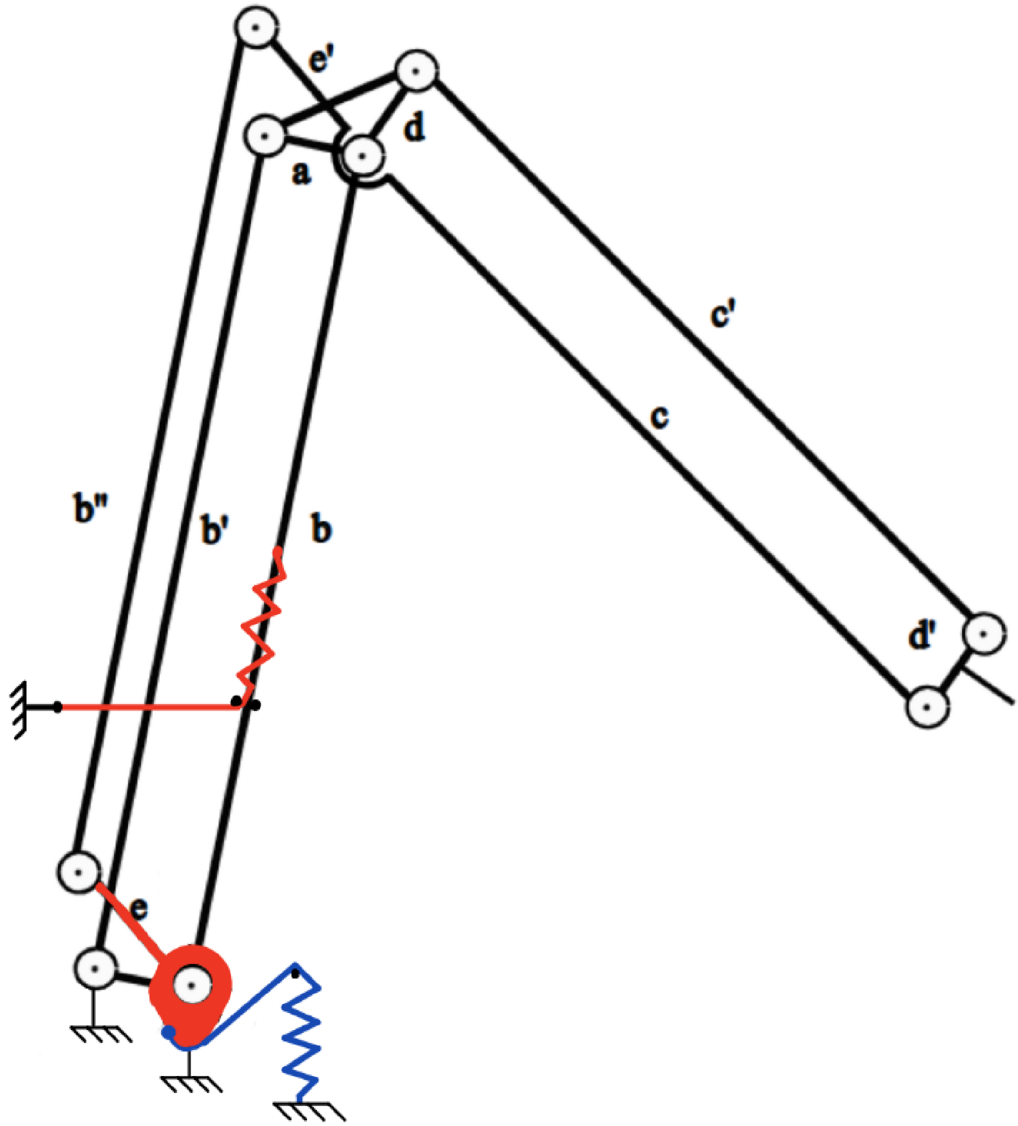


Figure 106 Balancing systems for the manipulator with 2 active joints on the base.

8 Conclusions and suggestions for future improvements

The objective of this thesis is to design a robotic arm for automatic wine sampling in a winery. The designed robotic arm consists of a 4-DOF manipulator with a double parallelogram configuration, which passively maintains a constant orientation of the end effector due to the parallelogram structure. The simplified model features an anthropomorphic manipulator mounted on a linear guide. The active joints are the linear guide J1, the vertical rotational joint J2, the shoulder J3, and the elbow J4. The robotic arm is firstly sized in order to reach the entire operational space, with the links $b = 1.14m$ and $c = 0.9m$. The shortest links of the parallelograms are all sized equal to $0.12\ m$. Through the chosen dimension the manipulator workspace is large enough to contain all the target points of the operational space.

The work proceeds with a kinematic analysis, solving both the forward and inverse kinematics problems to determine the joint variables required to reach the workspace boundaries. A dexterity analysis is conducted to identify the regions where the manipulator operates efficiently. The results show that singular configurations occur at the workspace boundaries, but by virtue of a proper dimensioning process, the operational space maintains a high level of dexterity.

The manipulator is designed using SolidWorks, with customized joint components that serve both as motor housing and pin supports. The pins are selected from the MISUMI catalog. Each link consists of two forks and a tube, where the forks are equipped with bearings and coupled with the pins to create the hinges. The tube diameters are set to 40 mm for links attached to active joints and 16 mm for those attached to passive joints. The motors are sourced from ROBOTIS, specifically the XH540-W270-T model.

