POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Implementation of path following and obstacle avoidance in omnidirectional platforms

Laurea Magistrale in Automation and Control Engineering - Ingegneria dell'Automazione

Author: Giulia Fasoli, Andrea Mauri

Advisor: Prof. Matteo Matteucci

Co-advisor: Simone Mentasti

Academic year: 2021-2022

1. Introduction

In the last decades, automation and robotics are gaining more importance, not only in the industrial field but also in everyday life due to the higher productivity guaranteed by robots, the huge efficiency in accomplishing tasks, and finally, their possibility to operate in conditions that can be dangerous. The aim of this thesis work is the implementation of a position controller through a PID which is integrated with the actual navigation method in an environment equipped with a motion capture system. Subsequently the development of an algorithm for obstacle avoidance that can be perfectly exploited by omnidirectional robots. Concluding with a comparison between the two adopted platforms based on their model and the experimental results.

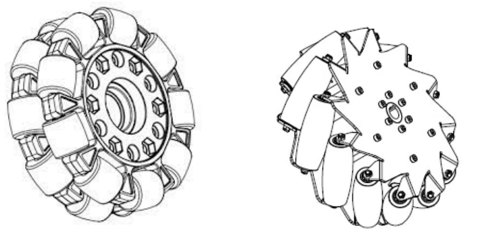
2. Mobile robots

Wheeled mobile robots are widely used to achieve robot locomotion and, although it is difficult to overcome uneven ground conditions, they are suitable for several target environments in practical applications. In general, wheeled robots are characterized by lower energy consumption and faster motion than other locomotion mechanisms (e.g., tracked vehicles or legged robots) [3].

2.1. Types of wheel

Nowadays multiple types of wheels with different properties and for various purposes are available. The wheels used in this thesis are omnidirectional wheels, characterised by unconventional design. An important feature to consider when analysing different kinds of wheels is their restriction on the motion. Each wheel has a defined number of degrees of freedom that impact the robot's range of motion. Omni wheels and Mecanum wheels have three degrees of freedom allowing motion in any direction.

Omni wheels are characterized by a combination of a main active wheel and passive freely rotating rollers in which the axes of passive rollers are orthogonal to the main wheel axis as it is shown in Figure 1a. Free rollers are employed to eliminate the non-holonomic velocity constraint. Merging the active rotation of several active wheels with the passive rotation of the rollers, it is possible to move a vehicle in any direction. Mecanum



(a) Omni wheel. (b) Mecanum wheel.

Figure 1: Omnidirecional wheel.

wheels are similar to the Omni ones but rollers are mounted with their axis at an angle of 45 degrees relative to the axis of the active wheelbase as we can see in Figure 1b.

2.2. Omnidirectional robot

Omnidirectional mobile robots can guarantee greater manoeuvrability and efficiency, indeed, they can move instantaneously in any direction from any configuration in a two-dimensional plane.
A torque is applied to each wheel by an independent actuator, so their respective direction of rotation can be arbitrarily performed. In addition, omnidirectional systems can slide perpendicularly to the torque vector, thanks to the passive rollers mounted on the edge of the wheels, allowing greater flexibility in a congested environment. On the other hand, omnidirectional robots are inefficient in terms of energy consumption because the wheels can generate opposing forces and, therefore, can wear out faster than conventional wheels [6].

3. Kinematic Model

Kinematic models are a subcategory of mechanical models describing limitations on a robot's motions, these can describe the spatial position of a rigid body, or system of bodies, neglecting the forces involved.
Starting from [2] we develop the kinematic model. The configuration of a mobile platform, that moves on a planer surface, as a rigid body can be described with three degrees of freedom of the body: two Cartesian coordinates x and y, and one orientation angle θ, i.e. *yaw* angle.
Starting from the centre of a wheel on its reference frame and then extending the analysis on the robot reference frame and, finally, the world inertial coordinate frame is possible to derive direct and inverse kinematics of the two platforms. Let's define with T_k the transformation matrix in equation that links the velocity of the centre of chassis with the angular velocity of the wheels and with R_θ the rotation matrix that links the global inertial frame to the frame of the robot chassis as a rotation of angle θ around the z axis. The inverse kinematic of a generic $n - wheel$ robot can be rewritten as:

$$\dot{q}_w = T_k R_\theta^\top \dot{q}_r \tag{1}$$

where:

- q_r represents the position of the robots' centre of mass in the global reference frame defined by the coordinates x and y and the orientation θ.
- $q_w = \begin{bmatrix} \varphi_1 & \dots & \varphi_n \end{bmatrix}$ is the configuration vector of the n wheels.

