Optimal control based motion planning strategy for mobile robotic platforms

Matteo Facci

Department of Computer, Control, and Management Engineering
Antonio Ruberti
Control Engineering
Sapienza University of Rome

June 10, 2022

Contents

1 Introduction **3**

2 Problem formulation **4**
2.1 Recalls on optimal control 5
2.2 Proposed approach 7

3 Mathematical model of the system **8**
3.1 General characteristics and operation of a Segway device 8
3.2 Choice of the mathematical model 9

4 Implementation of the proposed approach **13**

5 Simulations **20**
5.1 Design choices 20
5.2 The `fmincon` solver 21
5.3 Simulation parameters 22
5.3.1 Initial conditions 22
5.3.2 Constraints on inputs 23
5.3.3 Time horizon and sampling time 23
5.3.4 Switching threshold and weights in functional J 23
5.3.5 Simulation results and choice of the best setup 24
5.4 Motion planning through viapoints and return to base function 30

6 Conclusions and future developments **33**

1 Introduction

Automatic motion planning is one of the most significant challenges facing autonomous mobile robotics. The goal is to be able to specify a task in a high-level language and have the robot automatically compile this specification into a set of low-level motion primitives, or feedback controllers, to accomplish the mission.
The typical task is to find a path for a wheeled robot, such as a Roomba or a self-driving car, from one configuration to another.
Motion planning evolved from this early specific problem to address a huge number of variations on the problem, allowing applications in areas such as the animation of digital characters, automatic verification of factory layouts, mapping of unexplored environments, navigation of changing environments, and so on.
New applications necessitate new considerations in the design of motion planning algorithms.
Since physical laws, uncertainties, and geometric constraints rule actions in the physical world, the design and analysis of motion planning algorithms raise a number of questions in mechanics, control theory, geometry, and computer science.
One of the most significant advantages of the usage of an autonomous mobile robot in an ordinary context is its autonomy, which allows it to scan an area for obstacles on its own. This also applies to environments that may change over time, such as a group of people entering the working area and posing obstacles to the robot. For example, instead of a complete path, the robot is given the starting and ending points and is instructed to create its own path while scanning and avoiding anything that gets in its way.
When traveling from point A to point B, it is not required to take a specific route. This means that autonomous mobile robots are more adaptable to new tasks. They can be reprogrammed more easily for new routes or layouts.
Referring to the present work, a relatively simple trajectory planning task has been considered. A mobile robotic platform, thanks to an optimal control algorithm, must reach some viapoints (with relative intermediate configurations) placed in the work environment, at the end performing a classic operation of returning to the base (the starting point). All this is done by optimizing the path according to some constraints.
A problem of this type is the same common to domestic robots of daily use such as vacuum cleaners, industrial mobile robots such as those used for warehouse storage, helping customers build a safer working environment and more cost-effective productivity, and finally, robots used for individual mobility in the context of simple transport or other autonomous tasks such as the monitoring and security of an area. In this last framework the aforementioned optimal control algorithm has been developed.

2 Problem formulation

The requirements for the formulation of an optimal control problem are:

- A model which describes the behavior of the system to be controlled;
- A cost functional J (or cost index) that takes into account the specifications assigned and the choices made in the design phase;
- Any constraints assigned on the state and on the control;

The problem can be solved by means of the Pontryagin principle, a result of optimal control theory formulated around the mid-1950s by the Russian mathematician Lev Pontryagin. The principle is based on the so-called "maximum principle": it was developed in relation to the problems of maximizing cost functionals, and consists in identifying the necessary conditions to achieve the optimal control that brings a dynamic system from an initial state to a final state, respecting any constraints for the state or the controls. When the principle is satisfied, it returns a necessary condition to demonstrate the optimality of a selected trajectory. The imposition of the necessary conditions (which as such must be respected) makes it possible to determine quite simply a solution that, satisfying them, is a candidate to be the optimal one.
In this case, a cost index J is introduced in which the control action is weighted by a piecewise state-dependent constant function, whose different constant values are assigned a priori for each subset of the state space. By doing so the value of the functional will change according to the current state. The strategy adopted is therefore strictly dependent on the state. In this way, there will no longer be a unique cost index, but different cost indexes will be taken into consideration, each of which is defined in the corresponding region of the state space, which will weigh the control in a different way depending on the region in which the system acts. The assignment of different control weights according to the region of the state space corresponds to a different amplitude of the control action in the cost functional.
As long as the system evolves in the same region of the state space, the solution of the optimal control problem is the optimal solution for the control action. When, during the evolution itself, the trajectory passes from one region of the state space to another, in the instant in which the separation boundary is reached, a change of the cost functional (switch) takes place. From that moment on, a new formulation of the optimal control problem is taken into consideration, equivalent to the previous one except for the weight of the input in the cost functional. This procedure is repeated until the final state (iteration) is reached.
The general control thus becomes a switching control, whose switching instants between one region and another are unknown a priori and become a part of the solution to the problem, as they depend exclusively on the optimal evolution of the state within each

space region. The control strategy, therefore, changes continuously, based on the current state value, assuming controls with different weights in the cost functional. The solution of the problem in synthesis will be composed of the single local solutions obtained in the respective regions of the state space in which the system evolves.

2.1 Recalls on optimal control

Let the generic dynamic system be given

$$\dot{x} = f(x(t), u(t), t) \tag{1}$$

with $x \in \mathbb{R}^n$ state vector, $u \in U \subset \mathbb{R}^p$ control vector, $x(t_0) = x_0$ known initial state.
A general expression of the cost index is:

$$J(x(t), u(t), t_f) = G[x(t_f), t_f] + \int_{t_0}^{t_f} L(x(t), u(t), t)dt \tag{2}$$

where the first term weighs the final state and the final instant, while the integral weighs the entire evolution of the state and of the control variables in the time interval $[t_0, t_f]$.
The objective of the optimal control is to determine $u^o(t)$ in $[t_0, t_f]$ in order to minimize the index J, respecting the preassigned constraints on the state and on the input, and satisfying, if present in the problem, the final condition

$$\chi[x(t_f), t_f] = 0 \tag{3}$$

with χ of class C^1 of dimension $dim(\chi) = \sigma_f \leq n + 1$, representing an admissible set for the state at the instant t_f. In order to have a solution, at least one of the elements of $\chi[x(t_f), t_f]$ has to be reachable from the initial state x_0.
The hamiltonian function is therefore defined:

$$H(x(t), u(t), \lambda_0, \lambda(t), t) = \lambda_0 L(x(t), u(t), t) + \lambda^T(t) f(x(t), u(t), t) \tag{4}$$

where L is the lagrangian in the cost index (2) of class C^2, $f(x(t), u(t), t)$ is the system dynamics of class C^2 and $\lambda \in \mathscr{R}^n$ is the costate vector.
Let introduce two other constraints in addition to the previous one (3):

$$q(x(t), u(t), t) \leq 0 \tag{5}$$

$$q_a = 0 \tag{6}$$

with q of class C^1 of dimension $dim(q) = \beta$, and q_a of dimension $dim(q_a) = \beta_a$, where the subscript a stands for "active".
The goal is to find t_f^o, $u^o \in \overline{C}^0(\mathbb{R})$, $x^o \in \overline{C}^1(\mathbb{R})$ that satisfy the above constraints and minimize J. [1]

[1] $\overline{C}^k$: function with a derivative up to the k-th order continuous almost everywhere.

