

POLITECNICO DI TORINO |
CALIFORNIA STATE UNIVERSITY
OF LOS ANGELES

Mechatronic Engineering

Mechanical Design of an Autonomous Mobile Robot



CAL STATE LA
CALIFORNIA STATE UNIVERSITY, LOS ANGELES

Supervisor:

Giovanni Bracco

Co-Supervisor:

Marina Mondin

Co-Supervisor:

Paolo Rocco

Student:

Fausto Francesco Lizzio 251878

25/10/2019

Contents

1	Mobile assistant robot	4
1.1	Introduction	4
1.2	Case studies	5
1.3	Focus on the mechanical design	8
2	Innotech project	10
2.1	Project introduction	10
2.2	System Architecture	12
2.3	SLAM	13
2.4	System interface	14
2.5	Mechanical design	15
3	Wheel configuration	16
3.1	Robotic Mobility	16
3.2	Wheel type analysis	18
3.3	Wheel configuration analysis	21
3.4	Differential drive kinematics	30
3.5	Differential drive dynamics	34
4	Powertrain design	39
4.1	Overview on electrical motors	39
4.2	BLDC mechanical characteristic	40
4.3	Motor sizing	43
4.4	Gearmotor choice	46
4.5	Encoder choice	51
5	CAD Modelling	54
5.1	FDM	54
5.2	Printing Material	57
5.3	Design for Manufacturing	58
5.4	Design for assembly	62

5.5	Description of the main components	65
5.5.1	First layer	66
5.5.2	Second layer	69
5.5.3	Base frame	70
5.5.4	Body and head	73
5.6	Base Assembly	73
6	Conclusion	75

Abstract

This thesis regards the development of a mobile assistant robot working at San Diego International Airport. The project is run by Innotech, a newborn start up based in San Diego, CA. An overview of similar case studies and the description of the whole project is provided in the first chapters, where the system architecture is analysed. The main features covered by the project are: SLAM, mechanical design, object detection and system interface. Afterwards, a deeper focus on the mechanical design is presented from three points of view: a comparison between different wheel configuration lead to the choice of a differential drive; the power train design involved the selection of the motors, gearboxes and encoders; the CAD modelling in Solidworks of the whole structure has been carried on in order to 3D print the models and finally rapid prototyping the robot.

1 Mobile assistant robot

1.1 Introduction

Mobile autonomous robotics is experiencing a huge development in the last decades due to a substantial amount of factors: the decreasing cost of high computational power control boards, the higher accuracy of sensors and cameras, the efficiency of new communication protocols, and lastly, a growing acceptance of autonomous robotic systems among an increasing percentage of population. Mobility opens the doors to many possibilities for robots, allowing them to perform new and exciting tasks in a large variety of environments, taking care of either dangerous or repetitive duties. Thanks to mobile autonomous robotics, it is not the environment that has to be adapted to the robot, but it is the latter itself that can perceive and modify the former as desired. This represents a huge step ahead in bringing robotics more and more in our everyday life.

A mobile autonomous robot is a complex system since it has to interact with its surrounding environment, walk through it and modify it as desired. To achieve all what it takes to accomplish these tasks, a very high computational power is needed. Hence, powerful evaluation boards have to be installed, and the trend shown in figure 1 clarify how it is possible to afford such a power.

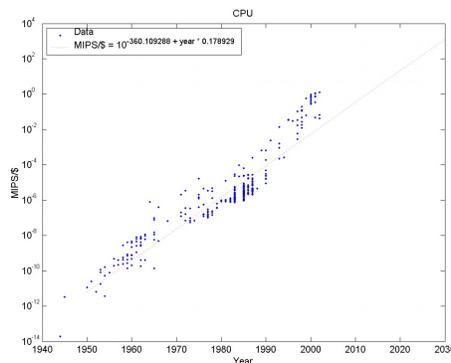


Figure 1 – Processing power available per dollar over time, (1)

The processing power available per dollar has dramatically increased in the last decades, making possible to develop an affordable autonomous robot.

Moreover, one of the most used sensor in autonomous vehicle application, Lidar, is expe-

riencing a steep growth in its market size, as it is possible to see in figure 2. This clearly shows how sensing an unknown environment (one of the autonomous systems' most difficult challenges) is becoming a prominent issue.

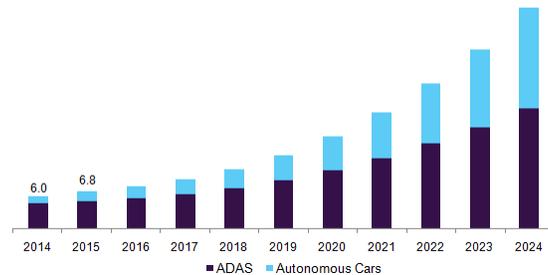


Figure 2 – U.S. automotive LiDAR market by application, 2014 - 2024 (USD Million), (2)

This is leading robotics to spread over new service sectors where autonomous systems can perform better or more efficiently than humans did before. One of the fastest growing sector is the assistant robotics: a huge number of firms and companies are starting to use robots to perform a vast amount of tasks, usually resulting in time and money saving, higher efficiency and a remarkable differentiation among other companies not using these new kinds of systems. The presence of robots frees employees from the need of performing low-level repetitive tasks that would otherwise be carried on by un-spurred workers, thus leading to an increased efficiency. Moreover, the cost a company has to bear to buy a robot is lower than the monthly salary of an employee. Lastly, in a more and more technologized world, a robotic “fleet” is for sure a way for a firm to distinguish itself from the others, reflecting an environment steering towards automation and digitalization.

1.2 Case studies

The development of this project started with a deep analysis of several case studies regarding assistant robots recently launched on the market. This allowed to understand the basic needs and requirements a mobile autonomous system shall have, how to address a specific market target and which kind of technologies are currently implemented in this field.

The first case study to be analysed has been Savioke, (3). It is a firm established in 2014 based in San Jose, California, that has developed a mobile robot, called Relay, able to deliver small items in different environments like hotels, hospitals and factories. It uses sonars,

lidars and stereo cameras for autonomous navigation and obstacle avoidance. When it finds an object or a person along its path, Relay is able to stop and go around the obstacles, in order to continue its own desired path. Through LTE and Wi-Fi it can communicate with hotel customers' phones or with technical support and is equipped with a tablet for a friendly user interface. Relay can even communicate with elevators, thus enabling multi-floor navigation. It is able to reach a charging station when needed and to adjust its own electrical contacts, making the charging process completely autonomous. The items are stored in a lockable drawer that can be opened by a designated app by hotel customers.



Figure 3 – Relay by Savioke

Savioke claims to increase revenue and productivity of the workplace by taking care of the repetitive time-consuming task of item delivery, rising job satisfaction among employees. A great attention is paid to customer experience, to the extent that Savioke defines Relay a "social media magnet", highlighting the sign the robot can leave to customers experiencing such an innovation.

The second case study regards LG's new airport assistant robots, (4). In Seoul's Incheon International Airport, a 57 million travellers per year hub, the company has decided to invest in two different robot concepts.

The left one takes care of airport cleaning, storing information about the most frequently dirty areas of the airport and using them to calculate the most efficient path to take. The one on the right is a guiding robot, roaming passengers to their own gate or to other locations in the airport like restaurants, restrooms, shops or info point by connecting to the airport central station. It is able to speak and understand four different languages (English, Korean, Chinese and Japanese), the most spoken languages in the airport. Moreover, it is equipped with a



Figure 4 – Airport assistant robot by LG

boarding pass scanner that directly takes passengers to their gate if their mother tongue is not one of the aforementioned. Also LG's robots use light detection and ranging (LIDAR) capabilities, sensitive bumpers for detecting obstacles, and simultaneous localization and mapping (SLAM) technology.

Another important case study is provided by an assistant robot working in the Sheraton Hotel in San Gabriel, California, (5). This robot has been developed by Aethon, and its main feature is the ability to carry customers' luggage and take it to their room, navigating through hotel's doors, halls and elevators.



Figure 5 – Carrying luggage Aethon's assistant robot

It is able to communicate with the technical support through Wi-Fi, and to automatically plug itself in for a charge. Navigation is not autonomous, as the robot works on programmed commands only walking through the fixed map of the hotel. It is able to avoid obstacles though, and, if needed, go around them before resuming the way to its task. Aethon, as Savioke, pays great attention on both customer and worker satisfaction, claiming to increase the job quality of the hotel staff by taking care of low level tasks.

The last case study to be analysed has been Knightscope, a company founded in 2013 and based in Mountain View, California. The firm has developed a fleet of four different robot concepts, all devoted to safety management, (6).



Figure 6 – Knightscope Fleet

The robots are enabled with 360 degree HD video streaming, people detection, plate recognition and thermal anomaly detection. These features help security and law enforcement personnel to detect and minimize public injuries and fatalities. When designing the different concepts, Knightscope has targeted very specific hazardous situations, so that customers can choose whichever design best fits to their own needs. The first one on the left is designed for indoor environments, the second one on the left is targeted for outdoor application while the second one from the right is stationary and it is aimed to operate in small indoor environments when motion is not needed. The last one is still under development and will work in uneven multi-terrain applications.

1.3 Focus on the mechanical design

The mechanical design of an assistant robot has to deal with several issues involving the technical functionalities of the system as well as its exterior design. The mechanical engineer has to work in order to achieve a design that can be reliable, efficient, robust against external

factors and at the same time good-looking, not intimidating and friendly also towards those who are not used to technology. It has been showed how people tend not to trust and feel comfortable in the presence of human-looking robots, as they could perceive automation as a threat rather than a helping opportunity. The theory of the Uncanny Valley (7) states that, when designing the exterior of robot, human-resembling features must be avoided. Looking at the curve 7, it is clear how empathy and positive feelings towards a robot grow up until a sudden dramatic decrease is reached.

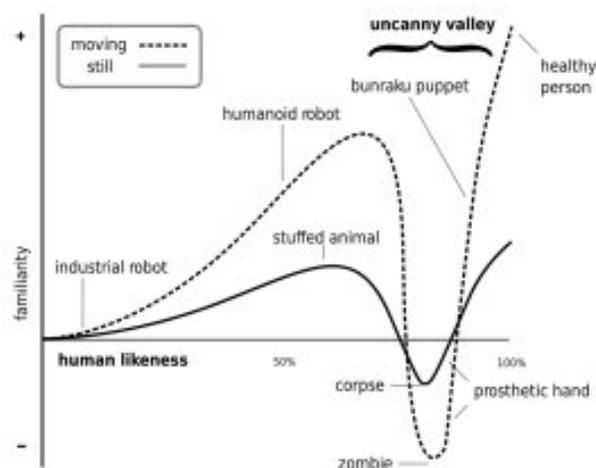


Figure 7 – The Uncanny Valley, (7)

This happens when a reference to the human body in whichever part of the robot is made too clear. That is why features like arms, clear faces or human-like motions must be avoided. Indeed, it is possible to notice how no one of the aforementioned companies used these features in their design: most of them present a cylindrical shape with no robotic arms, and a larger diameter at the bottom part in order to give the sense of a stable and solid design. The user interface is realized just with the use of a tablet in the upper part of the robot, avoiding the use of realistic face-resembling features. The height goes from 90 cm to 1.30 m, in order to be shorter than an average person, so that the user can actually feel a physical sense of power over the robot.

From a more technical point of view, it is helpful to analyse the internal structure of the described robots: it usually comprises different horizontal layers hosting the electronic components and the sensors. A delightful picture of the interior of Relay is reported in figure

8.

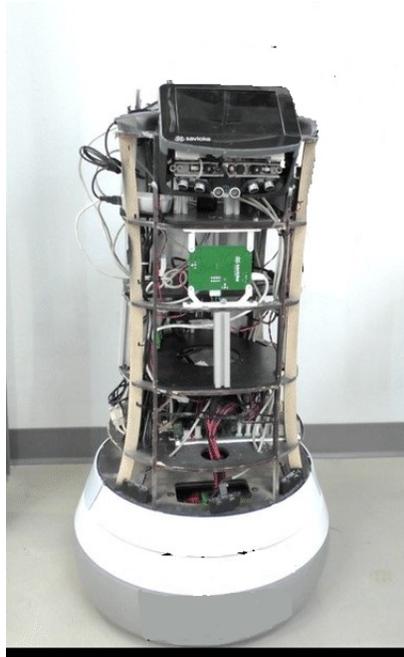


Figure 8 – Internal structure of Relay

It is possible to notice how the robot follows a modular design. Specifically, it is composed by:

- A base unit hosting the drive train, i.e., all what is needed for motion: wheels, motors, gears, encoders and batteries;
- A mid body divided in horizontal layers hosting all the electronic boards, fuse boxes, controllers, sensors, antennas and the optical communication nodes;
- An upper layer hosting the tablet with touch controlled graphic user interface or voice recognition.

2 Innotech project

2.1 Project introduction

The project described by this thesis is run by Innotech, a newborn start up based in San Diego, California. The company aims at developing a mobile assistant robot for San Diego

International Airport, one of the busiest airport on the West Coast.



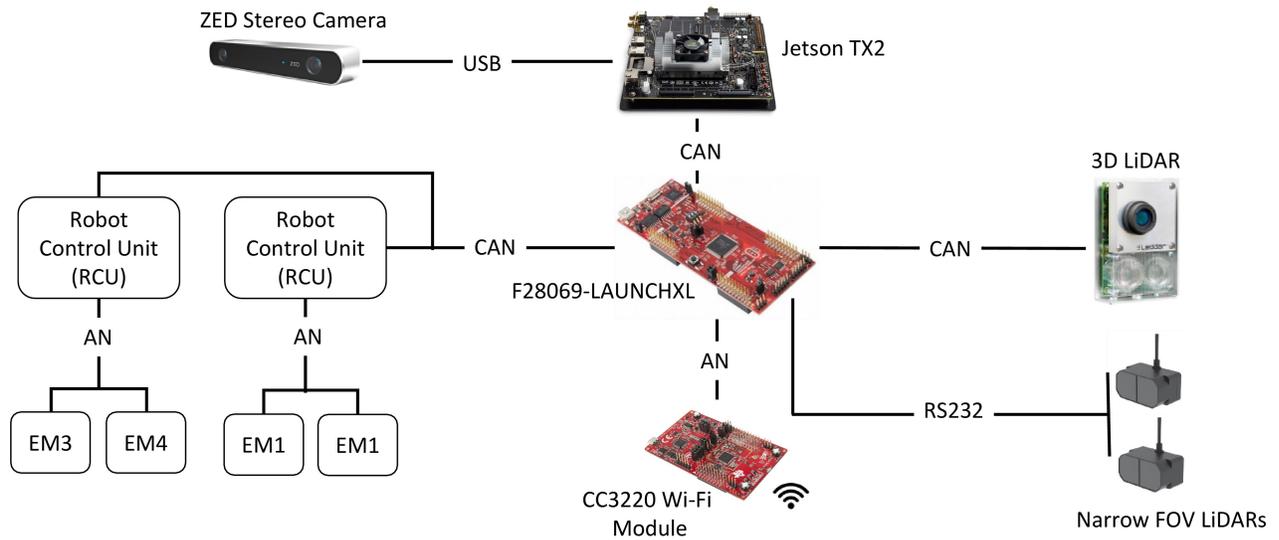
The robot has to assist and help travellers to find their way inside the airport to specific locations as gates, restaurants and restrooms. The presence of robotic platforms shall improve operational efficiency of airport personnel management and boost customer satisfaction. A strong collaboration between Innotech and San Diego Airport has been carried on in order to get feedback and suggestions on the steps to be performed during the development of the robot. The system shall be able to navigate autonomously inside the airport thanks to Simultaneous Localization and Mapping (SLAM) technology. It shall be able to detect objects along its path and to stop whenever an obstacle or a person is right in front of the robot itself. The interface shall be clear and user-friendly and accessible to people speaking different languages. Moreover, the robot shall travel at an average walking speed and have an innovative yet familiar-looking exterior design that airport passengers can empathize with. Hence the project has been divided into three macro areas regarding:

- SLAM implementation;
- System-user interface and speech recognition;
- Mechanical design.

The project started at the beginning of March 2019 and the team had to meet several deadlines with the Airport Committee. At the end of June 2019, a final presentation of the work done has been submitted, including: SLAM implementation on a prototype robot, object detection using ZED camera, the prototype of a user-interface app and a detailed CAD model of the base platform. This thesis strictly focuses on the mechanical design of the robot, briefly describing the other features of the project and the integration between the various systems.

2.2 System Architecture

The high-level system architecture is reported in figure 9.



InnTech

Figure 9 – High level system architecture of the system

The main components comprised in the architecture are:

- ZED Stereo Camera;
- Jetson TX2;
- Evaluation Board;
- Wi-Fi Module;
- 3D Lidar ;
- Narrow FOV Lidars;
- Motor controllers;
- Electric motors.

The ZED Stereo Camera is able to record a 3D map of the environment and is connected through USB to the Jetson TX2, a powerful embedded AI computing device in which the SLAM and the object detection algorithms run. The Jetson TX2 communicates through CAN bus with the F28069-Launch XL evaluation boards, as well as the 3D Lidar and the Narrow Field Of View Lidars. The latter provide information on the distance between the robot and the objects in its surroundings. The evaluation board communicates through AN with a CC3220 Wi-Fi module that connects the robot with the airport technical support, taking all the external info needed for service. The F28069-Launch XL take as input all the information coming from the sensors and provides commands to the motor controllers, that finally drive the motors.

2.3 SLAM

In robotics, Simultaneous Localization and Mapping (also known as SLAM) is the process of creating a three-dimensional map of an unknown environment and at the same time keeping track of the robot position inside it, (8). It is the main focus of the autonomous navigation challenge as it enables mobile robotic systems to move inside unknown surroundings and to simultaneously locate themselves, thus avoiding the need of a structured map. SLAM technology involves the use of depth-based sensors able to record and store 3D images of the environment. These images are then used to compute the robot position either with respect to a global reference frame or to a relative reference frame based on the previously stored image. The most commonly adopted 3D sensor for SLAM is the stereo camera shown in figure 10.