3.1. Omni wheeled robot

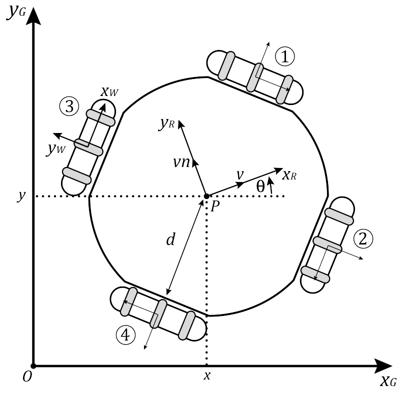


Figure 2: Omni wheeled robot

The first robot considered is designed to have four Omni wheels equally spaced every 90° in space. The inverse kinematic models become:

$$\begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \\ \varphi_4 \end{bmatrix} = -\frac{1}{r} \begin{bmatrix} -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & d \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & d \\ -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & d \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & d \end{bmatrix} \begin{bmatrix} v \\ vn \\ \dot{\theta} \end{bmatrix} \tag{2}$$

3.2. Mecanum wheeled robot

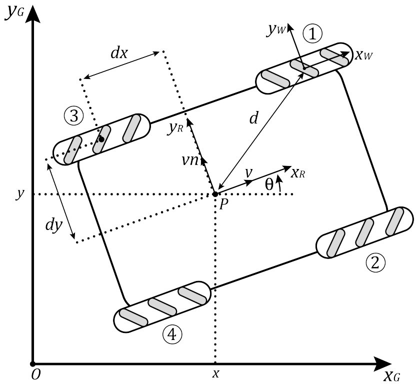


Figure 3: Mecanum wheeled robot

The second platform configuration considered has four Mecanum wheels, i.e. with 45° inclined rollers around the wheel's edge. The inverse kinematic models are expressed in Equation (3).

$$\begin{bmatrix}\varphi_1\\ \varphi_2\\ \varphi_3\\ \varphi_4\end{bmatrix} = \frac{1}{r}\begin{bmatrix}1 & -1 & -(d_x+d_y)\\ 1 & 1 & d_x+d_y\\ 1 & 1 & -(d_x+d_y)\\ 1 & -1 & d_x+d_y\end{bmatrix}\begin{bmatrix}v\\ vn\\ \dot{\theta}\end{bmatrix} \tag{3}$$

4. Black-Box Model

Black-box modelling is mostly used when the focus is on fitting the data regardless of the mathematical relationships of the model. The main features to supervise during gathering the data are the test duration and the velocity of the operating point. A long test represents better the steady state portion of the step response, while a short test represents better the transient part. The behaviour of the system changes with the speed of the chassis and because of that the parameters estimated during a test with a specific speed, might not be accurate enough if used in different velocity conditions. The data need to be collected as a trade-off between the two features mentioned above to have a good representation of the overall response and obtain a realistic model. To collect decoupled data we set different maximum speeds on the joystick and then move the robot remotely along a single axis each time. For both robots, we test different linear velocities from 0.6 m/s to 1.2 m/s and angular velocities from 2.0 rad/s to 3.5 rad/s, and for each test, we estimate the parameters of the transfer function. The final values are reported in Table 1 for the Omni wheeled robot and in Table 2 for the Mecanum wheeled robot.

axis	k_x	τ_x
x	0.807	0.207
y	0.784	0.206
$theta$	0.841	0.090

Table 1: Omni wheeled robot parameters.

axis	k_x	τ_x
x	0.724	0.175
y	0.729	0.176
$theta$	0.916	0.309

Table 2: Mecanum wheeled robot parameters.