Theorem 1 *Let (x^o, u^o, t_f^o) be an admissible solution such that*

$$rank\left\{ \frac{\partial \chi}{\partial (x(t_f), t_f)} \Big|^{o} \right\} = \sigma_f, \quad rank\left\{ \frac{\partial q_a}{\partial u} \Big|^{o} \right\} = \beta_a \qquad \forall t \in [t_0, t_f^o] \tag{7}$$

If (x^o, u^o, t_f^o) is a local minimum,
$\exists \lambda_0^o \geq 0,\ \lambda^o \in \overline{C}^1[t_0, t_f^o],\ \eta^o \in \overline{C}^0[t_0, t_f^o]$ *not simultaneously null in $[t_0, t_f^o]$ such that:*

$$\dot{\lambda^o} = -\frac{\partial H}{\partial x}\Big|^{oT} - \frac{\partial q}{\partial x}\Big|^{oT} \eta^o \tag{8}$$

$$0 = \frac{\partial H}{\partial u}\Big|^{oT} + \frac{\partial q}{\partial u}\Big|^{oT} \eta^o \tag{9}$$

$$\eta_j^o(t) q_j(x^o, u^o, t) = 0 \quad j = 1, ..., \beta \tag{10}$$

$$\eta_j^o(t) \geq 0, \quad j = 1, ..., \beta \tag{11}$$

$$\lambda^o(t_f^o) = -\frac{\partial \chi}{\partial x(t_f)}\Big|_{t_f^o}^{oT} \zeta, \quad \zeta \in \mathbb{R}^{\sigma_f} \tag{12}$$

$$H\Big|_{t_f^o}^{o} = -\frac{\partial \chi}{\partial t_f}\Big|_{t_f^o}^{oT} \zeta \tag{13}$$

$$H(x^o(t), w(t), \lambda_0^o, \lambda^o(t), t) \geq H(x^o(t), u^o(t), \lambda_0^o, \lambda^o(t), t), \quad \forall w \in U \tag{14}$$

The discontinuity of $\dot{\lambda^o}$ may occur in t_k where u^o has a discontinuity and $H\big|_{t_k^-}^{o} = H\big|_{t_k^+}^{o}$.
Furthermore, if $U = \mathbb{R}^p$, the minimum condition (14) reduces to $\frac{\partial H}{\partial u} = 0, \quad \forall t \in [t_0, t_f^o]$.

Theorem 1 highlights some important equations and inequalities that allow to define an optimal solution, that is the regularity conditions (7), the costate equation (8), the control equation (9), the trasversality conditions ((12)-(13)) and, conluding, the Potryagin inequality (14).
Once the constraints in Theorem 1 are respected, the solution obtained is the optimal solution for the specific cost functional $J(x(t), u(t), t_f)$. By changing this functional, the solution will also change. This means that the choice of the cost index J or, similarly, of the lagrangian $L(x(t), u(t), t)$, not only strongly affects the final result, but it represents a crucial aspect of the whole design.
A linear combination of linear or quadratic terms with constant coefficients representing the weight of each term in the sum is often used for the lagrangian L, i.e. they serve to define how much each term is important in reaching the optimal solution. This procedure is justified by the simplicity of both the problem and the calculation of the solution.

2.2 Proposed approach

Assuming a stationary problem with a normal solution ($\lambda_0 = 1$), the hamiltonian (4) can be re-expressed as:

$$H(x, u, \lambda) = L(x, u) + \lambda^T f(x, u) \tag{15}$$

In the lagrangian $L(x, u)$, a generic quadratic function of u, depending on the state, can be expressed as $u^T P(x)u$, where $P(x)$ represents the different weight of the input as a function of the state and therefore of the operating conditions. In the proposed approach, the state space is divided into N subsets I^i, such that $\cup_{i=1}^N I^i = \mathbb{R}^n$, each of which corresponds to a different strategy to be adopted.
Consequently, when $x \in I^i$, the function $P(x)$ is defined as

$$P(x) = \Pi_i \tag{16}$$

with $i = 1, ..., N$, and $\Pi_i \in \mathbb{R}^{p\times p}$ positive definite matrix whose elements are chosen at the design stage to control the cost of the input when the state changes.
While $x \in I^i$, the term $u^T \Pi_i u$ is used in the lagrangian, which can be newly expressed as $L_{\Pi_i}(x, u)$ to highlight this dependence.
Since no state-dependent weight is explicitly present in the formulation of the optimal control problem, the solution can be found according to the well-known approach that makes use of the hamiltonian

$$H_{\Pi_i}(x, u, \lambda) = L_{\Pi_i}(x, u) + \lambda^T f(x, u) \tag{17}$$

valid only while $x \in I^i$.
The optimal solution can be found by solving the necessary conditions from Theorem 1 and expressed for a stationary case with normal solution and t_f not fixed:

$$\dot{\lambda}^o = -\frac{\partial H_{\Pi_i}}{\partial x}\bigg|^{o\,T} - \frac{\partial q}{\partial x}\bigg|^{o\,T} \eta^o \tag{18}$$

$$0 = \frac{\partial H_{\Pi_i}}{\partial u}\bigg|^{o\,T} + \frac{\partial q}{\partial u}\bigg|^{o\,T} \eta^o \tag{19}$$

$$\eta_j^o q_j(x^o, u^o) = 0 \quad j = 1, ..., \beta \tag{20}$$

$$\eta_j^o \geq 0, \quad j = 1, ..., \beta \tag{21}$$

$$\lambda^o(t_f^o) = -\frac{\partial \chi}{\partial x(t_f)}\bigg|_{t_f^o}^{o\,T} \zeta, \quad \zeta \in \mathbb{R}^{\sigma_f} \tag{22}$$

$$H\big|_{t_f^o}^{o} = 0 \tag{23}$$

$$H(x^o, w, \lambda_0^o, \lambda^o) \geq H(x^o, u^o, \lambda_0^o, \lambda^o), \quad \forall w \in U \tag{24}$$

As long as $x \in I^i$ the solution obtained holds and is optimal in the i-th region.
If the solution has a trajectory that goes outside the region I^i, entering a contiguous region I^j, then a new problem must be formulated which has as an initial condition for the state the value of the boundary between regions I^i and I^j, achieved with the previous control action, using the new lagrangian $L_{\Pi_j}(x, u)$ and, subsequently, the new hamiltonian $H_{\Pi_j}(x, u, \lambda)$ under the aforementioned necessary conditions.
The final solution is obtained by concatenating all the calculated partial solutions. This solution cannot be defined as optimal because it is not calculated according to a unique cost index, but it is optimal if limited to each region of the state space.
To demonstrate the effectiveness of the proposed approach, a control strategy is developed for solving a motion planning problem of a mobile robotic platform, i.e. the calculation of the optimal trajectory from a given initial position to an objective final position. In particular, in this work, the procedure just illustrated is applied to a Segway device.
In the following chapters the mathematical model of the device will be introduced and described and the previously illustrated procedure will be implemented. Subsequently, after formulating an operative algorithm for the solution of this problem, the results will be evaluated through appropriate simulations carried out on the MATLAB calculation software.