Figure 10 – ZED Stereo Camera

It is a system embedding two classical cameras located at a fixed horizontal distance. This specific configuration allows the stereo camera to record the same image from two slightly different perspective; secondly, a triangulation algorithm is used to compute depth thanks to the intrinsic parameters of the cameras, like their focal length or their distortion.

SLAM process can be implemented in two ways: the robot can either navigate through an unknown environment by creating the map and at the same time locating itself inside it, assisted by additional sensors like LIDAR or Inertial Measurement Units (IMU). Alternatively, the robot can firstly generate a map of its surroundings while being remote controlled in manual mode; next, it can finally navigate the same environment, recording its surroundings and comparing them to the previously loaded map. In this way, it is possible to find its position without the need of additional sensors.

2.4 System interface

The aim of system interface is to create a simple and innovative system allowing customers to easily exploit all the robot's functionalities. This is why it has been chosen to design an application via the Integrated Development Environment Android Studio. In this way, it has been possible to fully design the app and customize it in order to be intuitive and accessible. The main aim is to reduce the number of steps to be performed by the user in order to interact with the robot; moreover, five languages (English, Spanish, Japanese, French and Italian) have been implemented in order to make the system accessible to a great number of customers. Speech recognition has been embedded in the application using the Hidden Markov Models, the most commonly used algorithm for this task.

From a system architecture point of view, the app has to be able to communicate with the other components of the robot, especially with the Jetson TX2 in which the SLAM algorithm runs. The MQTT protocol has been selected for this task, as it allows an efficient wireless communication by using just a modem device.

Moreover, it is prominent to communicate with the airport technical support in order to receive information about flight schedule or gate location. This can be done by using the SSH remote connection, allowing a safe connection to the airport server.

2.5 Mechanical design

The mechanical design has been carried on starting from the analysis of the case studies, that helped to grasp the fundamental features of the design of a mobile robot. The design process has involved three main areas that will be described in details in the following chapters: the choice of the wheel configuration, the drive train design and the CAD modelling of the structure. The first one has been done by comparing various frequently adopted wheel configurations for robotics from the point of view of the simplicity of control, the degree of steerability, the number of driving motors and the simplicity of mounting. The second one has been performed by making reasonable assumptions on the main features of the robot like weight, cruise speed and acceleration, comparing the most used electrical motor in mobile robotics applications. Finally, several CAD models have been developed basing on the needs expressed by the airport committee and by surveys taken to travellers in the actual airport. This continuous interaction lead to the development of five different ideas that are shown below.

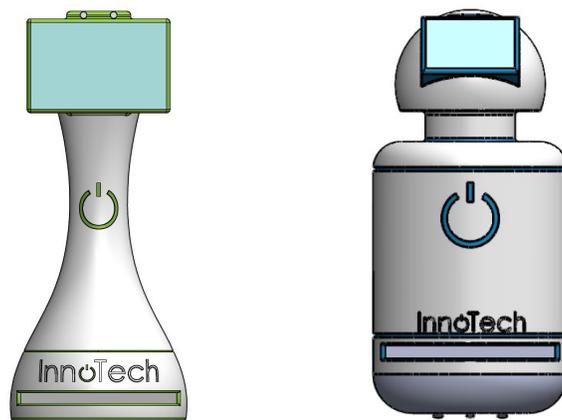


Figure 11 – Assistant robot concept 1 and 2

The Airport Committee specifically asked for a design that had no human-resembling features (as pointed out in section 1.3), that could show a bond between the robot and the airport and that could inspire travellers curiosity and willingness to interact with it. These directives led to avoid human features and to adopt rounded shapes for all the five concepts. The colors of San Diego Airport’s logo have been used for the fillets and decals, to create a connection between the robot and its work place. These sketches have been submitted to San

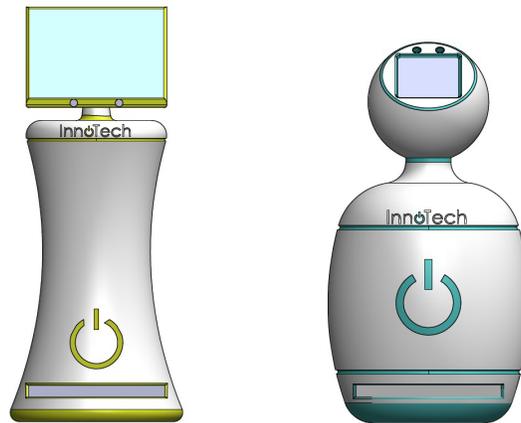


Figure 12 – Assistant robot concept 3 and 4



Figure 13 – Assistant robot concept 5

Diego Airport Committee that selected one out of them, after surveys and evaluations by customers. The design number 2 has been preferred among the others, and thus developed in details in chapter 5.

3 Wheel configuration

3.1 Robotic Mobility

The first step of the mechanical design has been analysing the most common wheel configurations used in robotic platforms and comparing them basing on several factors and considering the specific workplace the robot is going to operate in. Before delving in to the subject, the fundamental concept of holonomy has to be explained. A system is said to be holonomic if the

number of controllable degrees of freedom is equal to the total number of degrees of freedom, (9). A robot has to move in a plane (the ground), so it has 3 degrees of freedom: given a reference system, x and y define the position of the robot while θ define its orientation. If the robot can independently change its position along x , y and its orientation θ , it is holonomic. Otherwise, it is said to be non-holonomic, and various maneuvers have to be performed in order to reach a desired configuration. As an example, a car is a non-holonomic system, as it can only go straight and steer its front wheels: it has 2 controllable degrees of freedom. Hence, sideways motion is not allowed and parallel parking has to be performed following a certain number of maneuvers. From an analytical point of view, a system is holonomic if it only has holonomic constraints, i.e., they can be expressed in the following form:

$$f(x_1, x_2, \dots, x_i, \dots, x_N, t) = 0; \quad (1)$$

Where x_i are the system coordinates. It means that the system only has positional constraints at most depending on time. A system having constraints that can not be expressed in the previous form or that depend on the speed of the generalized coordinates is for sure non-holonomic. In robotics, non-holonomic robots are very common because of the simple design and the ease of control. In the simplest configurations, each controllable degree of freedom is independently driven by a specific actuator, further simplifying the control algorithm. Path planning can result difficult especially in narrow environments, as additional space is required to perform maneuvers. Holonomic robots are less common as the control is usually more sophisticated, but they offer full mobility and simple path planning.

Another important concept regarding wheeled robots is the so called instantaneous center of curvature (ICC): it is defined as the point of intersection of the each wheels' axis, (9). If such point exists, then each wheel will rotate without slippage about this point, and the linear velocity of each wheel during a rotation will just be given by:

$$v_i = \omega(t)d_i \quad (2)$$

where $\omega(t)$ is the angular velocity about the ICC, while d_i is the distance between the ICC and the wheel plane. This means that the relative velocity of each wheel with respect to the others is null, and is consistent with a rigid rotation. If such point does not exist,

i.e., the wheels' axis do not have an unique common point of intersection, then there will be slippage. If the robot is going straight, meaning that no rotation is occurring, all the wheels are parallel and their axis are considered to intersect at a point on the line at infinity. This means that the ICC exists and lies at infinity.

3.2 Wheel type analysis

Before analysing the various wheel configurations, it is convenient to study how a wheel is related to the robot frame and which kind of wheels are typically used in robotic platforms, (10). In particular, with respect to the robot frame, a wheel can be:

- **Active** if it is driven by a motor and contributes to traction. The number of active wheels determines how much torque is required to each wheel in order to move the robot;
- **Idler** if it is not driven by any motor and does not contribute to traction. An idler wheel is used to stabilize the robot and to distribute the weight among a greater number of wheels;
- **Fixed** if the axis of the wheel has a constant angular position with respect to the robot frame;
- **Pivot** if the axis of the wheel can change its angular position with respect to the robot frame;
- **Steering** if it actively modifies the orientation of the robot. Usually they are driven by a motor whose axis is perpendicular to the ground.

Finally, it is possible to distinguish between three different families of wheels: standard, omni-directional and spherical.

Standard wheels are the most common type of wheels. They are used in the majority of robotics platforms and usually do not allow holonomic motion. The most important parameters are the diameter and the width, as well as the load rating. They can be both active, if driven by a motor, or idler. As shown in the figure below, they can be fixed, steered

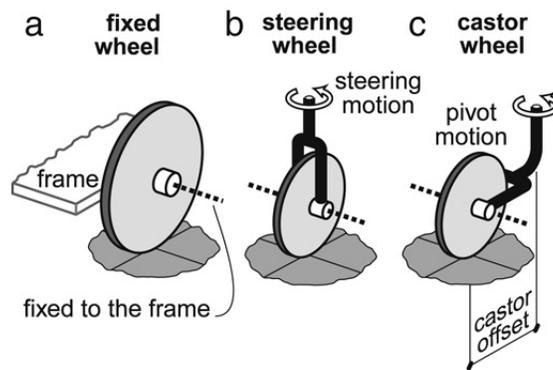


Figure 14 – Fixed, steered and pivot standard wheel.

or pivot. Motion along the axis parallel to the wheel's plane is called roll. Motion along any other axis is called skid.

When a standard wheel is equipped with a bracket (used to attach it to a platform), the whole assembly is called caster, (17). There are two families of casters, as can be seen in figure 15: swivel casters allow 360° motion when under load, thus they are actually pivot wheels. Rigid casters only allow forward/backward motion, hence they are fixed.



Figure 15 – Swivel caster (left) and rigid caster (right), (17)

Omni-directional wheels are equipped with small rollers along the circumferences of the wheel itself. This kind of wheel can be both active or idler; in the first case, they are driven exactly as a standard wheel, while the rollers are passive in any case. Moreover, they are always fixed, meaning that their axis has a fixed angular position with respect to the robot frame. What makes this kind of wheels unique is the fact that, thanks to the presence of the rollers, the friction along whichever direction is minimum, allowing sideways motion and steering maneuvers almost without any skid effects. This is why they are called omni wheels:

moving along any direction is almost as smooth as forward and backward motion also if the axis has a fixed angular position with respect to the robot frame. Depending on the number of rollers and on the angle between the rollers plane and the wheel plane, it is possible to distinguish three main kinds of omni wheels:

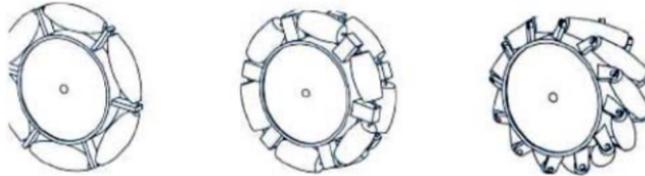


Figure 16 – Universal, double universal, mecanum omni wheel.

The universal wheels have a single line of rollers, while the double universal have two lines of rollers. The mecanum wheel have rollers whose plane is rotated of 45 degrees with respect to the wheel plane. This kind of wheels are often used in holonomic platforms.

Spherical wheels allow omnidirectional motion and thus can be used in holonomic platforms, as friction along any direction of motion is always the same. They are usually idler and pivot, and are used to stabilize the robot and hold its weight. The ball has theoretically a single contact point with the ground, and it is usually supported by smaller ball bearings. The use of spherical wheels has an advantage with respect to other kind of pivot wheels: the latter has to adjust its axis before a directional change can be achieved. This problem is completely overcome by the use of spherical wheels. A particular configuration involving this kind of wheels is the ball balancing robot, a platform that balances itself on a single spherical wheel. However, due to the dynamical instability and to the extremely hard control techniques, this configuration will not be mentioned in the following subsection.

Now it is possible to introduce a fundamental concept regarding mobility of wheeled robots: the degree of maneuverability. It is defined as:

$$\delta_M = \delta_m + \delta_s \quad (3)$$

where δ_m is called degree of mobility, while δ_s is the degree of steerability. In particular:

- $\delta_m = 1$ when the robot can only change its position by altering the speed of its standard wheel, i.e., only forward/backward motion is achievable.

- $\delta_m = 2$ when the robot can modify both its position and orientation by changing the velocity of its standard wheels.
- $\delta_s = 0$ if no steered wheel is present, meaning that there is not an explicit steering mechanism.
- $\delta_s = 1$ if one independent steering wheel is present.
- $\delta_s = 3$ only if no standard wheels are used in the platform.

Two robots having the same δ_M do not necessarily share the same drive train, as the sum could be given by different values of δ_m and δ_s . A robot having $\delta_M = 2$ has a ICC constrained to lie on a line. A robot having $\delta_M = 3$ has a ICC that can freely move on the plane.

3.3 Wheel configuration analysis

Once the main categories of wheels have been discussed, it is possible to describe the various configuration mostly used in robotic platforms. The main criteria chosen to analyse and compare these configurations are: holonomy, capability of zero radius turn, simplicity of control, number of motors and degree of steerability. It is important to state that the perfect configuration does not exist, as each of them has its own advantages and drawbacks. The designer has to weight each of the aforementioned criteria and then decide which platform best fits the project budget, the experience of the team, the time available for the development of the project as well as the operational conditions of the robot itself.

Since this project is run by a newborn start up trying to sell its own product to a big investor like San Diego Airport, budget is clearly a very important issue that has to be strongly taken in to account. That is why it is preferable to have the lowest number of motors, as they are usually quite expensive component. The deadlines the team has been given by the Airport Committee were very tight, so that the available time to develop a control algorithm for driving the motors will play a prominent role in choosing the wheel configuration. A solution requiring an easy control will be thus preferred. Moreover, zero turn radius capability is a feature that extremely facilitates path planning, as no maneuvers are needed to change the orientation of the robot frame. Hence, turning in place will play a key role. Furthermore, it is important to bear in mind where the robot is going to operate:

since the airport is a large environment usually lacking of narrow paths or lean obstacles, holonomy shall be considered as important, yet not fundamental in the decision process.

The configurations that will be analysed are :

- Tricycle drive;
- Differential drive;
- Skid steering;
- 3 Omni wheel drive;
- 4 Omni wheel drive;
- Ackermann Steering drive;
- Synchronous drive.

Tricycle configuration is one of the simplest driving configuration, consisting of two rear wheels attached to a common axle driven by a single motor , (11). The front wheel is steered and powered by a second motor whose axis is perpendicular to the ground. Hence, the total number of motors needed for this configurations is 2: a driving motor and a steering one. The motor powering the rear wheels has to provide by itself all the driving torque needed to move the robot. The control strategy is very simple, as forward/backward motion only depends on the speed of the driving motor, while the steering angle and speed only depend on the steering motor. This configuration is non holonomic and does not allow zero radius turn, thus making more difficult the implementation of path planning algorithms.

The control variables are the steering direction $\alpha(t)$ with respect to the robot frame, and the angular velocity of the output shaft of the motor $\omega(t)$. The linear velocity of the robot depends on the radius of the driving wheels, as $v(t) = \omega(t)r$. When the robot steers, it follows a radius of curvature defined as:

$$R(t) = d \operatorname{tg}\left(\frac{\pi}{2} - \alpha(t)\right) \quad (4)$$

where d is the distance between the steering wheel axle and the drive wheels axle. This equation is useful to define the angular velocity of the robot frame around the instantaneous center of rotation:

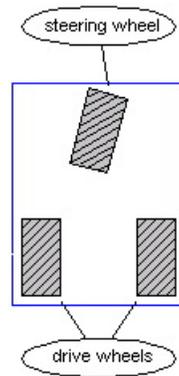


Figure 17 – Tricycle drive configuration.

$$\omega_r(t) = \frac{v(t)}{\sqrt{d^2 + R^2(t)}} \quad (5)$$

Differential drive is one of the most used platforms in robotics, (19). It consists of two driven wheels powered by two different motors, plus one (or more) pivot idler wheels used to stabilize the robot and hold its weight. The name differential comes from the fact that the behaviour of the robot depends on the difference between the speeds of the driving motors. Control strategy is considered simple, not as much as the tricycle configuration though, since forward/backward motion and steering must be handled jointly.

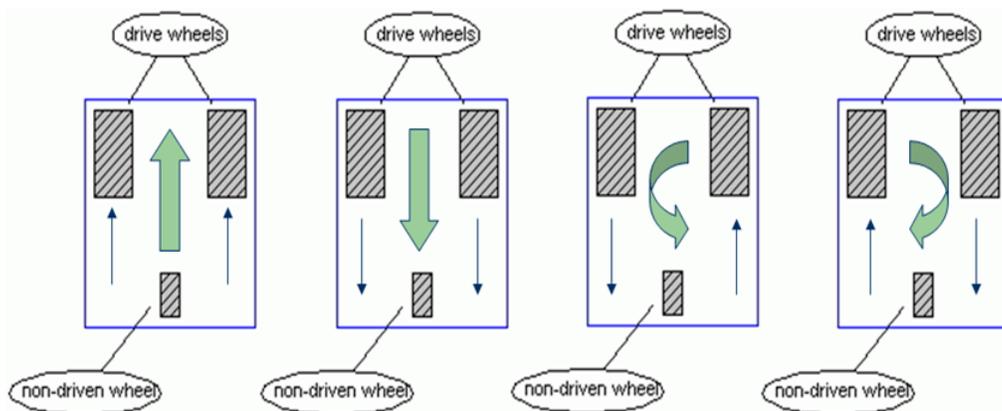


Figure 18 – Differential drive configuration.