5. Position Control

Omnidirectional platforms have three degrees of freedom, that are possible to control independently by closing three independent control loops with three different PIDs, one for each variable. Each controller receives as input the error between the set-point, i.e. the goal pose, and the measured actual position of the robot, obtained from OptiTrack™, producing as output the linear and angular velocities in the two-dimensional global reference frame. These are then converted into the robot's reference frame through a rotation matrix around the $z-axis$ and finally, according to each robot's kinematic, transformed into wheel velocities that are actuated by the four motors as specified in Section 3. Since the two platforms are probably going to operate in environments characterized by unknown obstacles, narrow places, and workstations, we decide to give more importance to percent overshoot and to steady-state error trying to reduce both to zero. In this way, the robots reach the goal in a slower but safer way, managing to arrive at the exact desired position without the risk of hitting the workstation. To avoid an aggressive proportional action that can cause a high peak of current, and therefore limit the lifetime of the actuator we set an error saturation, i.e. a maximum value that the position error can assume. Another reason to proceed at a lower speed is related to the environmental specifications. Indeed, robots work indoors where high velocities are dangerous because of the possible presence of people. For this reason, we also introduce a maximum linear saturation on the output of the PID controller equal to 1.2 m/s and saturation on the maximum angular velocity equal to 1.2 rad/s. The self-imposed specifications are listed in Table 3 .

overshoot	5%
steady-state error	0.04 m
error saturation	0.5 m
output saturation	1.2 m/s

Table 3: Control specification

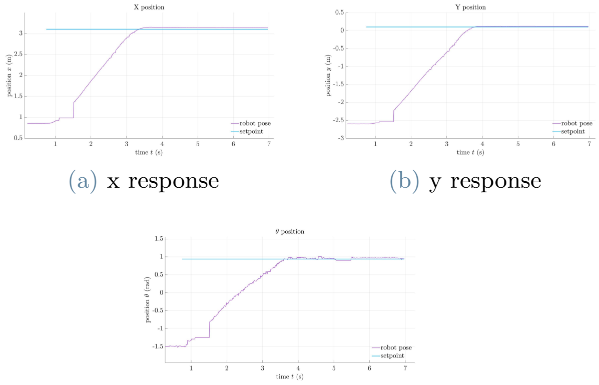
To tune the PIDs we start from Ziegler-Nichols tuning rule and then we settle the final value of the proportional, integral, and derivative gains according to a trial-and-error approach considering how a change in each gain affects the response of the system. This Ziegler-Nichols approach may generate gains that are so elevated to possibly be dangerous in a real application and, in particular, in an indoor environment like the one our robots are moving in. For this reason, as explained in Section 4, we create a black-box model of the full system, kinematic and dynamic, and we apply the tuning method in simulation through MATLAB® and Simulink® to obtain the first set of value for the gains.

5.1. Double navigation

In order to allow greater modularity during navigation for different possible applications, and to be able to use, in the best way, the in-front camera, in future works, we decide to implement the possibility to switch between two modes: the first one in which the platforms behave as a classical omnidirectional robot and another one in which the two robots are similar to differential drive ones.

5.2. Experimental results

Starting with the proportional term, we change it from a value of 0.5 to 2.5 at regular intervals of 0.5, moving from the same initial position to the same goal and analyzing the response for the three directions x, y and θ. The results of these tests for the Omni wheeled robot are shown in Figure 4 and for the Mecanum wheeled robot in Figure 5

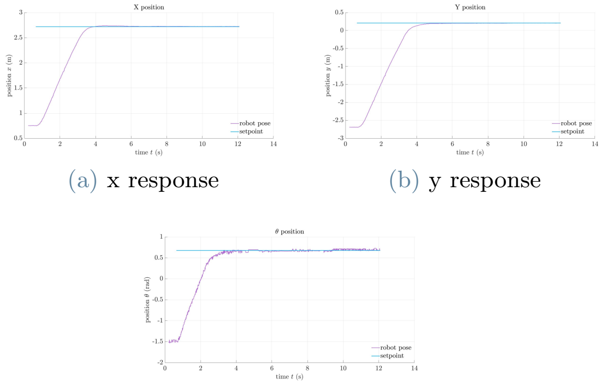


(a) x response (b) y response

(c) θ response

Figure 4: Response of the robot with proportional gain K_p=2

The response with $K_p = 2$ guarantees a response that is satisfactory both from the point of view of the percent overshoot and of the error. Since the response of the Omni wheeled robot satisfies the limitations that we impose and fulfils the specifications with just a proportional controller, we decide to not add the integral and the derivative terms.