3 Mathematical model of the system

The analysis of a particular application of the topics covered in this work has an illustrative value, intending to clarify the concepts introduced in the general context. On the other hand, the intent is to show that the approach adopted for the control of an autonomous mobile robot is not limited to this application alone, but can be effectively extended to generic dynamic systems of any type. Starting from a presentation of the general characteristics of the robotic device, the most appropriate mathematical model for solving the problem of calculating the optimal trajectory will be proposed.

3.1 General characteristics and operation of a Segway device

The Segway is a unicycle [2] electrical device used mainly for transport purpose, characterized by a platform ($65\,cm \times 63\,cm$) with two parallel wheels, able to start, stop, rotate and reverse with simple forward or backward movements of the passenger's body, and which makes turns with the help of a handlebar. It weighs around $40\,kg$ and can reach speeds of around $20\,km/h$. The operation is managed by a feedback control system, equipped with gyroscopic rotation sensors to maintain balance and make the movements in the safest and at the same time fastest way possible. The Segway in particular has five gyro sensors,

[2] Despite having two wheels, these are coaxial and independent of each other, so the model is kinematically equivalent to a unicycle.

and while it only needs three sensors to control lean forward or backward, left or right and steering left or right, the two additional sensors increase redundancy, to make the device more reliable.
All this inclination and steering information is transmitted to the vehicle's central system. The latter is composed of two identical groups of electronically controlled microprocessors, batteries, and motors that work together and share the task of driving the wheels. The microprocessors use advanced software that controls the Segway. This program monitors all stability information that comes from the gyroscopic sensors and adjusts the speed of the electric motors in response to this information.
The electric motors, which are powered by a pair of rechargeable lithium-ion (Li-Ion) batteries, can spin each of the two wheels independently at varying speeds. When the device tilts forward, the motors spin both wheels forward to prevent it from tilting too much. When the device tilts back, the motors spin both wheels backward. When the driver moves the handlebar to turn left or right, the motors spin one wheel faster than the other or, if the speed is slow, turn the wheels in the opposite direction so that the Segway rotates on itself.

3.2 Choice of the mathematical model

The vehicle is modeled as in a region I^i, with $i = 1, \dots, N$, in rigid motion on the plane, with a unit vector $\hat{v}$ representing the direction of travel.
Since this is a rigid motion, the problem has three degrees of freedom: the coordinates x_1 and x_2 indicate the position on the plane of some point of I^i (in this case the center of mass), and θ denotes the angle that the vector $\hat{v}$ forms with the X_1 axis (Figure 5).

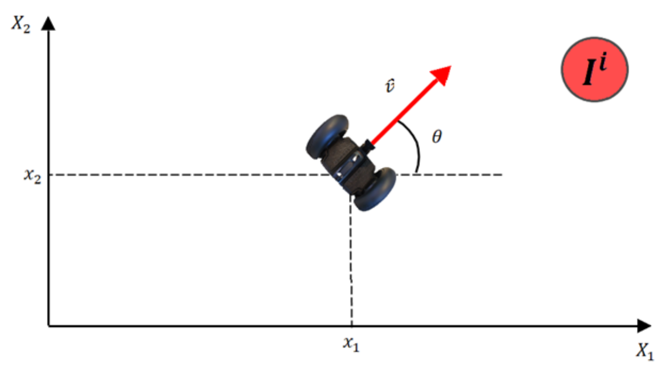


Figure 1: Vehicle model.

Relative to these coordinates, the vector $\hat{v}$ is given by

$$\hat{v} = \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} \tag{25}$$

In the problem of calculating the optimal trajectory, the Segway must reach an established final position starting from a given initial position, appropriately managing the speed and positioning when approaching the arrival point.
Thanks to its structure, the maximum speed of movement in a straight line of the Segway v_{max} corresponds to the maximum rotation speed that its two motors can supply to the wheels.
The structure in question can be described mathematically according to a model, called the unicycle model (Figure reffig:unicycle). In this case, the unicycle is characterized by two motors that act respectively on the left and right wheel of the robot, independent, non-steering, and located on the same wheelbase, so that the vehicle can rotate freely without limit of curvature, leaving unchanged the spatial position (coordinates x_1 and x_2). A motion parallel to the wheelbase is clearly impossible. From these considerations it follows that every point of the plane can be reached by the vehicle itself: the constraint imposed, called non-holonomic constraint [3], forces the speed of the robot to remain orthogonal to the axis that connects the two wheels so that each wheel is constrained to move with pure rolling motion without slipping.

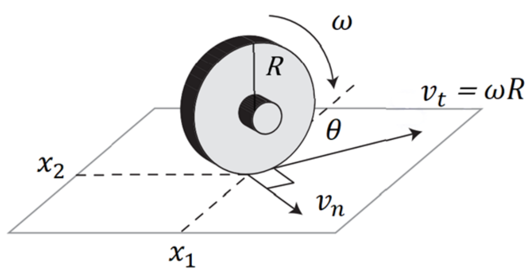


Figure 2: Unicycle model.

If there was no aforementioned constraint:

$$\begin{cases} v_{x_1} = v_t \cos\theta + v_n \cos\left(\theta + \frac{\pi}{2}\right) \\ v_{x_2} = v_t \sin\theta + v_n \sin\left(\theta + \frac{\pi}{2}\right) \end{cases} \tag{26}$$

however, since there is no slipping in the normal direction ($v_n = 0$), indicating the velocity components with $\dot{x_1}$ and $\dot{x_2}$, these will simply be:

[3]This is a limitation of the instant mobility of the device. It is not fully integrable and cannot be written in the robot configuration space (it is therefore not a space limitation).

$$\dot{x_1} = v_t \cos\theta \tag{27}$$
$$\dot{x_2} = v_t \sin\theta \tag{28}$$

while the constraint on mobility given by the absence of slipping will be:

$$\dot{x_1}\cos\Big(\theta + \frac{\pi}{2}\Big) + \dot{x_2}\sin\Big(\theta + \frac{\pi}{2}\Big) = \dot{x_1}\sin\theta - \dot{x_2}\cos\theta = 0 \tag{29}$$

As regards the control inputs to be managed, it must be borne in mind that, although in this case the available input variables are the two rotation speeds applied to the wheels, i.e. ω_R for the right wheel and ω_L for the left wheel, some considerations allow us to show that there is a one-to-one relationship that links the speed of rotation of the wheels with the speed of the midpoint of the axis of the wheels and the angular speed of the robot:

$$\begin{cases} v_R = \omega_R R \\ v_L = \omega_L R \end{cases} \tag{30}$$

from which

$$v = \frac{v_R + v_L}{2} \tag{31}$$
$$\omega = \frac{v_R - v_L}{2d} \tag{32}$$

where $2d$ is the length of the axis that connects the two wheels with radius R.
Equation (31) is obtained by applying the superposition principle, considering the contributions of the right and left wheels separately:

$$\begin{aligned} v\Big|_{v_L=0} &= \frac{v_R}{2} \\ v\Big|_{v_R=0} &= \frac{v_L}{2} \\ v &= \frac{v_R + v_L}{2} \end{aligned}$$