If the robot has to move straight, both the wheels have to spin at the same rate in the same direction. If the robot has to turn, one wheel will have to turn faster (or slower) than

the other, eventually resulting in a steering action. In particular, if the robot has to turn right, the left will have to be faster than the right one. This configuration is non holonomic, but is able to provide zero turning radius: if the wheels spin at the same rate but in opposite directions, the robot will eventually turn in place. The total number of motors needed is thus 2, and each of them provides half of the total torque needed to move the robot. Hence, the control variables are 2: the angular velocity of the right wheel $\omega_r(t)$, and the angular velocity of the left wheel $\omega_l(t)$. The linear velocity of the two wheels is straightforwardly obtained by multiplying the angular velocity times the radius, thus obtaining $v_r(t)$ and $v_l(t)$. The speed of the robot frame is given by:

$$v(t) = \frac{v_r(t) + v_l(t)}{2} \quad (6)$$

while the angular velocity about the ICC is given by:

$$\omega(t) = \frac{v_r(t) - v_l(t)}{L} \quad (7)$$

We can easily notice how, when $v_r(t)$ and $v_l(t)$ are equal in value and sign, no rotation will occur. When they are equal in value but opposite in sign, $v(t)$ is null and zero radius turn will occur. It is also clear how the position of the idler wheels does not affect the behaviour of the platform, only influenced by the distance L between the two driving wheels. A frequent adjustment of the speed on the two motors is needed when going straight. In fact, a high frequency measure of their speed is performed, as it is very likely that their actual speed is not perfectly equal (due to manufacturing tolerances, different ground conditions, ect..), resulting in an unwanted steering action. This configuration is very common since it eliminates the need of an explicit steering mechanism.

Skid steering is very similar to differential drive, as steering is achieved by changing the relative speed of the driving motors, avoiding the presence of a steering mechanism, (12). In skid steering though, each motor drives multiple wheels or tracks. The main advantage of this configuration is the increased traction provided by multiple wheels or tracks, useful in rough terrains. Moreover, the robot is stable and does not need any additional wheel.

Nevertheless, steering requires a great amount of torque, as tracks or wheels (especially the front and the rear ones) have to skid while the robot turns. In fact, the static friction force between them and the ground has to be overcome, resulting in a very high torque and in a

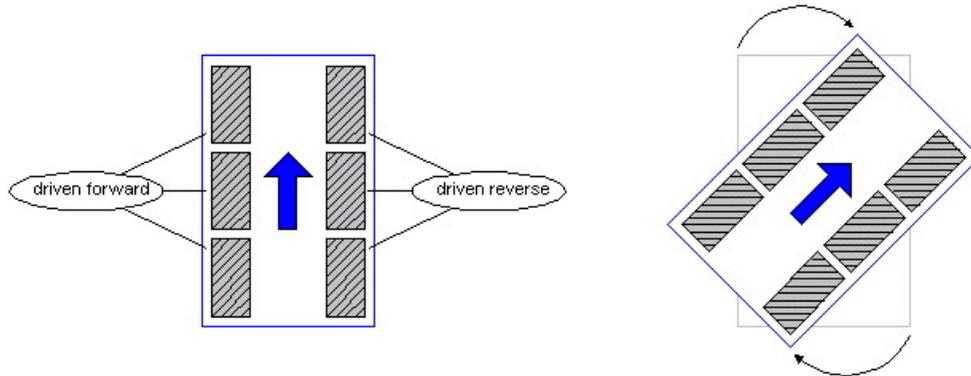


Figure 19 – Skid steering configuration.

reduced life cycle of the wheels (or tracks), caused by the high wear. Skidding also causes problems with odometry, so that sensors can not track exactly the position of the robot. Going perfectly straight is again a difficult task, likewise the differential drive.

3 Omni wheel drive consists in a platform usually driven by 3 omni (or mecanum) wheels spaced at 120 deg, (13). It is an holonomic configuration, thus enabling zero radius turn as well as sideways motion without any explicit steering mechanism. Given the geometry of this configuration, the ICC always exist and its position is fixed at the exact center of the platform. The robot reaches its desired position, speed and orientation by changing the relative value, direction and verse of the speeds applied to each wheel. In particular, given a desired linear speed $v_\alpha(t)$ pointing toward a direction α with respect to the robot frame, and a desired angular velocity $\omega(t)$, the velocity of each wheel is obtained by:

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = v_\alpha(t) \begin{bmatrix} \cos(30 - \alpha) \\ \cos(150 - \alpha) \\ \cos(270 - \alpha) \end{bmatrix} + \omega(t)R$$

where R is the distance between the ICC and the wheel planes. The projections of $v_\alpha(t)$

along the tangential direction of each wheel gives the amount of velocity needed from every motor to reach the desired translation speed.

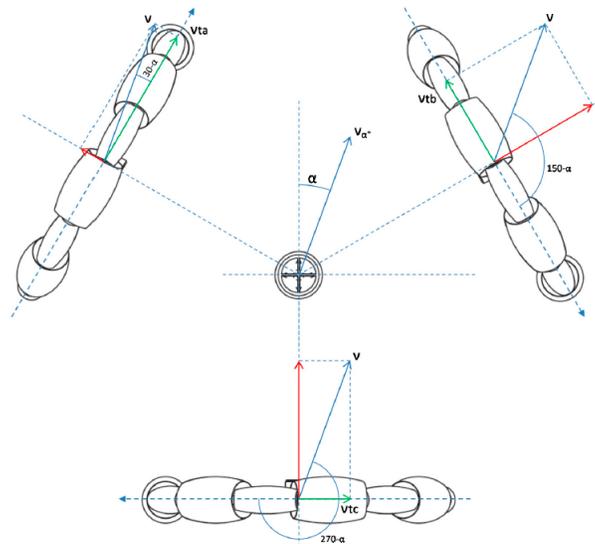


Figure 20 – 3 Omni wheel drive configuration for translation.

The amount of speed needed from every motor to reach the desired angular speed is given by the product vector between $\vec{\omega}(t)$ and distance \vec{R}_i .

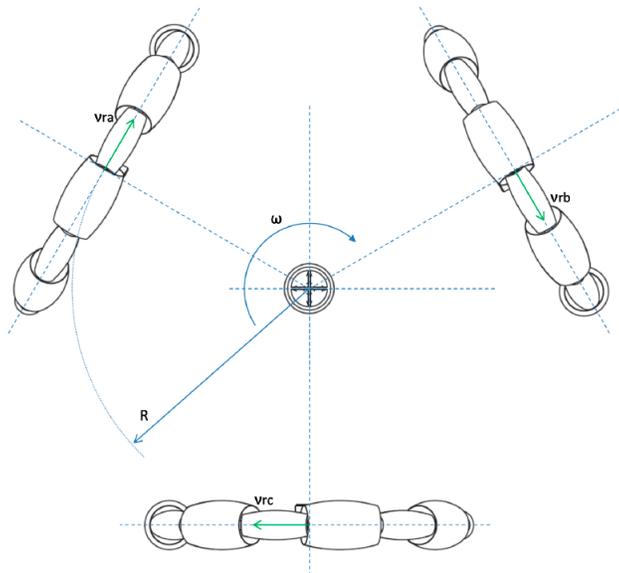


Figure 21 – 3 Omni wheel drive configuration for rotation.

Hence, each wheel must be driven by a motor and control is very difficult because 3 different computations have to be performed for the 3 wheels, as they are not aligned to the same axis. This is also the reason why the efficiency of the system is usually low: when going straight, only one motor will run at its rated efficiency, while the others will have to turn more slowly.

4 Omni wheel drive is a platform usually driven by 4 omni (or mecanum) wheels spaced at 90 degrees, so that two wheels are parallel to each other, and the other two are perpendicular to them, (14). This extremely eases the control strategy, as only two computations are needed (one for each pair of parallel wheels). Moreover, efficiency is increased with respect to the previous configuration: when going straight, two wheels will work at its rated efficiency, while the other two are completely idler (not wasting any power). The main disadvantage of this configuration is the cost of having a fourth wheel and motor. Of course, also the 4 omni wheel drive is holonomic, providing turning in place and sideways motion.

Synchronous drive uses 3 standard wheels arranged at the vertices of an equilateral triangle, (15). The wheels' axis are all parallel to each other and are driven by a single motor through a series of belt and chains, so that they always spin at the same rate in the same direction. A further motor is used to turn the wheels and thus steering the robot.

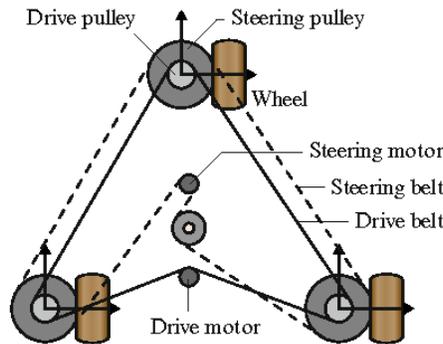


Figure 22 – Synchronous drive configuration.

The number of total motor needed is 2, and control is quite simple as forward/backward motion and steering are handled separately by the two motors. Given the angular speed of the steering motor $\omega(t)$, the angular speed of the driving motor $\omega_d(t)$ and the radius of the wheel, it is possible to compute the linear velocity of the wheels as $v(t) = \omega_d(t)r$ and the

evolution of x , y and θ with the following equations:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = v(t) \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega(t)$$

It is clear how linear and angular behaviour are decoupled from a control point of view. Straight motion is guaranteed mechanically, and no need for an adjustment is required (as in differential drive). Nevertheless, the axis of the wheels have to be perfectly parallel during construction and mounting processes, otherwise resulting in a not straight line motion that can not be adjusted by control algorithm. This platform allows holonomic motion, but the mechanical realization of this kind of configuration is very complex.

Ackermann steering is mostly used in car-like robots, (16). It comprises 4 wheels: the 2 in the rear are connected to the same axle and driven by a single motor, and are responsible for forward/backward motion. The two front wheel are connected to a second axle which is mechanically coupled to another motor, whose axis is perpendicular to the ground and is responsible for steering the platform.

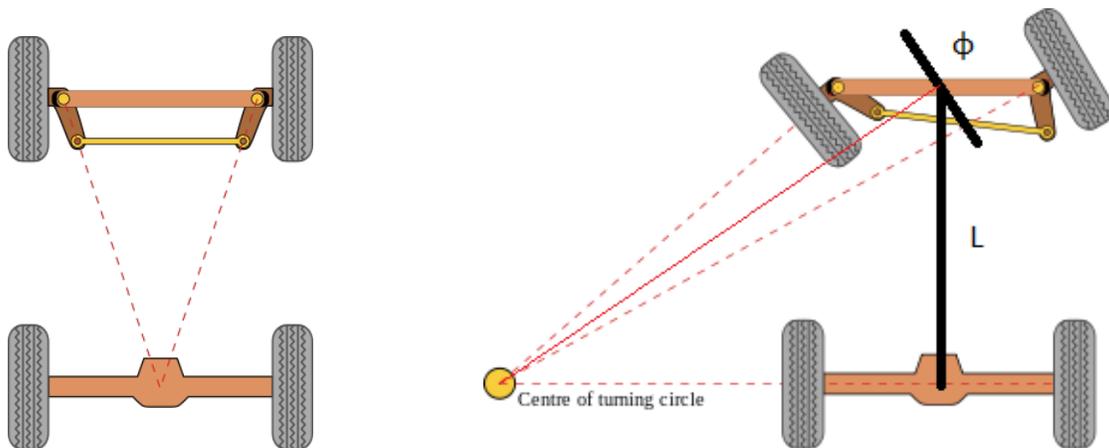


Figure 23 – Ackermann steering configuration.

During steering, the Ackermann design allows the inner tire to turn with a greater angle with respect to the outer tire, thus avoiding skidding that would appear if the front wheels

would remain parallel during steering. Control is simple, as separate motors handle translation and rotation, but this kind of configuration is non holonomic and does not allow zero radius turn. The evolution of the linear velocity along x and y as well as of the heading θ is given by:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = v(t) \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ \frac{\tan(\phi)}{L} \end{bmatrix}$$

Moreover, the mechanical design is quite complex and expensive to realize. It is mostly used for outdoor application where increased stability and control are needed.

It is possible to compare all the described configurations according to the criteria chosen for the analysis, in order to finally select the one that best fits the needs of the project.

	N of motors	N of control variables	0 radius turn	Holonomy	δ_M
Tricycle	2	2	NO	NO	2
Differential	2	2	YES	NO	2
Skid steering	2	2	YES	NO	2
3 Omni drive	3	3	YES	YES	3
4 Omni drive	4	2	YES	YES	3
Ackermann	2	2	NO	NO	2
Synchronous	2	2	YES	NO	2

Table 1 – Wheel configuration comparison

The best trade off between cost, control and maneuverability is given by differential, skid steering and synchronous drive. The differential drive has been preferred over the other the other two. In fact, it avoids skidding and thus the need for a high torque when turning in place; the higher number of wheels is not needed as the robot will not be that heavy. Moreover, the mechanical realization of the differential drive is much easier than the synchronous drive, thus allowing a faster and more reliable design and mounting.

In the next sections, the kinematics and dynamics behaviour of the differential drive will be analysed in detail.

3.4 Differential drive kinematics

Kinematics studies the motion of points and bodies regardless of the forces that caused their motion, (19). It studies how geometric parameters affect vehicle performance. In the robotic field, kinematics links the wheels speed to the robot speed. It also provides a mathematical formulation of the system's constraints, that in the case of a differential drive robot are:

- The presence of an unique contact point C between the wheels and the ground;
- The no-slip constraint, meaning that the contact point C has a zero velocity with respect to the ground.

In fact, if C moved with respect to the ground, i.e. had a non zero velocity, the wheel would slip and no pure rolling would occur.

When applied to robotics, it is possible to distinguish between forward and inverse kinematics.

Forward kinematics computes the pose of the robot from the values of the motors' angular velocities. In the case of a differential drive, it is possible to obtain the new pose of the robot given its previous pose and the angular velocity about the ICC, that in turn depends on the linear speeds of the two wheels. In particular, forward kinematics problem solves the following problem:

$$(v_r, v_l) \longrightarrow (x, y, \theta) \quad (8)$$

Since the angular velocities of the two motors ω_r and ω_l are proportionally linked to the linear speed of the wheels v_r and v_l by the radius of the wheels themselves, the forward kinematics problem actually links the pose of the robot to the motor's angular velocities.

The value of the linear velocity of the body V_B is given by the average of the linear velocities of the wheels:

$$V_B = \frac{v_r + v_l}{2} \quad (9)$$

from which it is possible to write that:

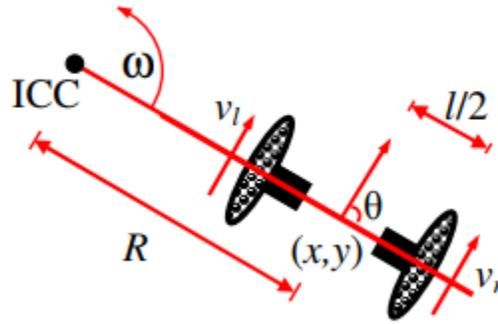


Figure 24 – Differential drive geometry.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = V_B \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} = r \frac{\omega_r + \omega_l}{2} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} \quad (10)$$

The previous equation shows how, for a given $\frac{\omega_r + \omega_l}{2}$, the radius of the wheels should be increased to get a higher linear velocity.

Regarding the rotational speed ω , calling l the distance between the two wheels' planes and R the distance of the ICC from the middle plane of the robot as in picture 24, it is possible to state that:

$$\omega \left(R + \frac{l}{2} \right) = v_r \quad (11)$$

$$\omega \left(R - \frac{l}{2} \right) = v_l \quad (12)$$

From the previous equations, it is straightforward to obtain:

$$\omega = \frac{v_r - v_l}{l} = \dot{\theta} \quad (13)$$

Equation 13 shows how, for a given value of $v_r - v_l$, the rotational speed of the robot is inversely proportional to the base length l . Moreover, the radius of curvature R given by:

$$R = \frac{l}{2} \frac{v_r + v_l}{v_r - v_l} \quad (14)$$

Given a fixed reference frame xOy , it is possible to compute the position of the ICC with respect to the origin as :

$$ICC = \begin{bmatrix} ICC_x \\ ICC_y \end{bmatrix} = \begin{bmatrix} x - R \sin \theta \\ x + R \cos \theta \end{bmatrix} \quad (15)$$

Since the robot is rotating about its ICC with an angular velocity ω , in an interval of time δt the robot has rotated of an angle equal to $\omega \delta t$. Its new heading is thus:

$$\theta' = \theta + \omega \delta t \quad (16)$$

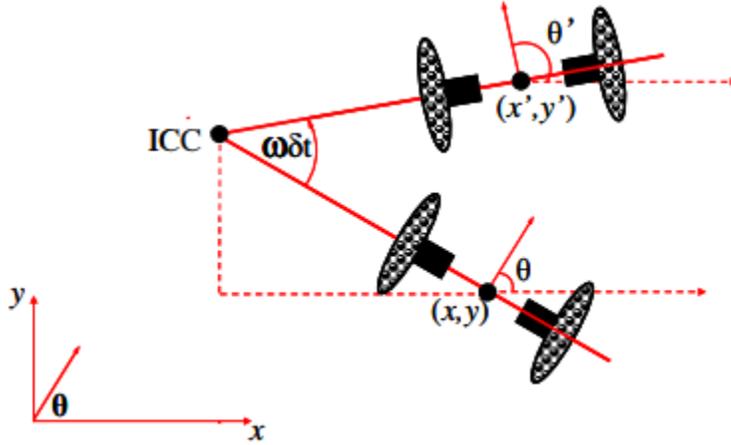


Figure 25 – Rotation about the ICC of $\omega \delta t$.

Given a starting position (x, y) , a 2D rotational matrix can be used to compute the new position (x', y') at time $t + \delta t$:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \underbrace{\begin{bmatrix} \cos \omega \delta t & -\sin \omega \delta t \\ \sin \omega \delta t & \cos \omega \delta t \end{bmatrix}}_2 \underbrace{\begin{bmatrix} x - ICC_x \\ y - ICC_y \end{bmatrix}}_1 + \underbrace{\begin{bmatrix} ICC_x \\ ICC_y \end{bmatrix}}_3 \quad (17)$$

This is equivalent to 1) translate the ICC to the origin of xOy , 2) perform a rotation of $\omega \delta t$ around it, and then 3) translating back the ICC. So, given ω , R and δt , it is possible to obtain the new pose from the previous one. Recall from equation 13 that ω depends on v_r and v_l , that are the linear velocities of the wheels.