(a) x response (b) y response

(c) θ response

Figure 5: Robot response with the implemented controller

Analyzing the obtained values, we conclude implementing a position controller composed of only the proportional term with a gain equal to 2.5. Also with this robot, as before, this type of controller is enough for our purpose, if it would be necessary to reduce the steady-state error or to obtain a faster response, an integrative term can be added even if this would increase the overshoot.

6. Spatial Horizon

To allow the robot to perform a motion that can approximate a path following, but using the waypoint following strategy with PIDs developed in Section 5 we implement an intermediate planner [5]. It can generate subgoals on the global path according to a user-determined look-ahead distance; these become the new set-points of the controller.

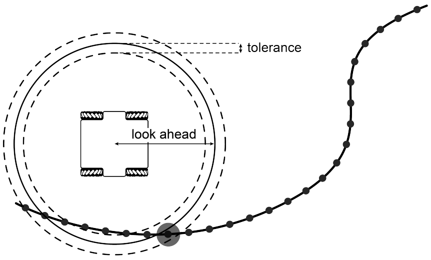
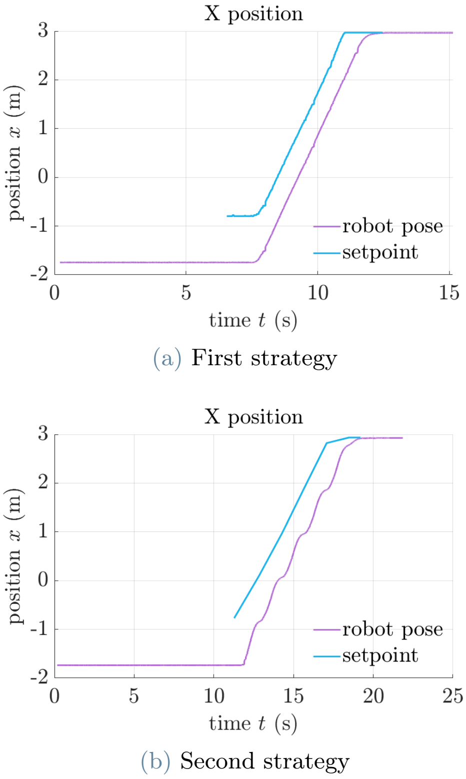


Figure 6: Spatial Horizon

The global path is composed of numerous close points. To generate the subgoal, we check all of these points evaluating the norm between the actual position of the robot and each of them. If this distance is greater than the difference between the look-ahead distance and the predetermined linear tolerance, but it's lower than the sum of the two, the farthest subgoal from the pose of the robot is selected and then published on the specific topic. From this point on we implement two possible approaches: the new subgoals are continuously published so that the robots are characterized by a smoother motion, or a new subgoal is determined only when the robots are close enough to the previous one. When the distance between the robot and the goal is lower than the look-ahead horizon, the new subgoal is set equal to the end pose and published. A comparison between the position of the robot on the x axis applying the two different strategies is depicted in Figure 7.