Equation (32), consequently, is obtained by subtracting the speed of the midpoint of the wheelbase from the speed of a wheel (obtaining the relative speed of the wheel), and dividing by the length of the wheelbase:

$$\begin{aligned} v_R - \frac{v_R + v_L}{2} = \frac{v_R - v_L}{2} &= \omega d \\ \omega &= \frac{v_R - v_L}{2d} \end{aligned}$$

It is therefore possible, as a first approximation, to consider the linear velocity v and the angular velocity ω as input variables, instead of the rotation speeds of the individual wheels.
However, it is necessary to evaluate the transformation of any (equal) maximum speed limits on the two wheels that this relationship entails (Figure 3).

Figure 3: Maximum achievable velocities.

Defining $v = u_1$, $\omega = u_2$, and $\theta = x_3$, with the conditions (27) - (28), the mathematical model used for the Segway is given by:

$$\begin{aligned} \dot{x_1} &= u_1 \cos x_3 \\ \dot{x_2} &= u_1 \sin x_3 \\ \dot{x_3} &= u_2 \end{aligned} \tag{33}$$

With the nonholonomic constraint (29) re-expresses as:

$$\frac{\dot{x_2}}{\dot{x_1}} = \tan x_3 \tag{34}$$

The state of the considered system is therefore given by $x = (x_1, x_2, x_3)^T$, so the state space will be $\mathbb{X} = \mathbb{R}^2 \times [0, 2\pi)$. It should be noted that in the case considered, the robot is bound to advance in a single direction perpendicular to the axis of the wheels, while the rotation is carried out by setting the speed of the two wheels differently since, it should be remembered, they can basically be managed independently. To introduce the control variables that can be created by the physical actuators, it is, therefore, necessary to define the appropriate sets for each of the input types chosen.

$$U_1 = [-v_{max}, v_{max}] \tag{35}$$

$$U_2 = \left[-\frac{v_{max}}{d}, \frac{v_{max}}{d} \right] \tag{36}$$

Ultimately, referring again to the model (Figure 5), the device can move only in the direction of $\hat{v}$, with speed $u_1 \hat{v}$ and $u_1 \in U_1$ (the vehicle can go back and forth within the

established limits) while, as regards the change in direction (the relative steering or the rotations in place), the angular velocity u_2 with $u_2 \in U_2$ must be managed.
However, for sake of completeness, it is still possible to introduce the generic case where the vehicle can go forward but not reverse. The reason for this choice is practical: most land and water vehicles, despite having the possibility to reverse, have a preferential direction of motion set by both locomotion (cars and ships can proceed backwards only at reduced speed) and from the driver's field of vision (visibility is reduced when reversing). Just change the definition set (35) with $U_1 = [0, v_{max} n]$.

4 Implementation of the proposed approach

The problem faced is essentially a positioning problem as already introduced in the previous chapters, the robotic device must settle in a specific target position starting from a generic position $(x_{1_i}, x_{2_i}, x_{3_i})^T$, appropriately managing the linear velocity and the angular velocity according to the distance that separates it from the point of arrival. The mathematical model defined in Section 3 lends itself well to the solution of this optimal control problem. In fact, for instance, still referring to the example in Figure 5, we want to bring the Segway, with physically reasonable velocities, to the target position $x_f = (0, 0, 0)^T$ (the center of the axes with zero orientation of the robot), optimizing the cost functional by calculating (minimizing) the optimal trajectory.

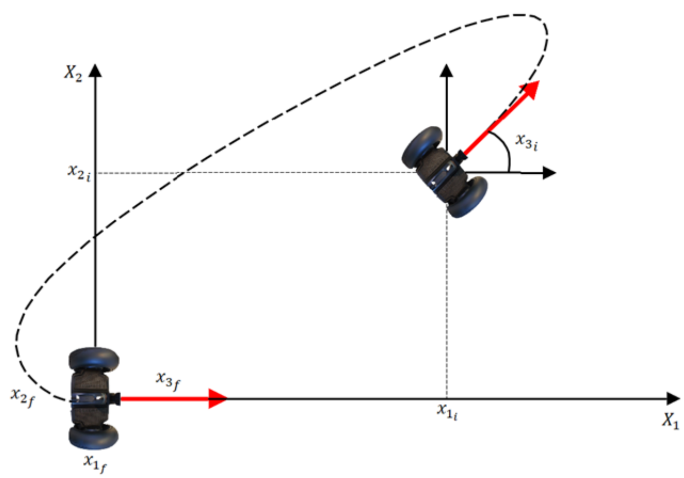


Figure 4: Target of the problem.

Different control strategies are then defined to be applied to the model according to the position of the device and the distance that separates it from the target. The problem can be set in the following way: given an initial state $x_i \in \mathbb{X}$ and a final state $x_f \in \mathbb{X}$ we want to find the trajectory π^o which minimizes the cost index J which will be introduced shortly.

$$\pi^o = argmin\Big\{\pi|x(t_0) = x_i \wedge x(t_f) = x_f \wedge$$
$$\{\forall t \in [0, t_f]x(t) \in \mathbb{X}, u_1(t) \in U_1, u_2(t) \in U_2\}\Big\}J(\pi) \tag{37}$$

Calling π^o the optimal trajectory connecting two generic points x_0 and x_1, regardless of the constraints, and J^o the cost of this trajectory:

$$J^o[x_0, x_1] = min\Big\{\pi|x(t_0) = x_0 \wedge x(t_f) = x_1\Big\}J(\pi) \tag{38}$$

$$\pi^o[x_0, x_1] = argmin\Big\{\pi|x(t_0) = x_0 \wedge x(t_f) = x_1\Big\}J(\pi) \tag{39}$$

Note that for every $t_0 < t < T$ it results that $J^o(x_0, x_1) = J^o(x_0, x(t)) + J^o(x(t), x_1)$. Therefore, the optimal trajectory π^o between the states x_i and x_f consists of a chain of connections between different states $(x_i, x_1, x_2, \ldots, x_f) \in \mathbb{X}$. The problem consequently translates into the computation of the optimal local trajectory between two single states.
In the case in question, two levels of control are defined: if the Segway's position is outside a certain threshold ξ, the linear velocity of the vehicle is controlled and optimized at the expense of angular speed; otherwise, once the threshold is exceeded, the opposite situation will occur, the angular velocity will therefore be optimized at the expense of the linear velocity. This is to allow the vehicle to approach the target position in the shortest possible time (high linear velocity) and then settle down precisely at the designated point with the required orientation (reduced linear velocity, increased angular velocity).