Inverse kinematics computes the motor's parameters required to get to a specific pose of the robot given the current one, (18). In particular, the inverse kinematics equations solve the following problem:

$$(x, y, \theta) \longrightarrow (v_r, v_l) \quad (18)$$

This problem is under-constrained and has infinite solutions. Some constraints have to be applied in order to be solved.

Given a desired heading and velocity, the evolution in time of the pose of the robot can be described by:

$$x(t) = \int_0^t v(t) \cos(\theta(t)) dt \quad (19)$$

$$y(t) = \int_0^t v(t) \sin(\theta(t)) dt \quad (20)$$

$$\theta(t) = \int_0^t \omega(t) dt \quad (21)$$

Since we are dealing with a differential drive:

$$x(t) = \frac{1}{2} \int_0^t (v_r(t) + v_l(t)) \cos(\theta(t)) dt \quad (22)$$

$$y(t) = \frac{1}{2} \int_0^t (v_r(t) + v_l(t)) \sin(\theta(t)) dt \quad (23)$$

$$\theta(t) = \frac{1}{l} \int_0^t (v_r(t) - v_l(t)) dt \quad (24)$$

The presence of a non holonomic constraint (like in the differential drive) does not allow to reach any desired pose just by setting appropriate values of v_r and v_l . In order to plan a trajectory, the zero radius turning ability plays a key role. In fact, the most commonly adopted strategy is: rotate the robot until its heading points toward the target position; drive straight until the latter is reached; rotate again the robot so that the target orientation is reached. In this simplified cases, the motion equations become:

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x + v \cos \theta \delta t \\ x + v \sin \theta \delta t \\ \theta \end{bmatrix} \quad (25)$$

if the robot is going straight, i.e., if $v_r = v_l = v$.

If the robot is turning in place, then $v_r = -v_l = v$, so that:

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta + \frac{2v\delta t}{l} \end{bmatrix} \quad (26)$$

The two previous systems can be inverted to find v and, consequently, v_r and v_l , thus solving the inverse kinematics problem stated in equation 18.

3.5 Differential drive dynamics

Dynamics is able to link the performance of the robot to its geometry and to its inertia variables (masses and moments of inertia), (19). This means that it studies the forces and torques that cause the motion studied by kinematics. In particular, it links the torques applied to the wheels to robot's acceleration. There are several ways to approach the dynamic model of a system; here, the Lagrangian approach will be used. The Lagrangian function of a system is defined as:

$$\mathcal{L}(q, \dot{q}) = K.E. - P.E. \quad (27)$$

i.e. the difference between the kinetic energy and the potential energy of the system. The variables q are called configuration variables, as they are able to completely describe the system at a given time. For a differential drive configuration, they are:

$$\vec{q} = [x \quad y \quad \theta \quad \phi_1 \quad \phi_2]' \quad (28)$$

i.e. the pose of the robot plus the angular position of the two driving motors' shafts. The figures below show the generalized coordinates of the system.

The Lagrangian approach consists in writing the following equation for each of the configuration variables:

$$\frac{d}{dt} \left[\frac{\partial \mathcal{L}(\dot{q}_i, q_i)}{\partial \dot{q}_i} \right] - \frac{\partial \mathcal{L}(\dot{q}_i, q_i)}{\partial q_i} = \mathcal{F}_i \quad (29)$$

The term \mathcal{F}_i is the vector of generalized forces applied to the system, i.e. the vector of forces and torques acting on each generalized coordinates. In the case of a differential drive,

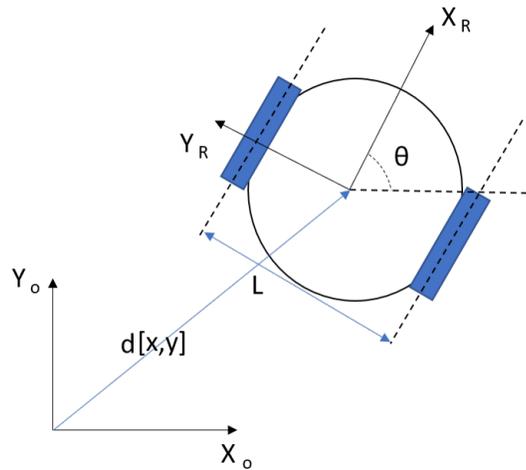


Figure 26 – Inertial reference system and robot reference system, top view.

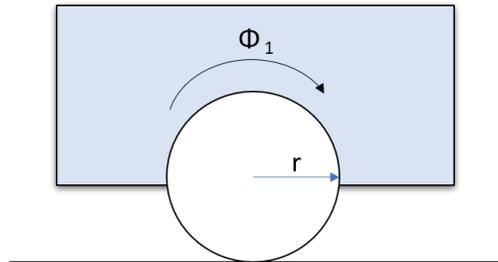


Figure 27 – Left view.

the only generalized forces acting on the system are the driving torques applied by the motors to the wheels, thus:

$$\mathcal{F}_i = \begin{bmatrix} 0 \\ 0 \\ 0 \\ T_1 \\ T_2 \end{bmatrix} \quad (30)$$

Solving this equation will eventually lead to link the robot's acceleration to the motor torques and to the other design parameters. Since there are constraints acting on the system, the method of undetermined Lagrange Multipliers has to be adopted in order to deal with those constraints. In particular, they can be arranged in the following way:

$$C(\vec{q})\dot{\vec{q}} = 0 \quad (31)$$

The constraints acting on a differential drive system ensures that the wheels' contact points with the ground have zero velocity. This result in 4 equations, one of which is redundant. Hence, there are 3 constraints acting on the system and 5 generalized coordinates, so the matrix C is a 3 x 5. It has to be inserted in the 29 in the following way:

$$\frac{d}{dt} \left[\frac{\partial \mathcal{L}(\dot{q}_i, q_i)}{\partial \dot{q}_i} \right] - \frac{\partial \mathcal{L}(\dot{q}_i, q_i)}{\partial q_i} - C(q_i)\lambda_i = \mathcal{F}_i \quad (32)$$

where λ_i is called the undetermined Lagrangian multipliers. In this way, it is possible to express the forces that are able to maintain the constraints. As already mentioned, they are the reaction forces at the wheel-ground contact that ensures the no-slip constraint.

Since the robot is going to move in a flat surface, the dynamic analysis will neglect the terms related to potential energy, so that:

$$\mathcal{L} = \sum_i K.E._i \quad (33)$$

i.e. the Lagrangian function will just be given by the sum of the kinetic energy of each generalized coordinate. Moreover, although the robot comprises many components (as batteries, electronic boards, sensors, ...), this analysis only takes care of the 3 components fundamental for the dynamics of the robot itself: the main body and the two wheels. The kinetic energy of each body is the sum of the translational and the rotational kinetic energies:

$$K.E._i = \frac{1}{2}m_i\|v_i^2\| + \frac{1}{2}\Omega_i^T I_i \Omega_i \quad (34)$$

where:

- m_i is the mass of the i-th component;
- v_i is the linear velocity of the i-th component;
- I_i is the moment of inertia of the i-th component;
- Ω_i is the angular speed of the i-th component.

Further simplifying assumptions can be made in order to ease the analysis: the center of mass of the main body lies along the axis of the wheels; the center of mass of each wheel lies exactly at the hub of the wheel itself; the body reference system is parallel to the inertial reference frame, i.e. the heading $\theta = 0$ and the two origins coincide, i.e. $d = 0$.

Under these assumptions, the linear and rotational velocity of the main body are:

$$\vec{V}_B = \begin{bmatrix} \dot{x} \\ \dot{y} \\ 0 \end{bmatrix} \quad \vec{\Omega}_B = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta} \end{bmatrix} \quad (35)$$

so that the kinetic energy of the main body is:

$$K.E._B = \frac{m_B}{2}(\dot{x}^2 + \dot{y}^2) + \frac{I_B}{2}\dot{\theta}^2 \quad (36)$$

The linear velocity of the wheels is given by the sum of the linear velocity of the body reference frame plus the contribution given by the angular rotation of the body itself. So:

$$\vec{V}_{wi} = \begin{bmatrix} \dot{x} + \frac{L}{2}\dot{\theta} \cos \theta \\ \dot{y} + \frac{L}{2}\dot{\theta} \sin \theta \\ 0 \end{bmatrix} = \begin{bmatrix} \dot{x} + \frac{L}{2}\dot{\theta} \\ \dot{y} \\ 0 \end{bmatrix} \quad (37)$$

while the angular velocity is given by:

$$\vec{\Omega}_{wi} = \dot{\phi}_i \begin{bmatrix} \sin \theta \\ \cos \theta \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 \\ \dot{\phi}_i \\ \dot{\theta} \end{bmatrix} \quad (38)$$

keeping in mind that $\theta = 0$ and that each wheel rotates both about its center of mass and about the origin of the body reference frame.

The moment of inertia of the wheels can be restored by the tensor of inertia of a cylinder, being r the radius of the wheel and t its thickness:

$$I_w^b = \begin{bmatrix} I_{xx}^b & 0 & 0 \\ 0 & I_{yy}^b & 0 \\ 0 & 0 & I_{zz}^b \end{bmatrix} = \begin{bmatrix} \frac{m_w}{12}(3r^2 + t^2) & 0 & 0 \\ 0 & \frac{m_w r^2}{2} & 0 \\ 0 & 0 & \frac{m_w}{12}(3r^2 + t^2) \end{bmatrix} \quad (39)$$

The apex b states that the tensor is written in the body reference frame. Remind that, in the case of a thin wheel, $t \ll r$, so that:

$$I_{xx}^b = I_{zz}^b \simeq \frac{m_w r^2}{4} \quad (40)$$

The previous tensor of inertia has to be transformed according to a rotation of angle θ about the Z axis:

$$I_w = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} I_w^b \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (41)$$

In the simplifying assumption that $\theta = 0$, the previous equation leaves I_w^b unchanged.

Now it is possible to write the equation of the kinematic energy of the wheels:

$$K.E._{wi} = \frac{m_w}{2} \|\vec{V}_{wi}\|^2 + \frac{1}{2} \vec{\Omega}_{wi}^T I_w \vec{\Omega}_{wi} \quad (42)$$

leading to:

$$K.E._{wi} = \frac{m_w}{2} (\dot{x}^2 + \dot{y}^2 + (\frac{l}{2})^2 \dot{\theta}^2 - 2(\frac{l}{2}) \dot{\theta} (\dot{x} \cos \theta + \dot{y} \sin \theta)) + \frac{I_{zz}^b}{2} \dot{\theta}^2 + \frac{I_{yy}^b}{2} \dot{\phi}_i^2 \quad (43)$$

Now it is possible to sum the contributions of the kinetic energy of all the components of the system, that in this case is also equal to the Lagrangian function \mathcal{L} :

$$\mathcal{L} = K.E._{tot} = K.E._B + K.E._{w1} + K.E._{w2} \quad (44)$$

that is the sum of equations 36 and 42. This equation allows to introduce two prominent parameters:

- $m_T = m_B + 2m_w =$ the total mass of the robot
- $I_T = I_B + m_B d^2 + 2m_w (\frac{l}{2})^2 + 2I_{zz}^b =$ total rotational inertia of the robot

Substituting the total kinetic energy found above in the Lagrangian equation 29, it is possible to relate the second derivatives (i.e. the accelerations) of each generalized coordinates to the robot inertial and geometrical parameters. In particular, with the assumption that

$\dot{x} = \dot{y} = 0$ (the robot is still), the forward linear acceleration is given by the second derivative of the first generalized coordinate x , equal to:

$$\ddot{x} \simeq \left(\frac{1}{r}\right)\left(\frac{1}{m_w + m_T}\right)(T_1 + T_2) \quad (45)$$

In order to increase the forward linear acceleration, the mass of the wheels and of the total vehicle should be of course minimized. Moreover, the total torque applied by the driving motors should be maximized; this can be achieved by choosing a proper gear ratio that increases the torque but reduces proportionally the rotational speed. Lastly, the radius of the wheel r should be the smallest practical one in order to get a high value of \ddot{x} . This is in contrast with the results regarding the top end speed of the vehicle; hence, a reasonable trade-off has to be reached when choosing the wheel radius.

The rotational acceleration is given by the second derivative of the third generalized coordinate θ , given by:

$$\ddot{\theta} \simeq \left(\frac{l}{2r}\right)\frac{T_1 - T_2}{I_B + 2I_{zz}^b + 3m_w\left(\frac{l}{2}\right)^2} \quad (46)$$

This equation refers to a situation in which the robot is operating at near zero velocity. If the applied torques T_1 and T_2 are equal in values but opposite in signs, it is possible to obtain the rotational acceleration achieved during a zero radius turn.

4 Powertrain design

4.1 Overview on electrical motors

The first step of the power train design involves the analysis and the sizing of the driving motor. In the mobile robotic fields, three main categories of electric motor are usually adopted, as explained in (21):

- Brushed DC motor;
- AC induction motor;
- Brushless DC motor (also called BLDC).

Brushed DC motors are the most well known kind of electric motor. A permanent magnet (the stator) creates the magnetic flux and a winded armature (the rotor) is free to spin inside it. The current flows through a set of brushes from the power supply to the armature, and a mechanical commutator is able to provide a constant torque. The efficiency of a DC motor is quite high (from 60 to 75%) and they are suitable to be used with a gearbox. They are usually cheaper with respect to other options but they are very high maintenance and require frequent replacement, as the brushes' materials wear during service because of the continuous contacts with the commutator.

In AC motors, a rotating magnetic flux is generated through a 3 phase winding in the stator; the rotor armature reacts trying to go against the variation of the flux itself (according to the Lenz Law), thus producing a torque that is able to make the rotor spin. The use of a mechanical commutation is avoided, taking advantage of the alternating AC power supply. This makes the AC motor low maintenance but also more expensive with respect to DC motors. The efficiency is lower (around 40 and 65%), as part of the input power has to be used in order to create the magnetic flux. Power density is also lower, meaning that usually AC motor are bigger than the other options. This could be an important issue in space-critical applications.

Brushless DC motor are similar to AC motor, as they work thanks to a rotating magnetic flux produced by the stator winding and they do not need mechanical commutation. The rotor though is made by permanent magnets. The commutation is performed electronically by a controller receiving feedback from the motor itself, measuring the rotor rotational position or the voltage of the stator's coils. They have the highest efficiency (from 65 to 80%), longest life service and highest power density, thus resulting in relatively small dimensions. The main drawback is cost, as BLDC motors are usually more expensive than all the other options. Their long life, low maintenance and high efficiency can actually be worth the initial investment.

This is the reason why BLDC motor has been chosen over brushed and AC.

4.2 BLDC mechanical characteristic

The mechanical characteristic of a motor is the curve relating the rotational speed to the provided torque, (20). For a brushless, the curve is a straight line with negative slope,

connecting two prominent points of the graph:

- The no load point, corresponding to a configuration in which the torque is null and the motor is free to spin at its maximum speed;
- The stall point, in which the motor provides the maximum torque available at null speed.

These two configurations are important because from the values of the stall torque T_s and the no load speed ω_o it is possible to draw the mechanical characteristic. Moreover, most of the data sheet provide these values in order to size the motor.

The mechanical characteristic can be found through:

$$\omega = \omega_o \left(1 - \frac{T}{T_s}\right) \quad (47)$$

Furthermore, it is possible to compute the mechanical output power as:

$$P = \omega T = \omega_o T - \frac{\omega_o}{T_s} T^2 \quad (48)$$

This is the equation of a parabola with downward concavity reaching its maximum at $T = \frac{T_s}{2}$.

The values of stall and no load current are often provided by motor manufacturers and allow to draw the current curve; since we know that in a DC motor the current is directly proportional to the provided torque, it is straightforward to obtain that:

$$i = i_o + (i_s - i_o) \frac{T}{T_s} \quad (49)$$

The current curve is a straight line with positive slope. Now it is possible to compute the efficiency of the motor, defined as:

$$\epsilon = \frac{P_{mech}}{P_{elec}} = \frac{\omega T}{V_o i} \quad (50)$$

the ratio between the mechanical output power and the provided electrical power, where V_o

is the nominal operating voltage. The maximum efficiency is given by:

$$\epsilon_{max} = (1 - \sqrt{\frac{i_o}{i_s}})^2 \quad (51)$$

occurring at:

$$\omega_{max} = \frac{1}{2}(\omega_o + \epsilon_{max}i_s \frac{V_o}{T_s}) \quad (52)$$

The wasted power:

$$P_w = P_{elec} - P_{mech} = (1 - \epsilon)V_o i \quad (53)$$

is proportional to the current i and thus to the torque T suggesting how the high torque range should be avoided for continuous operation.

The typical motor curves of a BLDC motor is shown in figure 28, (20).

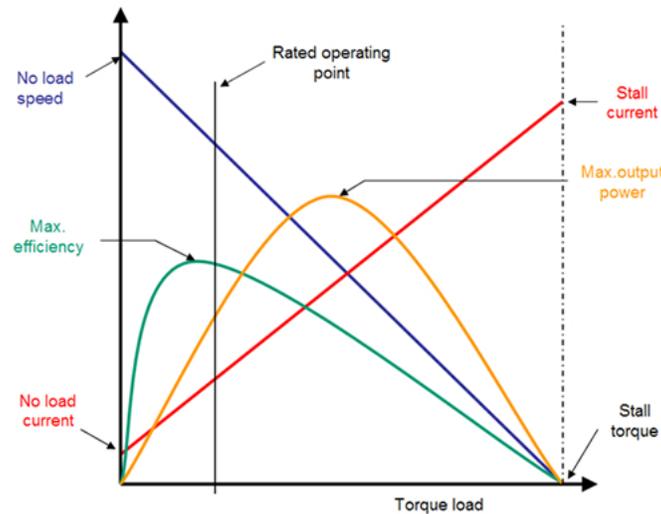


Figure 28 – BLDC motor curves

The rated operating point is a condition at which the motor can operate continuously without overheating. This is the most important point of the characteristic, as it is the best condition at which the motor should work. It is usually placed a little further the max efficiency point, and corresponds to a value of torque equal to 15 – 20% of the stall torque

T_s .