A transmission is designed for active revolute joints. Regarding the joint $J2$, the relative rotation is enabled by a slewing ring with a toothed outer ring and a gear mounted on the rotor. On the other hand, for both joints $J3$ and $J4$ the transmission consists of the driven gear attached to the fork and the driving gear attached to the rotor. Once the manipulator is totally assembled in SolidWorks, the mechanical constraints are verified to ensure that the manipulator may reach the entire operational space. In particular, the joint $J2$ allows for a rotation equal to $360°$, while for joints $J3$ and $J4$ the following constraints are valid : $40° \leq q_3 \leq 117.27°, q_4 \geq -158°$, $q_3 + q_4 \leq 15.22°$. Ultimately, the designed manipulator is able to reach the entire operational space.

The model is then simulated in the Simulink environment. The simulation setup involves automatically generating the plant from SolidWorks and importing it into Simulink using the Simscape Multibody Link. The primary goal of the simulation is to verify the correct behavior of the manipulator and to begin outlining a control logic. Two control models are designed to simulate the task execution: a comparison between joint-space control and operational-space control is discussed. However, the models developed using Stateflow contain flaws and do not yet implement a rigorous trajectory control algorithm suitable for real-world execution. While the controllers correctly simulate the achievement of target positions and implement basic control logic, the trajectory path needs further refinement. The controller shall receive as an input the signals about if the mobile platform is moving or not, about the coordinates of the tap to be reached, and the information about the correct extraction from the barrels.

The project concludes with a detailed study of balancing system methodologies to ensure static gravity compensation. Two general methods are explored: one using a linear spring and a circular pulley, and another involving a cam and spring system. Both methods are applied to the designed manipulator. For J4, the cam-based system proves to be the best solution in terms of space optimization due to the variable radius r_i of the cam. This design allows gravity compensation of link c with a single commercial linear spring, compared to the three springs required in the best alternative using only a linear spring and circular pulley. The cam-based approach effectively reduces the overall mass and dimensions of the balancing system.

For joint $J3$, the balancing of the torque T_3 depends on two joints variable, resulting more tricky to find a solution for the gravity compensation. The solution expressed with the linear spring and the circular pulley allows to reduce the torque, in particular the maximum resulting torque $T_3 - T_e$ is equal to $43.38\frac{N}{m}$. The transmission at joint J3 shall guarantee a transmission ratio of at least 4.4. Another possible solution to fully balance the torque T_3 is to shift the center of mass of the mass m_c to Joint 4 using counterweights, so that the torque T_3 becomes a function of only the angle q_3. A structural modification is also proposed: adding an additional parallelogram to the existing design allows both J3 and J4 to be actuated from the base. This configuration simplifies torque equations, making them dependent on a single joint variable. Using the two previously discussed balancing methods, this design achieves full gravity compensation and emerges as the best solution for completely balancing the manipulator.

Regarding the necessary future improvements, the project should continue with the design of the end effector, which plays a crucial role in determining the internal diameter of the tubes forming the links. Its weight significantly affects the torques and, consequently, the stress on the tubes and forks. Once the end effector's characteristics are defined, appropriate 3D printing materials for the SolidWorks components can be selected. In this thesis, calculations were performed assuming PLA 3D printing. Additionally, the material for the tubes must be chosen, with aluminum and carbon fiber being viable options. The end effector also influences the balancing system, requiring a decision between maintaining the current configuration and reducing torque or modifying the design to achieve full balance.

Reference List

[1] "https://www.ilsole24ore.com/art/il-mercato-vino-italia-crescera-19-miliardi-2025-anche-grazie-all-online-AERqtB9?refresh_ce".

[2] "https://www.eataly.net/it_it/magazine/guide/come-cucinare/come-si-fa-vino".

[3] "https://www.youtube.com/watch?v=oIxKe6M_C9s".

[4] Z. Liu, K. Peng, L. Han and e. al., "Modeling and Control of Robotic Manipulators Based on Artificial Neural Networks: A Review.," *Iran J Sci Technol Trans Mech Eng 47,* pp. 1307-1347, 2023.

[5] N. Ghodsian, K. Benfriha, A. Olabi, V. Gopinath and A. Arnou, "Mobile Manipulators in Industry 4.0: A Review of Developments for Industrial Applications.," *Sensors,* no. 23, 2023.

[6] H. Engemann, S. Du, S. Kallweit, P. Cönen and H. Dawar, "Omnivil—An autonomous mobile manipulator for flexible production.," *Sensors,* 2020.

[7] I. Garcia and e. al., "Autonomous 4DOF Robotic Manipulator Prototype for Industrial Environment and Human Cooperation," *IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Porto, Portugal,* pp. 1-6, 2019.

[8] H. Afrisal, B. Setiyono, M. S. R. Yusuf and O. Toirov, "Trajectory Planning with Obstacle Avoidance of 3 DoF Robotic Arm for Test Tube Handling System," *7th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE), Semarang, Indonesia,,* pp. 41-46, 2020.

[9] B. Wei and D. Zhang, "A Review of Dynamic Balancing for Robotic Mechanisms," *Robotica,* pp. 55-71, 2021.

[10] G. Endo, H. Yamada, A. Yajima, M. Ogata and S. Hirose, "A passive weight compensation mechanism with a non-circular pulley and a spring," *IEEE International Conference on Robotics and Automation, Anchorage, AK, USA,* pp. 3843-3848, 2010.

[11] H. S. Kim and J. Song, "Multi-DOF Counterbalance Mechanism for a Service Robot Arm," *IEEE/ASME Transactions on Mechatronics,* vol. 19, no. 6, pp. 1756-1763, 2014.

[12] V. van der Wijk and J. Herder, "Synthesis of Dynamically Balanced Mechanisms by Using Counter-Rotary Countermass Balanced Double Pendula.," *ASME. J. Mech. Des,* 2009.