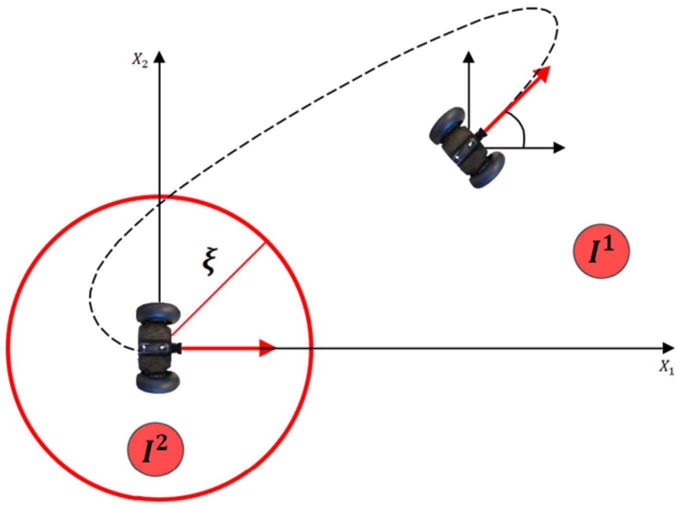


Figure 5: Graphical setting of the problem with the threshold ξ and the two regions of the state space separated by it.

The method suggested to develop this type of approach translates into a first phase, the approach phase, in which the Segway is positioned outside the edge of the circle of radius ξ (domain $x_1^2+x_2^2 \geq \xi^2$), where a greater control action on the linear velocity u_1 is required; and in a second phase, that of positioning, in which the Segway is positioned inside the edge of the circle of radius ξ (domain $x_1^2 + x_2^2 < \xi^2$), for which a greater control action on angular velocity u_2. To design the control, all this implies that, during the approach, a lower cost (or lower weight) will be assigned to the linear velocity than the angular velocity, in order to optimize the first at the expense of the second; vice versa, during the positioning of the vehicle, to optimize the angular velocity, the latter will be assigned a weight less than that assigned to the linear velocity.
So, the state space $x = (x_1, x_2, x_3)^T$, is divided into two regions:

$$I^1 = \left\{ x \in \mathbb{R}^2 \times [0, 2\pi) : x_1^2 + x_2^2 \geq \xi^2 \right\} \tag{40}$$

$$I^2 = \left\{ x \in \mathbb{R}^2 \times [0, 2\pi) : x_1^2 + x_2^2 < \xi^2 \right\} \tag{41}$$

where I^1 is the region in which the advance phase takes place, while I^2 is the region in which the positioning phase takes place. The most appropriate cost functional for this type of problem is the following:

$$\int_{t_0}^{t_f} \left[K_1\left(x_1(t) \; - x_{1f}\right)^2 + K_2\left(x_2(t) - x_{2f}\right)^2 + K_3\left(x_3(t) - x_{3f}\right)^2 + P_1(x(t))u_1(t)^2 \right. \\ \left. + P_2(x(t))u_2(t)^2\right] dt \tag{42}$$

with $K_i > 0, i = 1, 2, 3$, where the choice of the optimal trajectory can therefore be seen as the zero stabilization of the error between the current and the desired configuration, hence $\left(x_i(t) - x_{if}\right)^2$, and $P_1(x(t))$ and $P_2(x(t))$ are the weights placed respectively on the linear velocity and on the angular velocity.
If the objective coincides with the origin ($x_f = (0, 0, 0)^T$)), the integral can be simplified as:

$$\int_{t_0}^{t_f} \left[K_1\left(x_1(t)\right)^2 + K_2\left(x_2(t)\right)^2 + K_3\left(x_3(t)\right)^2 + P_1(x(t))u_1(t)^2 + P_2(x(t))u_2(t)^2\right] dt \tag{43}$$

The weights on the velocities as a function of the state $P_1(x(t))$ and $P_2(x(t))$, are defined as:

$$\begin{cases} x \in I^1 \\ P_1(x(t)) = \Pi_{11} \\ P_2(x(t)) = \Pi_{12} \\ \Pi_{11} < \Pi_{12} \end{cases} \tag{44}$$

$$\begin{cases} x \in I^2 \\ P_1(x(t)) = \Pi_{21} \\ P_2(x(t)) = \Pi_{22} \\ \Pi_{21} > \Pi_{22} \end{cases} \tag{45}$$

In this way, it is possible to optimize the two types of velocities according to whether the vehicle is in the advance or positioning phase. As a matter of simplicity, the initial conditions will be expressed as: $x_1(t_0) = x_{1,0}$, $x_2(t_0) = x_{2,0}$ and $x_3(t_0) = x_{3,0}$. Assuming non-trivial initial conditions with $x_0 \in I^1$, that is, that the device is outside the domain $x_1^2 + x_2^2 < \xi^2$, having as the first objective to reach the edge of the circle of radius ξ, the final condition is:

$$\chi[x(t_f), t_f] = x_1(t_f)^2 + x_2(t_f)^2 - \xi^2 = 0 \tag{46}$$

While regarding the limitation of resources, then the physical limits of the velocities, these can be expressed as:

$$q(u_1(t)) = \begin{pmatrix} q_1(t) \\ q_2(t) \end{pmatrix} = \begin{pmatrix} u_1(t) - U_1 \\ -u1(t) - U_1 \end{pmatrix} \leq 0 \tag{47}$$

$$q(u_2(t)) = \begin{pmatrix} q_3(t) \\ q_4(t) \end{pmatrix} = \begin{pmatrix} u_2(t) - U_2 \\ -u2(t) - U_2 \end{pmatrix} \leq 0 \tag{48}$$

where respectively the first components represent the upper bound and the second the lower bound (in case of the absence of reversing it will be sufficient to make the necessary substitutions at the lower bound). To solve the problem, the theory of optimal control formalized in Section 2.2 is applied; therefore the hamiltonian function in each region I^i is defined as:

$$\begin{aligned} &H_{\Pi_i}\left(x_1(t), x_2(t), x_3(t), u_1(t), u_2(t), \lambda_1(t), \lambda_2(t), \lambda_3(t)\right) = \\ &= K_1\left(x_1(t)\right)^2 + K_2\left(x_2(t)\right)^2 + K_3\left(x_3(t)\right)^2 + \Pi_{i1}(x(t))u_1(t)^2 + \Pi_{i2}(x(t))u_2(t)^2 + \\ &+ \lambda_1(t)\left(u_1(t)\cos x_3(t)\right) + \lambda_2(t)\left(u_1(t)\sin x_3(t)\right) + \lambda_3(t)\left(u_2(t)\right) \end{aligned} \tag{49}$$

with the necessary conditions:

$$\begin{aligned} \dot{\lambda_1} &= -\frac{\partial H_{\Pi_i}}{\partial x_1} = -2K_1(x_1(t)) \\ \dot{\lambda_2} &= -\frac{\partial H_{\Pi_i}}{\partial x_2} = -2K_2(x_2(t)) \\ \dot{\lambda_3} &= -\frac{\partial H_{\Pi_i}}{\partial x_3} = -2K_3(x_3(t)) + \lambda_1(t)(u_1(t)\sin x_3(t)) - \lambda_2(t)(u_1(t)\cos x_3(t)) \end{aligned} \tag{50}$$