Regarding the value V_o , it depends on the power supply system. Typical battery package for mobile robotics have the following operating voltages: 12V, 24V, 36V, 48V. The supply voltage is directly proportional to the rotational speed ω of the motor, so that working, for example, at 12V instead of 24V will halve the value of the speed and thus of the power, keeping unchanged the values of torque and efficiency as shown in figure 29, (20).

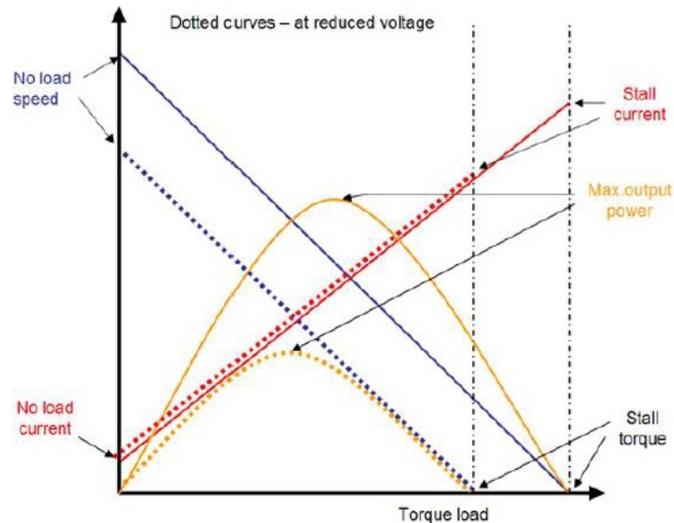


Figure 29 – BLDC motor curves at different voltages.

4.3 Motor sizing

In order to size the BLDC motor to insert in the design, the main specifications of the robot have to be taken into account. From these, it will be possible to come up with the required torque, speed and power, whose values will be considered when choosing the motor. The specifications are:

- **Weight** is assumed to be about 50 kg. This is an early assumption that will have to be verified once the CAD model will be finalized and the main components will be chosen, but is quite reasonable given the dimension of the robot and basing on the comparison with other robotic systems (like Savioke, etc...).
- **Cruise speed** is chosen to be similar to the average walking speed, that is around $1.5 \frac{m}{s}$, so that passengers will be able to follow the robot without any troubles.

- **Acceleration** is chosen so that the robot will be able to reach its cruise speed in 3 seconds, so that since $v = 1.5 \frac{m}{s}$, $\ddot{x} = a = 0.5 \frac{m}{s^2}$.
- **Wheel diameter** is chosen to be 150 mm (6 inches), as a reasonable trade off. Bigger wheels are able to go over obstacles more easily but require more torque to be driven. Smaller diameter implies proportionally smaller width, and could not be able to hold the weight of the robot, as said in (30). The mass of the wheel is assumed to be about 2 kg.
- **Rolling Friction** coefficient is chosen to be 0.015, that is the one between ordinary car type and concrete, (31). This is quite a conservative assumption, as the right value should actually be obtained performing tests in the airport pavement.

These information are sufficient to compute the required speed of the output shaft of the motors when travelling at cruise speed. In the data sheets it is usually expressed in RPM (revolutions per minute), so this unit of measurement will be used from now on when referring to the motor speed.

$$\omega_{RPM} = \frac{v}{2\pi r} 60s = \frac{1.5 \frac{m}{s}}{2\pi 0.075m} 60s \simeq 190RPM \quad (54)$$

Now it is possible to analyse the torques that the motors has to overcome in order to make the robot move. In this step, the total torques needed by the vehicle are analyzed. This means that subsequently, the torque values will have to be divided by 2, as there are two driving motors providing torque to the robot. Three phases can be identified: the acceleration phase, the constant speed phase, and the deceleration phase.

Acceleration phase	Constant speed phase	Deceleration phase
τ_1 =robot linear inertia	τ_3 =rolling friction resistance	τ_1 =robot linear inertia
τ_2 =wheel rotary inertia	τ_4 =air drag resistance	τ_2 =wheel rotary inertia
τ_3 =rolling friction resistance		
τ_4 =air drag resistance		

Table 2 – Torque required during acceleration, constant speed and deceleration phases.

In the deceleration phase, τ_3 and τ_4 are present and actually help the deceleration process, but in order to be conservative they will be considered equal to 0.

The robot linear inertia to be overcome is given by equation 45:

$$\tau_1 = (m_T + m_w)ar = 50kg \cdot 0.5 \frac{m}{s^2} \cdot 0.075m = 1.875Nm \quad (55)$$

The wheel rotary inertia is given by:

$$\tau_2 = I_w \alpha \quad (56)$$

where I_w is the moment of inertia of wheel with respect to its rotation axis, while α is the angular acceleration, equal to the ratio between the linear acceleration and the radius of the wheel. Assuming that the wheel weight is around 2kg,

$$I_w \simeq \frac{m_w r^2}{4} = \frac{2kg(0.075m)^2}{4} = 5.6 * 10^{-3}kg \cdot m^2 \quad (57)$$

and that:

$$\alpha = \frac{a}{0.075} = 6.7 \frac{rad}{s^2} \quad (58)$$

the value of τ_2 is about $0.02Nm$.

The rolling friction resistance is given by:

$$\tau_3 = C_{rr}Nr = C_{rr}mgr = 0.015 \cdot 50kg \cdot 9.81 \frac{m}{s^2} \cdot 0.075m = 0.55Nm \quad (59)$$

The air drag resistance (22) is given by :

$$\tau_4 = \frac{1}{2}c_d \rho v^2 Ar \quad (60)$$

where:

- $c_d = 0.7$ is the drag resistance coefficient of a hollow cylinder-shaped object in laminar flow;
- $\rho = 1.225 \frac{kg}{m^3}$ is the standard density of the air;
- $v = 1.5 \frac{m}{s}$ is the linear velocity of the robot;
- $A = 0.75m^2$ is the frontal area of the robot assuming a diameter of 0.5 m and a height of 1 m.

so that $\tau_4 = 0.055Nm$.

Finally, the torque required to perform a zero radius turn can be computed through 46. Assuming that the body of the robot is a hollow cylinder with a diameter of about $l = 0.5m$, it is possible to compute its rotational inertia as $I_B = m_B(\frac{l}{2})^2$. Moreover, it is possible to assume that the desired rotational acceleration about the Z axis allows the robot to reach in 2 seconds a speed of $\frac{1}{3}$ RPS (an entire revolution performed in 3 seconds). The corresponding value of rotational acceleration is:

$$\ddot{\theta} = \frac{\frac{1}{3}RPS}{2s} = \frac{2.1\frac{rad}{s}}{2s} = 1.05\frac{rad}{s^2} \quad (61)$$

This allows to compute the required torque through equation 46 as:

$$\tau_{rot} = \frac{2r\ddot{\theta}}{l}(I_B + 2I_{zz}^b + 3m_w(\frac{l}{2})^2) \quad (62)$$

This leads to a value of torque equal to:

$$\tau_{rot} \simeq 1Nm \quad (63)$$

As will be discussed more in detail in the following subsection, usually BLDC motors work at much higher speed providing a much lower torque than the ones computed before. This means that a gear reduction is necessary: a gear box is able to reduce the speed of the output shaft of the motor while increasing proportionally the torque provided, so that the output shaft of the gear box can be directly connected to the wheels. However, gearboxes usually have an efficiency of around 60% to 90%, meaning that only a certain amount of torque provided by the motor will actually be transferred to the wheels. This is why the torque values that came up from the previous calculations will have to be divided by the efficiency of the chosen gearbox in order to obtain the value of the requested torque at the motor.

4.4 Gearmotor choice

Now that the required torques are available, a motor with the following requirement has to be found: a starting torque at least greater than the sum of all the torques needed during the acceleration phase and a nominal operating point close to the constant speed phase

parameters.

The starting torque, i.e. the torque provided at null speed by each motor, must be at least equal to:

$$T_s = \frac{(\tau_1 + \tau_2 + \tau_3 + \tau_4)}{2\epsilon_m\epsilon_e} \quad (64)$$

where:

- ϵ_e is the electrical efficiency defined in 50. A reasonable value for a BLDC motor is about 80%;
- ϵ_m is the mechanical efficiency, taking into account the mechanical losses especially in the gearbox. A reasonable value for a good gearbox is about 85%;
- The total torque is divided by 2 as there are 2 driving motors in the differential drive configuration.

$$T_s = \frac{(1.875 + 0.02 + 0.55 + 0.05)Nm}{2 \cdot 0.8 \cdot 0.85} \simeq 1.7Nm \quad (65)$$

Regarding the constant speed phase, the continuous torque required to each motor is equal to:

$$T_c = \frac{(\tau_3 + \tau_4)}{2\epsilon_m\epsilon_e} \simeq 0.5Nm \quad (66)$$

This last condition should be as close as possible to the rated operating condition of the motor to be chosen. The nominal continuous power that the motor shall be able to provide is equal to:

$$P_c = \omega T_c \simeq 10W \quad (67)$$

The normal operating conditions of a BLDC motor do not suit the requested values of speed and torque for our application. Indeed, the rated speed is usually of the order of thousands RPM, while the rated torque is of the order of 10^{-2} Nm. This means that a gear reduction is needed in order to increase the provided torque and decrease proportionally the rotational speed. In particular:

$$\tau_2 = \tau_1 * R \quad (68)$$

$$\omega_2 = \frac{\omega_1}{R} \quad (69)$$

where R is the reduction ratio. The number 2 indicates the values of torque and speed at the gearbox shaft, while number 1 refers to the values at the motor shaft. When selecting a gearmotor, it is better to start looking at the rated power rather than at the rated speed or torque. Before delving into the matter, it is prominent to establish the nominal operating voltage of the robot. This value is determined by the battery package choice, that in turn is dictated by the power supply needed by the electronic components of the system architecture, like the Jetson boards, the controller, etc... A value of $12V$ is chosen as V_o .

Once a motor with a rated power close to P_c is found, the curves provided by the manufacturer are analyzed. The rated speed (corresponding to the rated power) is divided by $\omega = 190RPM$ in order to find the ideal value of the gear reduction ratio. Afterwards, the actual ratios provided by the manufacturer's gearboxes are checked, and the closest one is chosen. Now, the rated torque provided by the motor is multiplied by the aforementioned reduction ratio; the resulting value must be equal or greater than T_c . The following step is to check whether the stall torque of this motor is greater than T_s : at null velocity (starting condition), the motor must be able to overcome the torques needed during the acceleration phase.

It is important to notice how some manufacturer only provide the mechanical characteristic in the continuous operating range, stressing how the motor should only work in that specific area during its normal operating condition (in this case, the constant speed phase). However, the stall values of torque and current can easily be found by applying the linear equations 47 and 49.

Other considerations that have to be kept in mind while choosing a motor are the cost and the delivery area of the manufacturer. This is why it has been chosen to select a motor manufactured by Anaheim Automation, a company that builds cost-effective motors, gearboxes and encoders and is set in Anaheim, California, very close to San Diego.

A motor with the mechanical characteristic shown in figure 30 is analysed.

In this case, the manufacturer only provides the continuous operating range. Being an

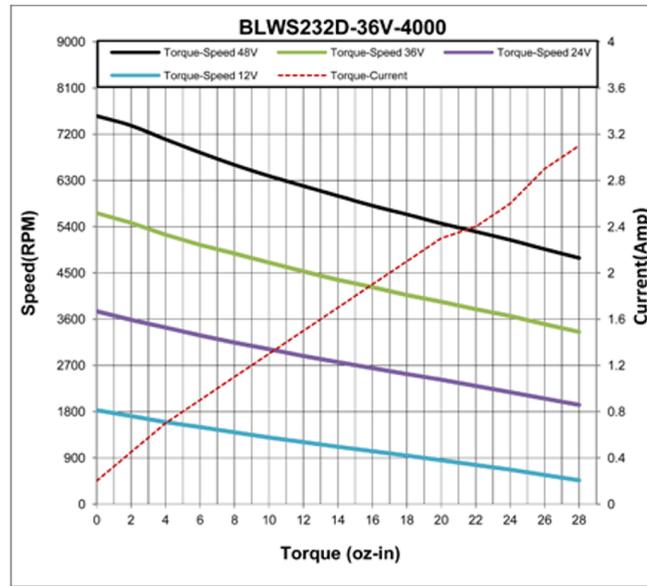


Figure 30 – Mechanical characteristic at different voltages, (23).

American company, the unit of measures of the torque is oz-in (ounce-inch) instead of Nm.

The nominal operating values of this motor are in table 3.

Rated power	42 W
Rated voltage	36V
Rated speed	4000 RPM
Rated torque	0.10 Nm

Table 3 – Rated values for the chosen motor.

Since the power supply of the system can provide 12V voltage, the analysed curve will be the cyan one, meaning that the actual nominal values of interest are in table 4.

Rated power	14 W
Operating voltage	12V
Rated speed	1330 RPM
Rated torque	0.10 Nm

Table 4 – Rated values at a lower voltage.

As already discussed and shown in figure 29, working at a lower rated voltage affects the

values of rated power and speed, while the rated torque is left unchanged. Dividing the rated speed by $\omega = 190RPM$, it is possible to obtain the ideal reduction ratio:

$$R_{ideal} = \frac{1330}{190} = 7 \quad (70)$$

The manufacturer (24) provides a gearbox whose reduction ratio is $R=7.5$, very close to the computed value. This means that, attaching a gearbox to the motor, the rated values of the gearmotor will be the ones listed in table 5.

Rated power	11 W
Operating voltage	12V
Rated speed	177 RPM
Rated torque	0.6 Nm

Table 5 – Rated values of the gearmotor.

The values of rated torque and rated power have been multiplied by the efficiency of the gearbox (around 80%). These rated values are very close to the ones required during the constant speed phase.

Now it is time to check whether the stall torque of the motor is greater than the starting torque required during the acceleration phase. Using the no load speed value and the rated values of torque and speed, it is possible to compute from 47 the values of the stall torque as:

$$T_{stall} = R \frac{T_{rated}}{1 - \frac{\omega_{rated}}{\omega_o}} = 2.9Nm \quad (71)$$

This value is much higher than the one needed for the starting condition, meaning that the motors are able to accelerate the robot.

The last check to be done involves the gearbox: a prominent parameter to consider when choosing it is the maximum amount of torque that it is able to handle. From the data sheet this value is about $4.9Nm$, meaning that the gearbox is able to provide the torque needed without any failure.

Hence, the gearmotor meets all the aforementioned requirements and it is chosen as the driving motor of the robot. Moreover, the motor has a double shaft of its rear, allowing the

attachment of an encoder.

The following picture shows the mechanical drawing of the gearmotor, (24). The front shaft has a diameter equal to 0.315 in that is $8mm$, while the rear shaft's diameter is equal to $0.25in$, that is $6.35mm$. These values will respectively have to be taken into when selecting the wheels and the encoder. The total length of the gear motor is equal to $L1 + L2 = 1.26in + 2.32in = 3.58in \simeq 90mm$, (24).

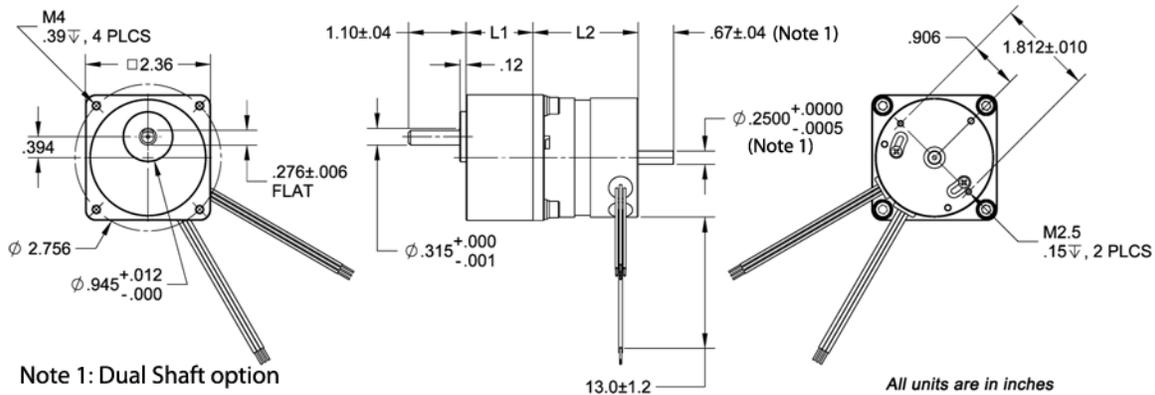


Figure 31 – Mechanical drawing of the selected gearmotor, (24)

4.5 Encoder choice

An encoder is an electro-mechanical device used when position or speed feedback from the motor is needed (28). It is able to transform the mechanical rotation of the motor shaft into electronic signals fed to the controller, that thus can track a desired position or speed profile.

An optical encoder is composed by: a slotted disc, a dual light detector and a light source, as explained in (26). The light source continuously points at the disc that rotates with the motor shaft. The most important parameter when choosing an encoder is its resolution, measured as CPT - counts (or pulses) per turn-, which is the number of slots present in the disc. On the opposite side with respect to the disc, the dual light detector turns the amount of light received into quadrature output pulse signals, called A and B. Each time one of the two pulse signals goes up or down, a counter is increased, thus recording the incremental position of the disc. Hence, the actual resolution is four times the CPT. The direction of rotation can also be detected by determining which signal is leading and which is lagging.

This kind of encoder is indeed called incremental optical encoder, as it is able to provide

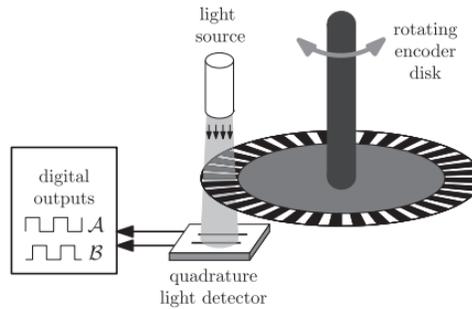


Figure 32 – Incremental optical quadrature phase encoder, (26)

only a relative positional information. A third channel called Index can be added; this channel only provides one pulse per revolution and can be used as an absolute reference for homing functions.