(a) First strategy

(b) Second strategy

Figure 7: Comparison between SH approaches

7. Obstacle Avoidance

To perform obstacle avoidance we decide to customise the VFH algorithm [1]. The behaviour of a VFH-controlled mobile robot as the response of the platforms depends on the likelihood of the presence of an a priori unknown object. Two histograms are then created: the histogram grid, in which the probability is expressed as a certainty value; and a polar histogram, which transforms the previous information into the width and height of the column of the graph. Vector Field Histogram is appropriate in situations where inaccurate sensor data are present, e.g. sensor fusion. When VFH is enabled, the subgoals from Spatial Horizon obtained, as described in Section 6, are not given directly to the PID controller but are first analyzed to check if they are on a path that can cause a collision. When the intermediate goal is defined, VFH builds a circumference with a radius equal to the distance between the actual pose and the subgoal and centred with the chassis of the robot. For each of these points, we apply Bresenham's algorithm [4] to approximate the line between the centre of the robots and the 360 points of the circumference so that we can identify all the pixels in between and calculate the cost of each line. The cost is calculated by computing the mean of the cost of all the costmap's cells that represent the likelihood of having an obstacle at that position. If the robot is stuck, i.e. no suitable points are found, we implement a procedure to recover. It consists of determining new circles with decreasing radius until the robots are able to determine a feasible path. Figure 8 shows the global path, the external circumference, and the line connecting the robot reference frame and the subgoal chosen.

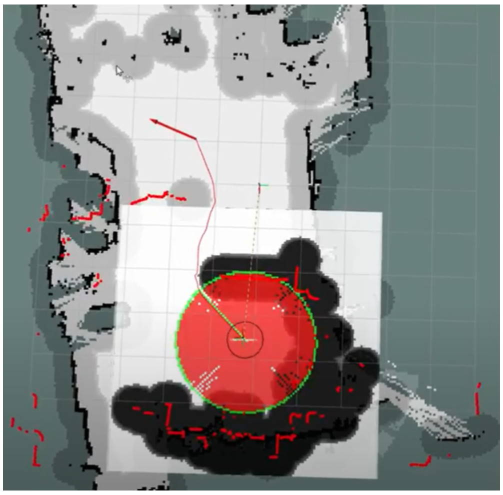


Figure 8: VFH representation on ROS

8. Conclusion

In this thesis work, we show the development of a control strategy that enables efficient navigation and obstacle avoidance of two omnidirectional platforms. In particular, we adopt a position controller, based on feedback measurement of the output from a Motion Capture system, that is able to reach the position and orientation of waypoints. The position controller is integrated with the traditional navigation approaches from ROS package move_base. Experimental results show that the intermediate planner that we implement guarantees a motion that is close to a path following, moreover the customized obstacle avoidance algorithm allows the robot to perform well and safely with simple and common obstacle shapes. In addition, all the tests are performed with both the motion possibilities that we implement: the omnidirectional one and the differential-like one. The results are similar meaning that the user can choose the mode according to its necessities with the same performances. The themes developed in this thesis allow several possible future works:

- Definition of a precise dynamic model to obtain an accurate theoretical model and improve the tuning of the controller.
- Substitution of the Motion Capture strategy (i.e., a set of external cameras that define the pose of the robot in the arena) with other localization methods.
- Implementation of a more advanced controller, e.g. MPC, to minimize specific cost functions.
- Integration of an algorithm for image recognition with the currently onboard sensors.

9. Acknowledgements

We would like to thank the AirLab staff and the stimulating atmosphere created; we came out enriched, not only as engineers. A special thanks to professors Matteucci and Mentasti for their advice and the opportunity to work on this experimental thesis together. We learned a lot.

References

[1] J. Borenstein and Y. Koren. The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7(3):278–288, 1991.

[2] Iñigo Moreno Caireta. Model predictive control for a mecanum-wheeled robot in dynamical environments. *Universitat Politècnica de Catalunya Facultat de Matemàtiques i Estadística, Universitat Politècnica de Barcelona: Barcelona, Spain*, 2019.

[3] Guy Campion and Woojin Chung. Wheeled rob 17 . wheeled robots. 2008.

[4] Kenneth I. Joy. On-line computer graphics notes: Breshenham's algorithm, 2022.

[5] Linh Kästner, Teham Buiyan, Xinlin Zhao, Zhengcheng Shen, Cornelius Marx, and Jens Lambrecht. Connecting deep-reinforcement-learning-based obstacle avoidance with conventional global planners using waypoint generators. *CoRR*, 2021.

[6] Hamid Taheri, Bing Qiao, and Nurallah Ghaeminezhad. Kinematic model of a four

mecanum wheeled mobile robot. *International journal of computer applications*, 113(3):6–9, 2015.