$$\begin{aligned} 0 &= \frac{\partial H_{\Pi_i}}{\partial u_1} + \frac{\partial q_1}{\partial u_1}\eta_1 + \frac{\partial q_2}{\partial u_1}\eta_2 = \\ &= 2\Pi_{i1}(x(t))u_1(t) + \lambda_1(t)\cos x_3(t) + \lambda_2(t)\sin x_3(t) + \eta_1(t) - \eta_2(t) \\ 0 &= \frac{\partial H_{\Pi_i}}{\partial u_2} + \frac{\partial q_3}{\partial u_2}\eta_3 + \frac{\partial q_4}{\partial u_2}\eta_4 = \\ &= 2\Pi_{i2}(x(t))u_2(t) + \lambda_3(t) + \eta_3(t) - \eta_4(t) \end{aligned} \tag{51}$$

$$\begin{aligned} \eta_1(t)q_1\left(u_1(t)\right) &= 0 \\ \eta_2(t)q_2\left(u_1(t)\right) &= 0 \\ \eta_3(t)q_3\left(u_2(t)\right) &= 0 \\ \eta_4(t)q_4\left(u_2(t)\right) &= 0 \end{aligned} \tag{52}$$

$$
\begin{aligned}
\eta_1(t) &\geq 0 \\
\eta_2(t) &\geq 0 \\
\eta_3(t) &\geq 0 \\
\eta_4(t) &\geq 0
\end{aligned} \tag{53}
$$

$$
H_{\Pi_i}(x(t_f), u(t_f), \lambda(t_f)) = 0 \tag{54}
$$

$$
\begin{aligned}
&\lambda_1(t_f) = -\frac{\partial \chi[x(t_f), t_f]^T}{\partial x_1(t_f)}\zeta_1 = -2x_1(t_f)\zeta_1 \\
&\lambda_2(t_f) = -\frac{\partial \chi[x(t_f), t_f]^T}{\partial x_2(t_f)}\zeta_2 = -2x_2(t_f)\zeta_2 \\
&\lambda_3(t_f) = -\frac{\partial \chi[x(t_f), t_f]^T}{\partial x_3(t_f)}\zeta_3 = 0 \\
&\zeta_i \in \mathbb{R}, \quad i = 1, 2, 3
\end{aligned} \tag{55}
$$

Taking into consideration also the Pontryagin principle (14).
Let us now introduce the two functions $W_1(t)$ and $W_2(t)$ obtained from the necessary condition (51), that is the control equation:

$$
W_1(t) = -(\lambda_1(t) \cos x_3(t) + \lambda_2(t) \sin x_3(t)) \tag{56}
$$

$$
W_2(t) = -\lambda_3(t) \tag{57}
$$

Since the feasible regions, U_1 and U_2 are defined with constraints of the type $-U_i \leq u_i(t) \leq U_i$, $\forall t \in [t_0, t_f]$, $i = 1, 2$ then the control law compatible with the Pontryagin minimum principle corresponds to:

$$
u_1^i(t) = \begin{cases} -U_1 & \text{if } \frac{W_1(t)}{2\Pi_{i1}} < -U_1 \\ \frac{W_1(t)}{2\Pi_{i1}} & \text{if } -U_1 < \frac{W_1(t)}{2\Pi_{i1}} < U_1 \\ U_1 & \text{if } \frac{W_1(t)}{2\Pi_{i1}} > U_1 \end{cases} \tag{58}
$$

$$
u_2^i(t) = \begin{cases} -U_2 & \text{if } \frac{W_2(t)}{2\Pi_{i2}} < -U_2 \\ \frac{W_2(t)}{2\Pi_{i2}} & \text{if } -U_2 < \frac{W_2(t)}{2\Pi_{i2}} < U_2 \\ U_2 & \text{if } \frac{W_2(t)}{2\Pi_{i2}} > U_2 \end{cases} \tag{59}
$$

Calculating the cost functional, denoted by $J^i\left(t_f^i, x_1^i(t), x_2^i(t), x_3^i(t), u_1^i(t), u_2^i(t)\right)$ the solution obtained for the time interval $\left[t_0, t_f^i\right]$, this is also the optimal solution for $x(t) \in I^i$.

Optimal control can be calculated and applied as long as the state belongs to the given region I^i; crossing the border of the region, it is necessary to update all the parameters, in particular the weights Π_i, and calculate a new optimal control for the new region of the state space I^j.
Since the Segway is initially in the region I^1, then $x_0 \in I^1$, and $x(t) \in I^1$, $\forall t \in [t_0, t_f^1]$, it will be necessary to integrate by setting as index $i = 1$, obtaining as an optimal solution $J^1\left(t_f^1, x_1^1(t), x_2^1(t), x_3^1(t), u_1^1(t), u_2^1(t)\right)$ and, according to the final condition, $x_1(t_f^1)^2 + x_2(t_f^1)^2 = \xi^2$. In practice, at instant t_f^1, the device will be exactly on the edge of the circle with radius ξ and the robot's approach phase can be considered completed. Indicating for simplicity $t_f^1 = t_1$, it will therefore result that $x(t_1^-) \in I^1$ while $x(t_1^+) \in I^2$, and it is for this reason that the instant of time considered is defined as the instant of switching, in fact it is the instant in which the Segway reaches the threshold and in which the new formulation of the problem must be developed.
The new formulation differs from the previous one only in some fundamental aspects. In particular, the initial conditions will change, for which the new instant $t_0 = t_1$ and $x(t_0) = x(t_1)$ will be defined, and the final condition, which always in case of $x_f = (0, 0, 0)^T$) can be rewritten as: $\chi[x(t_f), t_f] = x_1(t_f)^2 + x_2(t_f)^2 + x_3(t_f)^2 = 0$ as the device must simply be in the origin with a constraint also concerning the orientation (zero in this case). By changing the final condition, the necessary condition (55) will change as well, which will be:

$$\begin{aligned}
\lambda_1(t_f) &= -\frac{\partial \chi[x(t_f), t_f]^T}{\partial x_1(t_f)}\zeta_1 = -2x_1(t_f)\zeta_1 \\
\lambda_2(t_f) &= -\frac{\partial \chi[x(t_f), t_f]^T}{\partial x_2(t_f)}\zeta_2 = -2x_2(t_f)\zeta_2 \\
\lambda_3(t_f) &= -\frac{\partial \chi[x(t_f), t_f]^T}{\partial x_3(t_f)}\zeta_3 = -2x_3(t_f)\zeta_3 \\
\zeta_i &\in \mathbb{R}, \quad i = 1, 2, 3
\end{aligned} \tag{60}$$