There are two main kinds of speed control through an incremental encoder: pulse counting and pulse timing, (27).

The first one is usually performed with lower resolution encoders and consists in counting the number of pulses n recorded in a specific sampling time t , thus computing the average time for one pulse $\frac{t}{n}$. From this ratio it is possible to compute the rotational speed of the shaft in $\frac{rad}{s}$, given by:

$$\omega = \frac{2\pi n}{Nt} \quad (72)$$

where N is the encoder resolution. This method is well suited for high speed application, as at low speed the resolution may be too poor.

The second method is performed with higher resolution encoders. In pulse timing, a high frequency clock signal is counted during the interval between two adjacent pulses. The number of cycles m that the clock has completed is divided by its frequency f in order to find the time between two successive pitches. This value is useful to compute again the rotational speed of the shaft in $\frac{rad}{s}$:

$$\omega = \frac{2\pi f}{Nm} \quad (73)$$

Pulse timing can not be applied to high speed application, as the time $\frac{m}{f}$ between two pulses may be too short to be properly evaluated.

Since the encoder will be attached to the motor whose rated rotational speed is about $1300RPM$, pulse counting method is preferred over pulse timing. An encoder with a moderate number of CPT and a high maximum speed rating has to be selected, (25).

In fact, if in equation 72 the value of counts n is set to 1, it is possible to obtain the speed resolution. For an encoder with 500 slots (i.e. 2000 CPT), and a controller whose sampling time t is of the order of $1kHz$, the speed resolution is about $30RPM$. Since the motor's operating range is of the order of thousands RPM, such a value is by far acceptable.

A cost-effective solution is offered by Anaheim Automation: the encoder shown in figure 33 has the same bore size of the rear shaft of the gearmotor 31, comprises the index channel for absolute reference and has a number of slots equal to 1000, thus corresponding to an actual resolution of $4000CPT$.

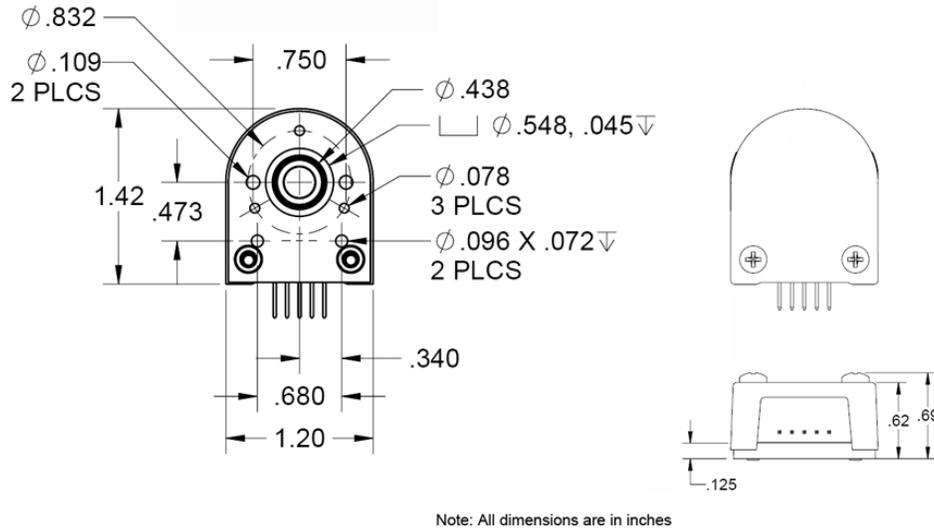


Figure 33 – Mechanical drawing of the selected encoder, (29)

At this point, the wheel configuration has been decided, the power train has been sized and its components (motor, gearbox and encoder) have been selected. Now it is finally possible to focus on the design of the whole system.

5 CAD Modelling

5.1 FDM

Before starting the design of the structural components of the system, it is prominent to introduce the manufacturing technique that will be deployed. Each technique has its own benefits and limitations, as well as specific design guidelines that have to be followed in order to come up with feasible models. Indeed, avoiding the most common design errors is a key factor especially when money and time are limited.

In the robotic field, the additive manufacturing process is a method that ensures very fast building time, together with good structural properties and a large variety of cost-effective raw materials, (32). In general terms, it consists in several techniques in which a layer of material is built upon the previous one. Thus, instead of cutting holes and specific shapes in an initial raw piece (like in the subtractive manufacturing processes), the part is built adding material layer by layer as seen in figure 34.

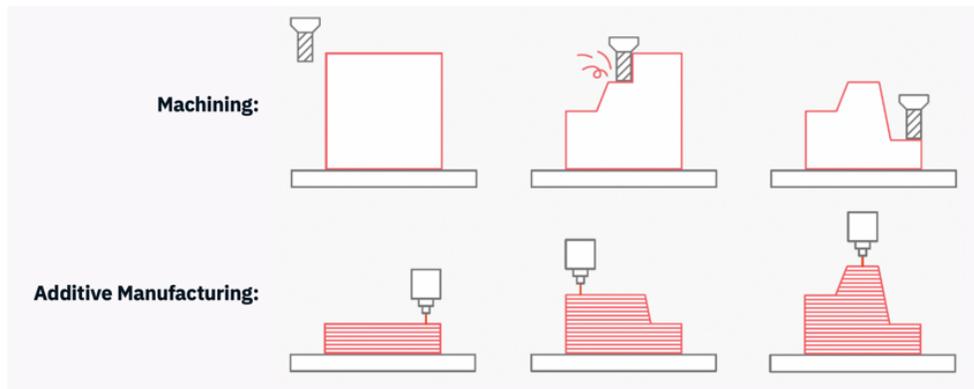


Figure 34 – Subtractive machining VS Additive manufacturing process, (33)

The steps to be followed when designing a part for additive manufacturing are as explained in (32):

- 3D modelling: through a specific Computer Aided Design (CAD) software, a 3D digital model of each part is designed following the guidelines and geometry constraints corresponding to the chosen manufacturing method. In this thesis, Solidworks has been used;

- **STL conversion:** the CAD file is converted in a STL file, in which surfaces are described by polygons (usually triangles). Secondly, a slicer program transforms the STL file in G-code, a particular programming language that is able to interact and control the automated tool used to produce the part;
- **Printing:** the raw material is loaded into the machine in different forms depending on the technique used (filament, powder, liquid resin), and the actual production process can start;
- **Removal of the print and of the surplus material:** the print is removed from the base and eventual support material is cut from the main part;
- **Post-processing:** This step may include coloring, air-cleaning, polishing as well as eventual hole-tapping.

Among the different additive manufacturing processes, FDM (Fused Deposit Modelling) or FFF (Fused Filament Fabrication) is one of the most famous method for rapid prototyping and building a system, (34). Usually thermoplastic polymers are used as raw materials.

The plastic is inserted in the printer in a specific spool connected to a nozzle, which is heated up to a temperature allowing the melting of the material. The nozzle is attached to a 3-axis system that can move in all the three directions. The melted material is extruded through the hot nozzle, that deposits it layer-by-layer. The plastic cools down and solidifies while the nozzle moves up and deposits a new layer of material. In some printers, it is the build platform moving down while the nozzle maintains its vertical location as in figure 35.

This manufacturing technique also allows a great freedom in choosing some important parameters, as the temperature of the nozzle and the build platform, the speed at which the nozzle moves and the height of the layers. The latter parameter can especially have a prominent role in the quality of the finished part. Indeed, a small layers height implies very good details quality and smooth surfaces, while increasing time and cost of the printing process. Thus, a reasonable trade off based on the specific designer's need has to be found.

One of the main disadvantage is the maximum build size of the printers; desktop printer usually have a maximum limit of 200x200x200 mm, while industrial machine can at most reach 1000x1000x1000 mm. This means that if a bigger part has to be manufactured, it is mandatory to split it into smaller parts to be assembled later, (34).

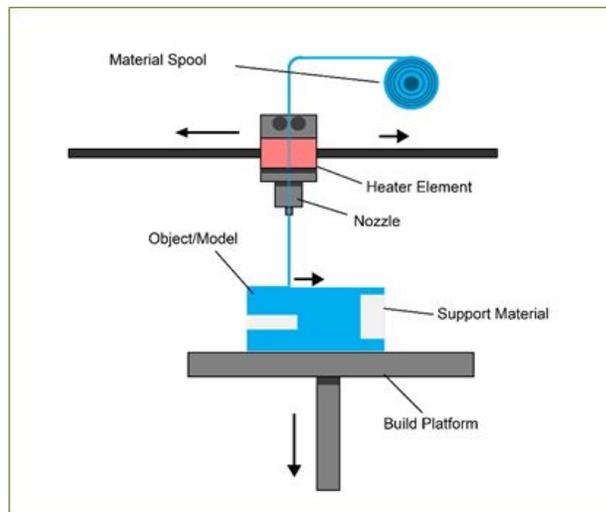


Figure 35 – Fused deposit material process with platform moving in the Z direction, (36)

Another drawback is given by the fact that the cooling process of the printed layers tends to shrink the dimension of the final part; this can have serious consequences when assembling together the whole system. Moreover, cooling does not occur in an uniform way all over the part, so that internal stresses may arise leading to warping. This especially occurs in large flat areas, in which the underlying layers are pulled upwards by the internal stresses caused by cooling, as it is possible to see in figure 36. Also sharp corners are subject to warping, as stresses are more intense in their nearby.



Figure 36 – Warping in large flat areas, (34)

Another intrinsic issue regarding FDM is the anisotropic structural properties of the parts. This directly comes from the fact that bond strength between the Z axis, i.e. the various layers, will always be weaker than the base strength of the material in the X-Y plane (34). This means that the part will inherently have a very different behaviour under different load conditions, as shown in figure 37.

Moreover, parts printed with FDM technology will most likely have visible layer lines,

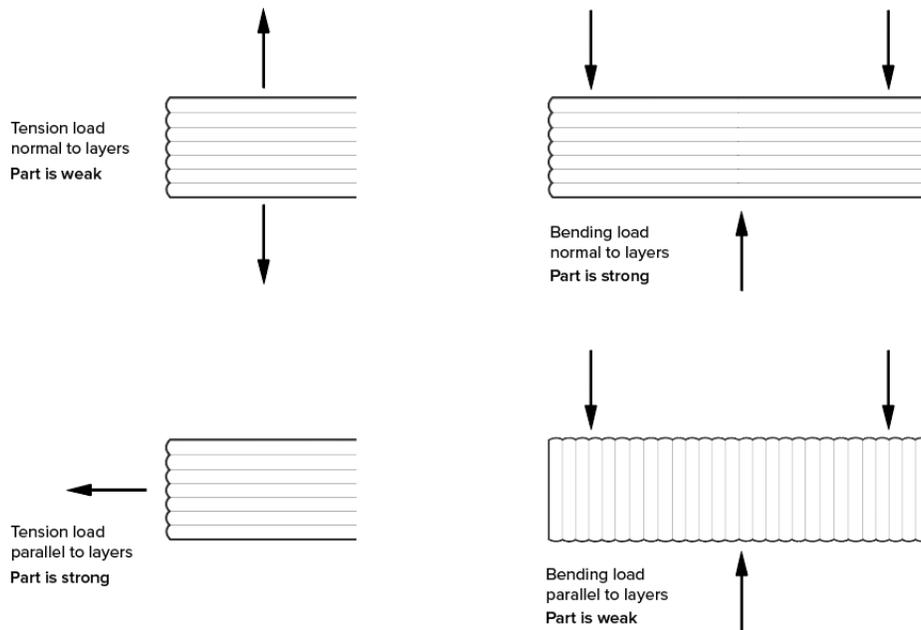


Figure 37 – Different behaviour under different load conditions, (35)

depending of course on the chosen layer height.

All of these drawbacks can though be mitigated by following accurate design guidelines or post processing methods, that allow to fully exploit the benefits of FDM.

5.2 Printing Material

One of the most common material used in FDM 3D printing is Acrylonitrile Butadiene Styrene (ABS), an opaque amorphous thermoplastic "comprised of three monomers, acrylonitrile, butadiene and styrene", (40). The properties of this polymer are a combination of the properties of each individual component it is comprised of, resulting in a very appealing printing material. Indeed, the Acrylonitrile provides thermal and chemical stability, developing a polar attraction with the other two monomers. Butadiene is the monomer mainly responsible for toughness and strength ABS is characterized by. This provides ABS a very high impact resistance combined with a great flexibility, meaning that parts printed in ABS will distort, then bend, before finally breaking, (42). Moreover, it is also the component causing ABS to have an opaque milky look. Lastly, Styrene gives the polymer its typical shiny, glossy finish surface.

ABS is one of the cheapest printing thermoplastic and this is the reason why it is commonly used for low-cost prototyping and design verification. It can be easily post processed after printing; procedures like drilling, tapping or colour dyeing can be carried on more easily with respect to other thermoplastics.

Glass transition and melting temperature are very important parameters when dealing with 3D printing materials: in fact, the nozzle temperature must be kept above the latter, as the extruded plastic has to be molten, as said in the previous section. Build platform temperature instead must be kept below the glass transition one in order not to alter the printing process. For ABS, glass transition temperature is about 104° , making it a quite heat resistant material. Indeed, it could be heat up to more than 100° and still keep its mechanical properties. Being an amorphous material, it has not a proper melting point but it liquefies well at around 200° C.

This means that nozzle temperature has to be taken up to at least $210^{\circ} - 250^{\circ}$ C. Build platform temperature value is influenced by the following issue: ABS is quite sensitive to temperature changes. This makes it heavily prone to warping, especially when large flat pieces are being printed. In order to avoid warping, a smooth and slow cooling process should be achieved. This can be done by adjusting the build platform temperature up to at least 100° C for the initial layers, and keeping it higher than $80^{\circ} - 90^{\circ}$ C afterwards. If a cooling fan is embedded in the build platform, it should be turned off during printing of large flat parts, (41). Moreover, it is suggested to enclosure the printer during its functioning with ABS in order to avoid fast temperature changes that could also lead the part to shrink, resulting in dimensional inaccuracy.

5.3 Design for Manufacturing

The design guidelines to be followed when designing for 3D printing are due to the intrinsic nature of the additive manufacturing process and are based on the so called Design for Manufacturing and Assembly (DFMA). It is a general practice allowing a "product design to be efficiently manufactured and easily assembled with minimum labor cost", as said in (37). It prevents unfeasible and unobtainable design, and helps saving time and cost of production.

In FDM 3D printing, one of the main geometry constraints arises from the following issue: each layer is built on top of the previous one, and of course building a layer on thin air is

not possible. This leads to some design considerations.

A common way to clarify the consequences of this issue on the design phase is given by the Y-H-T rule, shown in figure 38:

- Letter Y can be successfully manufactured, as the inclination of the two upper branches is relatively low, and this gradual slope allows the printer to reach good layer adhesion and final surface quality, keeping the material from dropping;
- Letter H can cause problem in the manufacturing process, as the bridge between the two vertical lines has not a lower support layer. If the bridge length overcomes a certain value, bulging and curling of the material could occur. This value depends on the material, the machine and the technique used;
- Letter T will certainly cause the aforementioned problems when manufactured, as the overhanging horizontal branches have completely no support to lie on.



Figure 38 – Y-H-T rule in 3D printing, (38)

Hence, any design involving inclined branches, bridges or horizontal overhangs must be designed carefully.

A first common design rule sets to 45deg the maximum slope that a 3D printer can manufacture maintaining good surface quality and structural properties. As it is possible to see in figure 39, exceeding this limit gradually leads to poor and eventually inadequate printing results.

Regarding the bridging issue, it is convenient not to design any bridge whose length is greater than 5mm. Overcoming this limit will certainly lead to sagging and curling of the material, (35).

Finally, when an horizontal overhang is strictly necessary in a part, it can be useful to rearrange the printing orientation, as it could completely eliminate the problem. For example, printing the letter T upside down will solve this issue as no overhanging branches are present.



Figure 39 – Printed part with gradually increased inclination, (35)

If no one of the previous design rules can avoid the presence of overhanging parts or a long bridge is needed, vertical supports must be used. Supports can be realized with the same raw material used for the main part, as well as with a different material spread by a secondary nozzle, (34). By using supports one can design horizontal branches as well as bridges and inclined parts. This comes at the price of wasting additional material that eventually has to be taken off manually after printing. Moreover, the removal process usually leaves marks on the final surface unless post-processed, (35).

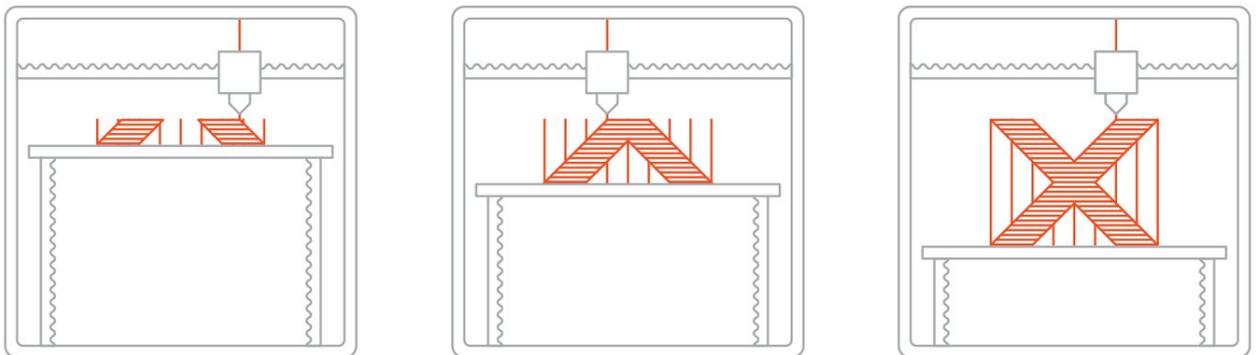


Figure 40 – Printed part with support structure, (39)

Figure 40 clearly shows an example of a part (letter X) that necessarily needs vertical supports to be properly manufactured.