All remaining considerations will remain unchanged from the previous phase.
The Segway will now be in the positioning phase in the I^2 region, with $x(t_1) = x_1 \in I^2$, and $x(t) \in I^2$, $\forall t \in [t_1, t_f^2]$, therefore it will be necessary to integrate by setting as index $i = 2$, obtaining as optimal solution $J^2\left(t_f^2, x_1^2(t), x_2^2(t), x_3^2(t), u_1^2(t), u_2^2(t)\right)$ and, according to the final condition, the device will settle in the position $x_f = (0, 0, 0)^T$.
In this way, the final time for covering the whole trajectory is $T = t_f^2$, optimizing the control inputs according to the region of the state in which it is operated, and minimizing the total cost functional J^o, which is obtained by the concatenation of the cost indexes calculated and minimized for the solution of the two subproblems in the regions I^1 and I^2:

$$J^o\left(T, x_1^o(t), x_2^o(t), x_3^o(t), u_1^o(t), u_2^o(t)\right) = \\ = J^1\left(t_f^1, x_1^1(t), x_2^1(t), x_3^1(t), u_1^1(t), u_2^1(t)\right) + J^2\left(t_f^2, x_1^2(t), x_2^2(t), x_3^2(t), u_1^2(t), u_2^2(t)\right) \tag{61}$$

5 Simulations

The structure according to which the problem object of this work was set up, in the light of the theoretical considerations developed in the previous sections, suggests two types of approaches for the resolution of the same: a first approach, typical for determining the optimal path for mobile devices, which consists in solving the system of differential equations composed of the hamiltonian (49) and the necessary conditions; and a second approach, which takes up the initial conception of an optimal control problem, consisting in the calculation and minimization of the cost index (42).
Not having a finite time interval available for solving the problem, as both the final instant and the switching instant are unknown a priori, it is more appropriate to opt for the second proposed approach. This preference, through appropriate choices in the simulator, implies the possibility of arriving at the result while at the same time obtaining the key instants object of the problem, thus reducing the complexity of the final calculation, increasing precision, and avoiding the non-trivial complication of having to operate on an infinite time interval.

5.1 Design choices

The structure of the model is one of the main requirements. The simulator uses this model to make predictions on the future behavior of the system, predicting all subsequent states. In this way, the optimal control input can be determined and applied accordingly. However, in order to use the mathematical model (33) in the simulator, it is necessary to obtain the discretized equivalent, that is to convert the continuous time model into a sampled data model with a sampling step T_s.
It is assumed that the generic input $u(t)$ is constant between one sampling instant and the next: $u(\tau) = cost = u(iT_s)$, with $iT_s \leq \tau \leq (i+1)T_s$.
For N samples over a total time T the discretized model will be:

$$\begin{aligned} &x_1(i+1) = x_1(i) + T_s(u_1(i)\cos x_3(i)) \\ &x_2(i+1) = x_2(i) + T_s(u_1(i)\sin x_3(i)) \\ &x_3(i+1) = x_3(i) + T_s(u_2(i)) \\ &x_1(1) = x_{1,0}, \quad x_2(1) = x_{2,0}, \quad x_3(1) = x_{3,0} \end{aligned} \tag{62}$$

with $i = 1, \ldots, N$ and $T_s = \frac{T}{N}$.

The discretization process is not limited only to the model, but must be applied to all the theoretical considerations formulated for continuous time, in particular this will also involve the cost functional J (42)

$$\begin{aligned} J &= \int_{t_0}^{t_f} L(x(t), u(t))dt = \sum_{i=1}^{N} \int_{iT_s}^{(i+1)T_s} L\left(x\left(iT_s\right), u\left(iT_s\right)\right) dt \\ &= \sum_{i=1}^{N} L\left(x\left(iT_s\right), u\left(iT_s\right)\right) \cdot \left[(i+1)T_s - iT_s\right] = T_s \cdot \sum_{i=1}^{N} L\left(x\left(iT_s\right), u\left(iT_s\right)\right) \end{aligned} \tag{63}$$

for which the new discretized functional will be expressed in the form:

$$\begin{aligned} J = T_s \cdot \sum_{i=1}^{N} K_1(x_1(i) - x_{1f})^2 + K_2(x_2(i) - x_{2f})^2 + K_3(x_3(i) - x_{3f})^2 \\ + P_1 u_1(i)^2 + P_2 u_2(i)^2 \end{aligned} \tag{64}$$

5.2 The `fmincon` solver

There are several optimization methods available, and solvers are normally designed to find the desired solution in an optimization problem using one or a combination of these methods.

In this work, the `fmincon` function, contained in the MATLAB Optimization Toolbox, will be used, which allows to find a local minimum of a constrained nonlinear problem such as:

$$\begin{cases} \min f(x) \\ c(x) \leq 0 \\ ceq(x) = 0 \\ A(x) \leq b \\ Aeq(x) = beq \\ lb \leq x \leq ub \end{cases} \tag{65}$$

where x are the control variables, which can be both scalars and vectors, and $f(x)$ is the objective function that must be minimized by returning a scalar. The lower and upper bounds of x are lb and ub, respectively. $A(x)$ and $Aeq(x)$ are the linear constraints bounded by vectors b and beq respectively, while $c(x)$ and $ceq(x)$ are functions representing nonlinear constraints, which return vectors. The simplest form of the command is: `x=fmincon('f',x0,A,b,Aeq,beq,lb,ub,'nonlcon')`, where `f` is a string that contains the name of the function to be minimized, the vector `x0` is a starting point (solution estimate) that must be provided by the user, while `nonlcon` is a string that contains the name of the function that describes the nonlinear equality and inequality constraints.

By adapting the structure of the function `fmincon` to the problem in question, also in this case particular precautions must be adopted: having to minimize the discrete cost functional, a function of both the state variables x and the input variables u, it will be necessary to reformulate the problem according only to the control variables u which will have to be optimized.
Based on the choices made during the design phase of the mathematical model, it is clear that the only constraints concerning the input variables u_1 and u_2 are the lower and upper bounds so that in the new formulation of the constrained nonlinear problem it will be sufficient to simply set:

$$\begin{cases} \min\ J(u_i) \\ lb_i \leq u_i \leq ub_i \end{cases} \tag{66}$$

with $i = 1, 2$.

5.3 Simulation parameters

This section will present how parameters can be chosen to ensure reasonable results and to ensure that the solver executes the given commands without errors. The vehicle, as explained abundantly in the theoretical section, can start from any position with any orientation and is required to reach the reference point or objective located, for this purpose, in the origin or the center of the simulation area. The optimization process is not always simple and some choices can be made in favor of one parameter and at the expense of another, parameters which can be, for example, simulation times or the overall error.