Another issue related to the FDM printing method regards holes. In general, a vertical axis hole is preferred over a horizontal axis one, as the shape of the latter will hardly be perfectly rounded since the circumference of the hole would be constructed layer by layer. However, FDM eventually "often print vertical axis holes undersized", (35). During the printing process, in order to ensure layer adhesion, the nozzle compresses the new layer on top of the previous one. The circle parallel to the X-Y plane experiences a deformation to a "wider and flatter shape", (35), due to the compressing force. The wider area of contact between the two layers improve their adhesion, though resulting in a deformed extruded cut shaped like an ellipse.

This means that the actual diameter of the hole is eventually reduced, corresponding to the minor semi-axis of the ellipse. When a very precise hole diameter is needed, it is recommended to print it undersized, and then drill it during post-processing.

When dealing with sharp edges and corners, nozzle compression also causes the first layers of a print to be slightly wider than expected, resulting in a sort of elephant's foot. This can be avoided by adding " a 45 deg chamfer or radius on all edges touching the build plate", (34).

Other general design guidelines regards:

- Wall thickness: for FDM a minimum wall thickness of $0.8mm$ should be respected;
- Fillets should be preferred over sharp corners as the latter increase stress concentration;
- Engraved details or text should be preferred over extruded ones, and should not go below the lower dimensional limits shown in figure 41.

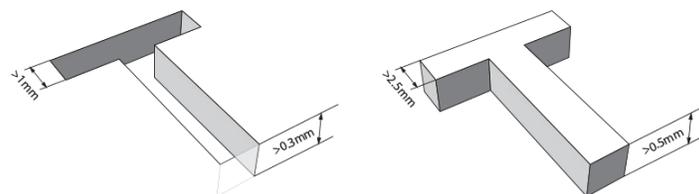


Figure 41 – Lower dimensional limits for engraved and extruded details, (34)

5.4 Design for assembly

While Design for Manufacturing refers to guidelines and common considerations applied to the production of a single part, Design for Assembly regards a series of engineering practises adopted with the aim of easing the assembly process, reducing time and cost related to these procedures. Design for Assembly (DFA) heavily influences series industrial production, as the minimization of time and effort payed by the workmen in the assembly process becomes one the key factor in production optimization, (43). Nevertheless, also when designing for rapid prototyping, it is convenient to pay attention to DFA guidelines as much as possible. Indeed, including features and details suggested by DFA will optimize the assembly process and reduce the related costs.

The main principles of DFA are aimed at avoiding time-consuming operations while mounting the assembly and at reducing as far as possible potential confusion and misunderstanding in the assembling procedures. In general terms, it is highly recommended to:

- Minimize part count: "cost of assembly decreases as the number of components to assemble is reduced and the quality and consistency of the assemblies also improves", (43). Thus, it is suggested to design separate parts only when strictly necessary for the functional behaviour of the system or depending on printing size limitations;
- Design parts with self-locating and self-fastening features: this will drastically increase consistency and ease of assembly and reduce potential mistakes or confusion while mounting, as each part has a self-evident location inside the assembly;
- Minimize reorientation of parts during mounting and enhance top-down approach. "Ideally, an assembly should stack one part on top of another with only the help of gravity", (43). It is convenient to start the assembly process from a base and then continuing by locating each following component on top of it, avoiding any reorientation of the base itself;
- Use of standardize parts and minimum use of threaded fasteners: as shown in (44), the use of threaded fasteners can slow down the mounting process, consuming 20 – 50% of assembly labor. This means that, whenever they are not necessarily needed, it is strongly suggested to avoid their use, preferring other locking features that will be

described shortly after. When it is not possible to substitute threaded fasteners, it is mandatory to use standardize parts. Moreover, it is highly recommended to the same standardize threaded part for the entire assembly, in order to use a single fastening tool for the whole system.

In order to avoid the use of threaded fasteners, several options can be taken into account in order to join two parts together. The most common ones are snap-fit connections and interlocking joints. Not only they can easily be 3D printed together with the main part, but they also reduce the number of overall components and represent intuitive self-locating targets, helpful for the user. Hence, both options are useful to follow more closely the DFA guidelines listed above.

The most common type of snap-fit connection is the Cantilever Snap-fit shown in figure 42. It consists in a protrusion located at the end of a Cantilever beam. This protrusion will be inserted in a slot specifically cut for this purpose, and will deflect upon insertion. "Once fully inserted the protrusion bends back locking the connection into place", (45). When 3D printing a Cantilever snap-fit connection, some expedients should be considered in order to reduce the stress on the beam. First of all, it is convenient to fillet the base of the Cantilever in order to distribute the stress in a broader area, resulting in a stronger connection.

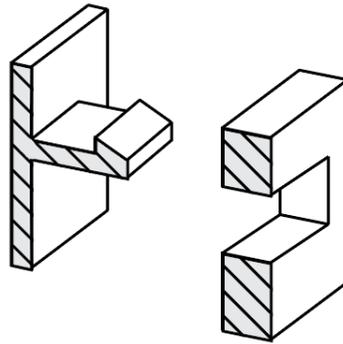


Figure 42 – Cantilever snap-fit connection, (45).

Moreover, a higher width of the clip corresponds to higher strength of the design; a minimum width of $5mm$ should be considered. The beam should be only deflected during the insertion, so that the final position of the snap-fit has to be designed accordingly. Furthermore, it is prominent to take care of the build direction: as already stated, 3D printing is

an anisotropic process, providing stronger bond in the X-Y plane rather than along the Z axis. This means that, when possible, it is highly recommended to avoid Cantilever snap-fit built vertically in the Z direction. In fact, as explained in (46), this drastically reduces the elongation before break of the beam down to about 50% with respect to a beam built horizontally.

A possible self-locating feature that can be added to 3D printed parts are interlocking joints. They are usually chosen when two parts have to be regularly assembled or disassembled. Interlocking joints provide a simple method of connection that ensures the reduction of the number of overall components of the assembly; they simplify the mounting procedure thanks to the intuitiveness of their functioning; lastly, they can help to overcome printer limitations when overhangs and bridges would be otherwise needed to connect two parts together, (48). In order to fit the parts together, a certain space between them is needed. This value, called clearance, can range from $0.2mm$ to $0.6mm$, (49). The smaller is the clearance, the tighter is the joint. This leads to an increased friction between the parts that will eventually result in a higher force needed to pull them apart. Friction has to counteract tension, i.e., the force that tries to pull the joint apart as seen in figure 43. Also shear forces have to be considered; they are perpendicular to the tension and try to pull the joint sideways.

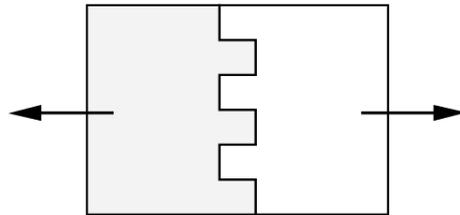


Figure 43 – Tension acting on an interlocking joint, (48).

Finally, another useful and intuitive self-locating feature is shown in figure 44. A wall is horizontally extruded from part A to part B; at the same time, an horizontal cut is extruded in B to locate the wall. A threaded fastener could be used to lock the parts and join them together. This feature will be particularly useful in the design of this project, as will be explained later.

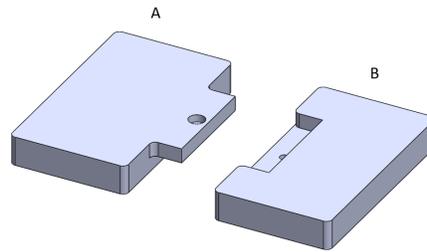


Figure 44 – Self-locating mounting feature.

5.5 Description of the main components

The requirements and guidelines provided both by the DFM (Design for Manufacturing) and by the DFA (Design for Assembly) have been applied to finally model the components of the robot in Solidworks, to design their mounting features to eventually 3D print them.

As already stated in subsection 1.3, the internal structure of the robot will be divided in horizontal layers surrounded by an external rounded shaped body on top of which the tablet for user interface will be mounted.

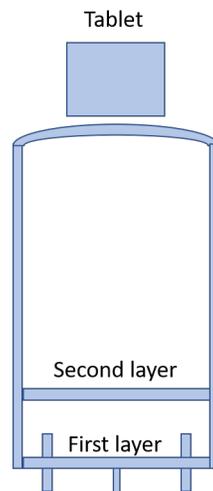


Figure 45 – Scheme of the internal structure of the robot.

In particular, as shown in figure 45 the structure of the robot will be comprised of two layers: a first one hosting the powertrain; a second layer hosting the main electronic boards and sensors; a head hosting both the tablet for user interface and the stereo-camera; a

cylindrical shaped body wrapping the two horizontal layers and supporting the head of the robot.

In figure 45, also the two active wheels and the front idler wheel have been reported.

In this subsection, the components are described individually, highlighting their functionalities and their main dimensions.

5.5.1 First layer

The first layer of the robot is the lowest one; it is mainly dedicated to traction, thus hosting the elements of the driving system. The detailed list of the components is described in table 6.

Component	Main Dimensions [cm]	Quantity
Active wheel	15.24 x 5.08	2
Flanged Hub	6.67 ϕ x 3.05	2
BLDC Motor	5.89 (length)	2
Gearbox	3.20 (length)	2
Encoder	1.75 (length)	2
Idler wheel	6 x 6 (mounting plate)	1
Motor support	6.5x6.5x6.5	2
Battery	15.1 x 9.8 x 9.5	1
Caster support	6 x 6	1
Cooling fan	9 x 9 x 2.5	2
Relay	5 x 7.6	1

Table 6 – List of the components in the first layer.

Each gearmotor is attached to its corresponding active wheel via a flanged hub, whose bolt pattern matches the wheel's one; the rear shaft of each gearmotor is attached to an encoder. Finally, two specifically designed L-shaped supports fix the motors in their position.

The battery is the heaviest components of this layer, and it is prominent to put it right in the center to ensure a proper mass balance. Usually a battery holder is designed and printed to hold the battery in its position. Nevertheless, it has been decided to avoid building a specific holder for cost-saving, so that the battery is held in its position by supporting walls extruded in the layers. A rectangular cut shall be extruded right underneath the battery in order to avoid overheating issues due to the contact between the working battery and the

plastic material. The presence of a Relay between the charge port and the battery prevents the system from current peaks.

Moreover, a cooling fan is needed in order to ensure air circulation inside the body of the robot to prevent overheating. Hence, it is necessary to design several air holes underneath the fan in order to make the air go out the body.

The layer is circular, as the robot shall have a cylindrical shape as said in section 1.3. In differential drive configuration, the active wheels can either be located at the middle of the platform or in the rear part, as shown in figure 46. In the latter case, an idler wheel is necessary to stabilize the robot and prevent it from tilting forward. This configuration ensures that each wheel is always in contact with the ground, as a plane (the ground plane in this case) is only defined by 3 points. The supporting surface upon which the center of mass has to lie is triangular.

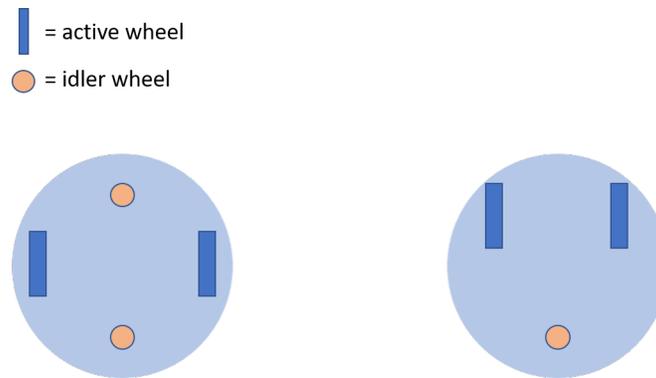


Figure 46 – Active wheels located at the middle (left) and at the rear (right) part of the platform.

Putting the wheels in the middle of the platform implies the need of 2 idler wheels for weight balancing to prevent the robot from tilting either forward (or backward) if the center of mass lies towards the front (or rear) of the body. This configuration does not ensure that all 4 wheels are always in simultaneous contact with the ground plane. Nevertheless, the supporting surface is greater than in the previous case and each wheel has to bear a lower weight.

Eventually, it has been decided to use the triangular configuration with only one idler wheel in the front, in order to ensure proper ground contact of each wheel notwithstanding the actual position of the center of mass. A swivel caster wheel has been chosen for the

purpose and a support has been designed to be mounted on the first layer in order to fix the caster mounting plate in its position.

The diameter of the first layer is heavily influenced by the size of the printer build platform (25cm x 25cm x 21cm). According to these dimensions, it has been decided to design a layer whose diameter is 50 cm. Hence, it will be necessary to split the first layer just in to four smaller parts that will be assembled together after printing.

The design of the first layer is shown in the following figure 47.

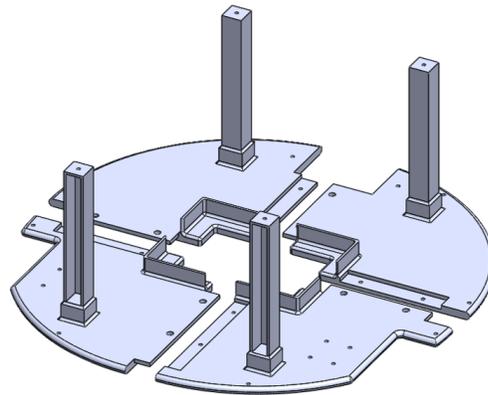


Figure 47 – Exploded view of the assembly of the first layer.

As it is clear from picture 47, each part comprises the self locating feature described in figure 44. The thin walls in the center of the layer (figure 47) define a rectangle in which the battery will be located. Four vertical stand-off are extruded directly in the layer, and will be useful in supporting the second layer and mounting it upon the first one, thus enhancing the top-down approach suggested by DFA.

The printing orientation of these parts has to be equal to the one shown in figure 47: in this way, no issues related to overhangs or bridging occur. The lower surface of each parts has been filleted in order to avoid the elephant's foot problem described in DFM. Warping could be an issue due to the fact that these surfaces are large and flat; nevertheless, heating up the build platform can reduce the problem.

5.5.2 Second layer

The second layer hosts the components listed in table 7 . With respect to the architecture described in figure 9, some slight modifications in the electronic equipment have been performed during the development of the system. First of all, it has been decided to use two Jetson Nano instead of a Jetson TX2, as the former reach a satisfactory computational power yielding a lower cost. Moreover, the Wi-Fi module has been substituted with a router.

Component	Main Dimensions [cm]	Quantity
Jetson Nano	6.9 x 4.5	2
LIDAR	7.6 ϕ x 4.1	1
Slamware board	9.1 x 12	1
Wi-Fi Router	5.1 x 1.3	1
RCU	8 x 6	1
Relay	5 x 7.6 x 1.5	2

Table 7 – List of the components in the second layer.

The structure of the second layer is way simpler with respect to the first one, as it only has to carry the boards listed in table 7 that can be easily mounted on top of it without the need of any particular feature.

An important issue to consider when designing the model for the second layer is the position of the LIDAR. As already stated, its function is to detect objects and obstacles along the path of the system and in its surroundings, thus it should have a clear view of where the robot is heading towards. In order to get a wider and more comprehensive view, it is better not to put the LIDAR along the circumference of the layer, as it would miss the information regarding the lateral surroundings of the body. The optimal position of the LIDAR is shown in figure 48.

This means that half of the second layer has to remain empty to ensure good LIDAR performances. Thus, all the other components of table 7 have to be located in the rear half of the layer. Additionally, several air holes have to be designed to allow air circulation inside the robot.

Also the second layer has to be split in four parts as the first one; again, the same locating features will be used to join the four parts together. The second layer can be directly mounted on top of the first one thanks to the presence of the stand-off extruded in the first layer.

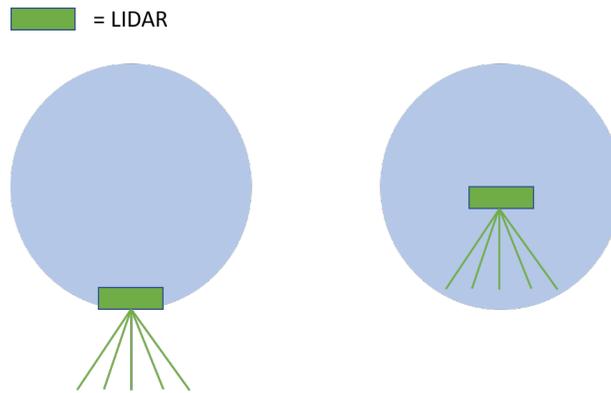


Figure 48 – Poorly located LIDAR (left) vs Optimal position (right).

5.5.3 Base frame

This component is the lower part of the chassis of the robot, wrapping the two aforementioned layers in a cylindrical shaped body. Moreover, it also hosts some prominent components mainly related to charging and control. The detailed list of the components present in this part is given in table 8. The narrow FOV LIDARs described in figure 9 have been replaced by ultra sonic sensors.

Component	Main Dimensions [cm]	Quantity
Ultrasonic Sensor	4.5 x 2 x 1.5	4
HDMI port	1.47 x 1.93	1
USB port	1.47 x 1.93	1
Ethernet port	1.47 x 1.93	1
Status LED	1.2 ϕ	3
Power ON/OFF switch	1.3 x 1.94	1
Charge port	1.5 x 0.6	1

Table 8 – List of the components in the lower frame.

It is convenient to design a control panel located in the rear part of this frame in which the interface components are inserted; in particular, this control panel will host the HDMI, USB and Ethernet port, as well as the power ON/OFF switch, the charge port and 3 status LED whose task is to indicate the following system conditions: a red one means that a generic fault is occurring; a blue one states that the system is in normal operating conditions; a green one indicates that the robot is recharging.

The design of this frame will have to be split into multiple parts in order to fit the build platform size. In this case, also the Z axis printing dimension limit has to be taken into account, as it is not possible to print parts higher than 21 cm. It has been decided to split the design of the lower frame in to eight parts. Four lower parts having the shape of a quarter of a hollow cylinder will be mounted upon the first layer, thus wrapping the components related to the drivetrain. The height of these parts is equal to the distance between the first and the second layer, so that since the latter is directly mounted upon the first layer, its vertical position corresponds to the height of these four cylindrical components. The vertical distance between the first and the second layer is determined by the LIDAR vertical position, that has to be located at 20 cm from the first layer for optimal performances. The exploded view of these four parts is given in figure 49.

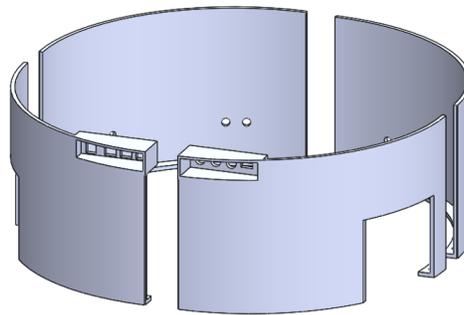


Figure 49 – Exploded view of the assembly of the lower base frame.

It is convenient to attach the design of the control panel to the lower part of the base frame; this will avoid adding a new component to the whole assembly. The interface components located in the control panels are mounted by means of snap-in connectors, thus simplifying the mounting procedure. The HDMI port is shown as an example in figure 50. It is clearly visible the Cantilever snap-fit previously described.

Furthermore, eight circular holes will be extruded in the front part of the lower frame of the base in order to locate the ultra sonic sensors, whose vertical distance from the first layer has to be equal to 5 cm.

The remaining four upper parts of the base frame are shown in figure 51 and will have again the shape of a quarter of a hollow cylinder, and will be mounted upon the second

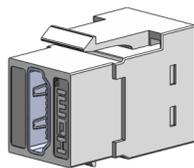


Figure 50 – HDMI port with Cantilever snap-fit connection.

layer. The two front quarters of these upper parts leave to the LIDAR a proper angular clearance. In fact, as already explained previously, the LIDAR shall have a clear view of its surroundings, meaning that the frame does not have to wrap the space directly in front of the sensor.

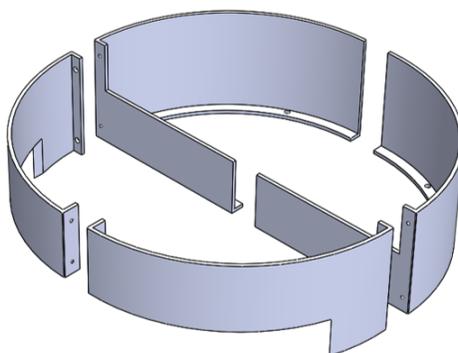


Figure 51 – Exploded view of the assembly of the upper base frame.

The extruded vertical walls in center of the frame are useful to separate the LIDAR from the remaining electronic components located in the rear part of the second layer.

Printing orientation can be exploited in order to avoid bridging issues, so that the two front quarters of the upper base frame will be printed upside down; all the other parts of the base frame will be printed as they are shown in figures.

In this way, the use of support material is avoided as much as possible. In particular, it will be necessary to use supports in order to attain the circular holes for mounting the ultrasonic sensors; moreover, support material will be needed also in the extruded cuts in

the base frame corresponding to the wheels positions.

The top-down approach suggested by DFA is applied, as each part just described is mounted on top of the previously mounted one.

5.5.4 Body and head

For privacy issues, it is not possible to show the detailed design of the body of the robot. Nevertheless, it is possible to describe the main functional components that will be embedded in this part; the detailed list is shown in table 9.

Component	Main Dimensions [cm]	Quantity
Tablet Galaxy S4	24.9 x 16.4 x 0.71	1
Stereo Camera	17 x 3 x 3.3	1
E-Stop	3.7 x 3 x 7	1

Table 9 – List of the components in the upper frame and head.

Other than the tablet and the stereo camera, already mentioned in the system architecture description, it is strongly suggested to include in the body design an emergency stop button (E-stop) that can be easily used in potentially hazardous situations. Since it should be promptly reachable, it is convenient to put it in the external body frame of the robot, so that it is clearly visible and easily usable.

The body will be directly mounted upon the base frame following the top-down approach.

5.6 Base Assembly

Now it is possible to assemble in Solidworks all the structural components previously described, adding the components listed in tables 6, 7 and 8.

The front view of the base frame is shown in figure 52. It is possible to notice the LIDAR sensor located at the front of the second layer; furthermore, it is also possible to see the four ultrasonic sensors (colored in orange) located in the lower base frame.

Furthermore the actual components related to the drive train have been included in the model, as well as the cooling fan, the relay and the battery and are clearly shown in figure 53.



Figure 52 – Front view of the base assembly.

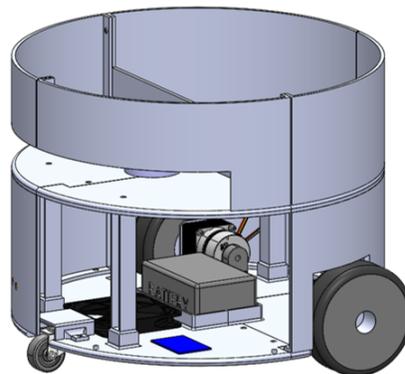


Figure 53 – Detailed view of the drive train components in the base assembly.

Most of the other electronic boards are located in the rear half of the second layer and are represented by symbolic colored rectangles displaying the real dimension of the boards as shown in figure 54.

In particular, the red board is the Jetson Nano; the cyan one is the router; the purple one is the RCU; the yellow board is the slamware. The two Jetson Nano can be mounted one on top of the other, thus saving space in the second layer.

According to the guidelines of DFA, a single size of standard threaded fasteners (M4x0.7) has been selected and adopted in the whole structure, in order to speed up the mounting process. The Ansi Metric standard has been selected since the printing process and assembly procedure will be deployed in the United States.

The mass of the structural components of the base assembly is computed through Solid-

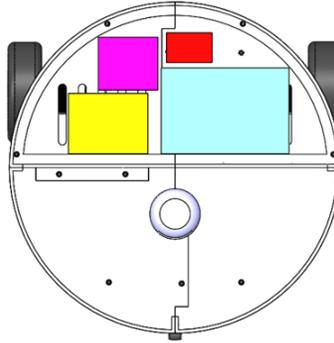


Figure 54 – Top view of the base assembly.

works and it is equal to 7.2 kg for a ABS density of about $1.07 \frac{kg}{m^3}$. The final footprint of the base is 50 cm diameter per 40 cm of height.

The computed weight can be added to the weight computed through Solidworks of the remaining structural parts (body and head) and to the ones of the other electrical and mechanical components found in the datasheet. A total value of 35 kg is estimated to be the final mass of the robot. The previous assumption of 50 kg was higher and more conservative, meaning that the chosen power train will be able to actually drive the platform.

The base frame components are thus ready to be printed and assembled, so that the autonomous navigation software can be tested on the platform.

6 Conclusion

The work of this thesis regarded the mechanical design of a wheeled mobile robot servicing San Diego International Airport, in collaboration with Innotech, a newborn start up based in San Diego, California. A strong interaction with the Airport Committee has been carried on during the whole duration of the work to earn continuous feedback mainly coming from customer surveys and airport staff attitude towards this project. The design process followed successive steps, starting from a benchmark analysis of similar firms to the detailed mechanical modelling of the system.

In particular, robotic systems developed by companies like Savioke and LG have been

studied to get delightful insights about the main challenges that mobile service robotics is facing nowadays, as well as to gain knowledge about prominent guidelines to be adopted when dealing with the design of an autonomous robot. The analysis of the mechanical framework of these systems resulted in the choice of a layers-oriented structure.

Five different concepts of the robot design have been developed and submitted to the Airport Committee, that chose the one preferred by customer surveys.

Robotic mobility and its main issues have been studied in details in order to get the most suitable driving system for this project; several wheel types and configuration have been compared with respect to their mechanical performance, the cost and simplicity related to their implementation and the easiness of motion control and path planning. The differential drive configuration has been chosen and its kinematic and dynamic models have been reported to get a deeper knowledge about its features.

A brief review of the most commonly adopted electrical motors in robotic field led to the choice of using brushless DC motors as the driving motors of the system. The entire drivetrain, comprising also the gearboxes and the encoders, have been sized and specific commercial components have been selected.

The 3D printing method has been chosen to build the robot: consequently, the main features regarding the additive manufacturing process have been highlighted to assess its main benefits as well as the most prominent limitations due to the intrinsic nature of the method. An overview on the guidelines coming from DFMA (Design For Manufacturing and Assembly) was helpful to learn the most stringent rules to be followed as well as the common design errors to be avoided.

Finally, the 3D models of the system have been developed in Solidworks, exploiting all the requirements and specifications previously collected. A detailed design of the base frame is provided, and some of the most distinctive mounting features and procedures are described. The main drawbacks and constraints related to 3D printing have been tackled and overcome, resulting in a feasible cost-effective design.

In the coming period, the remaining components of the base frame will be printed and mounted on top of the first layer, in order to eventually assemble the whole structure.

List of Figures

1	Processing power available per dollar over time, (1)	4
2	U.S. automotive LiDAR market by application, 2014 - 2024 (USD Million), (2)	5
3	Relay by Savioke	6
4	Airport assistant robot by LG	7
5	Carrying luggage Aethon's assistant robot	7
6	Knightscope Fleet	8
7	The Uncanny Valley, (7)	9
8	Internal structure of Relay	10
9	High level system architecture of the system	12
10	ZED Stereo Camera	13
11	Assistant robot concept 1 and 2	15
12	Assistant robot concept 3 and 4	16
13	Assistant robot concept 5	16
14	Fixed, steered and pivot standard wheel.	19
15	Swivel caster (left) and rigid caster (right), (17)	19
16	Universal, double universal, mecanum omni wheel.	20
17	Tricycle drive configuration.	23
18	Differential drive configuration.	23
19	Skid steering configuration.	25
20	3 Omni wheel drive configuration for translation.	26
21	3 Omni wheel drive configuration for rotation.	26
22	Synchronous drive configuration.	27
23	Ackermann steering configuration.	28
24	Differential drive geometry.	31
25	Rotation about the ICC of $\omega\delta t$.	32
26	Inertial reference system and robot reference system, top view.	35
27	Left view.	35
28	BLDC motor curves	42
29	BLDC motor curves at different voltages.	43
30	Mechanical characteristic at different voltages, (23).	49

31	Mechanical drawing of the selected gearmotor, (24)	51
32	Incremental optical quadrature phase encoder, (26)	52
33	Mechanical drawing of the selected encoder, (29)	53
34	Subtractive machining VS Additive manufacturing process, (33)	54
35	Fused deposit material process with platform moving in the Z direction, (36)	56
36	Warping in large flat areas, (34)	56
37	Different behaviour under different load conditions, (35)	57
38	Y-H-T rule in 3D printing, (38)	59
39	Printed part with gradually increased inclination, (35)	60
40	Printed part with support structure, (39)	60
41	Lower dimensional limits for engraved and extruded details, (34)	61
42	Cantilever snap-fit connection, (45).	63
43	Tension acting on an interlocking joint, (48).	64
44	Self-locating mounting feature.	65
45	Scheme of the internal structure of the robot.	65
46	Active wheels located at the middle (left) and at the rear (right) part of the platform.	67
47	Exploded view of the assembly of the first layer.	68
48	Poorly located LIDAR (left) vs Optimal position (right).	70
49	Exploded view of the assembly of the lower base frame.	71
50	HDMI port with Cantilever snap-fit connection.	72
51	Exploded view of the assembly of the upper base frame.	72
52	Front view of the base assembly.	74
53	Detailed view of the drive train components in the base assembly.	74
54	Top view of the base assembly.	75

List of Tables

1	Wheel configuration comparison	29
2	Torque required during acceleration, constant speed and deceleration phases.	44
3	Rated values for the chosen motor.	49
4	Rated values at a lower voltage.	49
5	Rated values of the gearmotor.	50
6	List of the components in the first layer.	66
7	List of the components in the second layer.	69
8	List of the components in the lower frame.	70
9	List of the components in the upper frame and head.	73

References

- [1] AI Impacts. (2015). <https://aiimpacts.org/trends-in-the-cost-of-computing/>
- [2] Robert Bosch GmbH, Denso Corporation, Velodyne LiDAR, Inc., Novariant, Inc., LeddarTech, Continental AG, and Quanergy Systems, Inc.(2016). Automotive LiDAR Market Analysis By Application Type (ADAS And Autonomous Cars), By Region (North America, Europe, Asia Pacific, Latin America, And Middle East And Africa), And Segment Forecasts, 2018-2024.
- [3] Kone. (2018) <https://www.kone.com/en/news-and-insights/stories/the-robot-butler-is-coming-to-a-hotel-near-you.aspx>
- [4] LG. (2017) <https://www.lg.com/sg/press-release/lg-airport-robots-take-over-koreas-largest-airport>
- [5] Aethon. (2018) <https://aethon.com/sheraton-hotel-to-use-tug-robots/>
- [6] Melia Robinson. (2017). The tech giants of Silicon Valley are starting to rely on crime-fighting robots for security.
- [7] Ed Grabianowski. (2017). How the Uncanny Valley Works.
- [8] Hugh Durrant-Whyte, Tim Bailey. Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms.
- [9] J.A. Battle, A. Barjau. (2008) Holonomy in mobile robots.
- [10] Jigar J. Parmar, Chirag V. Savant. (2014) Selection of Wheels in Robotics.
- [11] Ayush Pandey, Siddharth Jha and Debashish Chakravarty. (2016). Modeling and Control of an Autonomous Three Wheeled Mobile Robot with Front Steer.
- [12] Krzysztof Kozłowski, Dariusz Pazderski. (2016) Modelling and control of a 4-wheel skid steering mobile robot.
- [13] Mariane Dourado Correia, André Gustavo Scolari Conceicao. (2012). Modeling of a Three Wheeled Omnidirectional Robot Including Friction Models.

- [14] Jun Qian, Bin Zi, Daoming Wang , Yangang Ma, Dan Zhang. (2017). The Design and Development of an Omni-Directional Mobile Robot Oriented to an Intelligent Manufacturing System
- [15] Robot Platform. Wheel control theory.
- [16] Backyard Robotics. Emulating an Ackermann Steering Vehicle.
- [17] Douglas Equipment Company. <https://www.douglasequipment.com/casters-1/difference-casters-wheels/>
- [18] Alessandro Rizzo. (2018). Mobile and service robotics lecture notes.
- [19] Caltech ME 72: Engineering Design Laboratory. Analysis of a Differential Drive Vehicle.
- [20] John Piccirillo. (2015). The Art and Science of Selecting Robot Motors.
- [21] Oriental motor. <https://www.orientalmotor.com/brushless-dc-motors-gear-motors/technology/AC-brushless-brushed-motors.html>
- [22] Ben Franzluebbbers . (2013). Drag Coefficients of Inclined Hollow Cylinders:RANS versus LES.
- [23] Anaheim Automation. <https://www.anaheimautomation.com/products/brushless/brushless-motor-item.php?sID=146pt=itID=96cID=22>
- [24] Anaheim Automation. <https://www.anaheimautomation.com/products/brushless/brushless-gearmotor-item.php?sID=153pt=itID=99cID=47>
- [25] Tech Briefs. (2017). <https://www.techbriefs.com/component/content/article/tb/features/articles/2628>
- [26] Bharat Joshi. (2016). DC Motor Speed Control using Encoder Feedback.
- [27] Danielle Collins. (2017). How are encoders used for speed measurement?
- [28] Dynapar. https://www.dynapar.com/technology/encoder_basics/motor_encoders/
- [29] Anaheim Automation. <https://www.anaheimautomation.com/products/encoder/incremental-rotary-item.php?sID=352serID=3pt=itID=1054cID=422>

- [30] Robot Shop. (2018). <https://www.robotshop.com/community/tutorials/show/how-does-wheel-size-affect-my-robot>
- [31] Ron Kurtus. (2016). Coefficient of rolling friction.
- [32] Ben Redwood. (2018). The additive manufacturing process.
- [33] Leo Christodoulou.(2019). Comparison of Additive Manufacturing CNC Machining.
- [34] Alkaios Bournias Varotsis. (2018). Introduction to FDM 3D printing.
- [35] Ben Hudson. (2018). How to design parts for FDM 3D Printing.
- [36] The technology house. <https://www.tth.com/3d-printing/fdm-prototyping/>
- [37] Quality one International. <https://quality-one.com/dfm-dfa/>
- [38] ALL3DP. Top 3 Tips for Designing for 3D Printing. <https://all3dp.com/top-3-tips-designing-3d-printing/>
- [39] ALL3DP. 2019 3D Printing Technology Guide – 10 Types of 3D Printers. <https://all3dp.com/1/types-of-3d-printers-3d-printing-technology/>.
- [40] Adreco Plastic. ABS Plastic properties and application. <https://www.adrecoplastics.co.uk/abs-plastic-properties-and-application/>
- [41] Italia 3D Printing. ABS <https://www.italia3dprint.it/warping/>
- [42] ALL3DP. PLA vs ABS Filament. <https://all3dp.com/1/pla-vs-abs-filament-3d-printing/>
- [43] Sunny Sahota. (2019). The Fundamentals of Hardware Assembly Design.
- [44] Matt Gibbons. (2010). Fasteners Can Consume 20-50% of Assembly Labor.
- [45] James Low. (2019). How to design Snap-fit Joints for 3D Printing.
- [46] Dave Evans. How to Design Snap Fit Components.
- [47] Bayer Material Science. Snap-fit joints for plastic, a design guide.
- [48] Ben Hudson. (2019). How to design Interlocking Joints for fastening 3D Printed parts.

[49] Tian Ooi. (2019). 3D Printed Joints – The Basics