5.3.1 Initial conditions

To guarantee the reliability of the result and the optimality of the trajectory, the experiments carried out foresee an initial position not too far from the objective (however, the possibility of considering a high initial distance from the objective is not excluded), for the following reasons: first of all because it is essentially a problem of positioning (more attention is paid to how the vehicle approaches the target position with respect to the path taken); secondly to allow the optimizer to converge much more quickly, reducing the number of computations (with consequent reduction of the probability of error), thus ensuring the feasibility of the calculated trajectory, reaching the goal or arriving asymptotically very close to it.
For these reasons we will consider an arbitrary starting position of the type:

$$x(1) = \begin{pmatrix} 9 \\ 11 \\ 0 \end{pmatrix} \tag{67}$$

5.3.2 Constraints on inputs

As for the physical parameters of the device to which the solver will refer, these are the lower and upper bounds of the control variables, therefore of the linear and angular velocities.
As regards the lower bound of linear velocity, in the present work the forward motion is preferred to the reverse motion, in the light of the considerations made previously at the end of the Section (3.2), however, the complete case with forward and reverse motion together is considered for the choice of the ideal setup.

$$\begin{aligned} lb &= \begin{bmatrix} -5.5556\, m/s \quad \text{or} \quad 0 \\ -17.0941\, rad/s \end{bmatrix} \\ ub &= \begin{bmatrix} 5.5556\, m/s \\ 17.0941\, rad/s \end{bmatrix} \end{aligned} \tag{68}$$

5.3.3 Time horizon and sampling time

Another important aspect to consider is the time horizon considered, that is the total duration of the experiment and the time that the user sets for the resolution of the problem. This implies that as the length of the horizon increases, the total elapsed time of the entire process also increases. In the case in question and, in general, for all models of wheeled vehicles (models that consider the orientation), the optimization of this parameter significantly affects the results obtained, as the lengths of the time horizon can influence how vehicles travel. It is therefore advisable to keep the parameter as small as possible for computational reasons, but large enough for the feasibility of the optimization problem and the achievement of the goal. Therefore, the smallest possible time horizon will be set for each simulation among those feasible, which may vary according to the simulation setup considered.
The choice of sampling time T_s is crucial as well in the process, to make simulation results more or less realistic. The goal is to respect the system dynamics used and make the vehicle's trajectory as regular as possible and physically acceptable. Considering the characteristics of the model and the technical specifications of the vehicle, an acceptable solution, below which there is a risk of compromising the feasibility of the trajectory and its regularity, is to set $T_s = 0.5\ s$.

5.3.4 Switching threshold and weights in functional J

One of the problems examined in this work is the best choice of threshold values ξ and weights P_1, P_2, and K ($K_1 = K_2 = K_3$). The variation of these parameters involves, with the same initial conditions and sampling time, a different resolution of the problem, therefore a different level of optimization, a different control effort, different times, errors, and final trajectories. The following table shows the threshold and weight parameters that will be set in subsequent simulations, to be able to choose the best configuration for the optimization problem by means of judging criteria. For the switching threshold ξ, the examined values are (in meters): $2, 4, 6, 8$. The weights on the P_i controls will assume, depending on the range to which they belong, values between $1, 10, 100, 1000$. The weights K, being placed on the error between the current position and the desired one, will have an always greater value (at least ten times) with respect to the weights on the controls, therefore taking as reference the highest value of the weights P_i for each configuration, the weights on the error will assume values between $100, 1000, 10000$. In total 12 simulations will be carried out, each with a different type of configuration, therefore with different values for the threshold and the weights, as shown below.

THRESHOLD	CONFIGURATION	INTERVAL	P_1	P_2	K	SIMULATION
$2\ m$	1	I_1	1	10	100	1
		I_2	10	1		
	2	I_1	1	100	1000	2
		I_2	100	1		
	3	I_1	1	1000	10000	3
		I_2	1000	1		
$4\ m$	1	I_1	1	10	100	4
		I_2	10	1		
	2	I_1	1	100	1000	5
		I_2	100	1		
	3	I_1	1	1000	10000	6
		I_2	1000	1		
$6\ m$	1	I_1	1	10	100	7
		I_2	10	1		
	2	I_1	1	100	1000	8
		I_2	100	1		
	3	I_1	1	1000	10000	9
		I_2	1000	1		
$8\ m$	1	I_1	1	10	100	10
		I_2	10	1		
	2	I_1	1	100	1000	11
		I_2	100	1		
	3	I_1	1	1000	10000	12
		I_2	1000	1		

Table 1: Threshold values and weights for simulations.

5.3.5 Simulation results and choice of the best setup

In light of the results obtained from the 12 simulations in this sort of trial and error method, the problem now falls on the choice of the best configuration, that is the best

combination between the threshold value and the weights in the control problem. It is necessary to specify, however, that all the data and the trajectories resulting from the tests are optimal solutions to the problem, since as regards the final total error and as regards the total time taken to reach the objective, the results are all physically acceptable and, from best to worst case, in simulation these assume values between $[0.1281, 0.4902]$ for the final position error, and between $[2.952475\ s, 3.295152\ s]$ for the total time. Therefore, the choice of the best setup would fall on an evaluation of the various possible combinations, based on a range of values limited to a few tenths both in terms of the final error and in terms of time.
Once the simulation parameters have been set, each solution obtained is the best for that type of setup, but this may not necessarily be the best when compared with the results of the other simulations. It is, therefore, necessary to evaluate, through judgment criteria, which configuration, among those available, can guarantee results that are closest to the ideal solution. In the case in question, the ideal solution provides, for example, that among all the optimal solutions there is one that has a minimal final error and that at the same time reaches the goal in the shortest time. This combination is obviously not possible since, as will be seen in the overall results, a higher total time often corresponds to reduced total errors. This is because, considering for example the parameters of the threshold, low values are linked to a very reduced space for maneuver after the problem has been switched, on the other hand, considering the values of the functional weights, higher weights imply greater control of one variable with respect to another, with even considerable differences on the final result and on the trajectory after the switching instant. The best choice is therefore to select, from all the available setups, the one that best balances the final error and the total time.
For this purpose, two initial rankings have been drawn up, both in ascending order of values, the first for the final total error, calculated as:
$e_{tot} = \sqrt{(e_{x_1})^2 + (e_{x_2})^2 + (e_{x_3})^2}$, and the second for the total time spent. For both rankings, each setup was assigned a score from 1 to 12 based on positioning (the lower the score, the better the solution).
With the same values, the solutions with a higher threshold parameter were preferred (greater space for maneuver after the problem was switched).

THRESHOLD	CONFIGURATION	TOTAL ERROR	POSITION RANKING
2 m	1	0.128100	1
2 m	2	0.137000	2
4 m	1	0.204500	3
6 m	2	0.211000	4
4 m	2	0.216400	5
6 m	3	0.373800	6
8 m	2	0.375400	7
8 m	1	0.422000	8
8 m	3	0.431500	9
6 m	1	0.439300	10
4 m	3	0.490200	11
2 m	3	0.490200	12

Table 2: Ranking based on total error.

THRESHOLD	CONFIGURATION	TOTAL TIME	TIME RANKING
8 m	2	2.952475	1
6 m	3	2.954746	2
8 m	1	2.970222	3
8 m	3	2.976055	4
6 m	1	2.976979	5
4 m	3	2.986207	6
2 m	3	2.986207	7
2 m	1	3.063850	8
2 m	2	3.112999	9
6 m	2	3.225610	10
4 m	1	3.273598	11
4 m	2	3.295152	12

Table 3: Ranking based on total time.

By adding the scores obtained in the previous rankings, a further final ranking was drawn up, according to which the best setup is the one with the lowest overall score (best combination of time and total error). In case of same score, for the sorting criterion, in this case, the setting with the lowest total error was preferred.

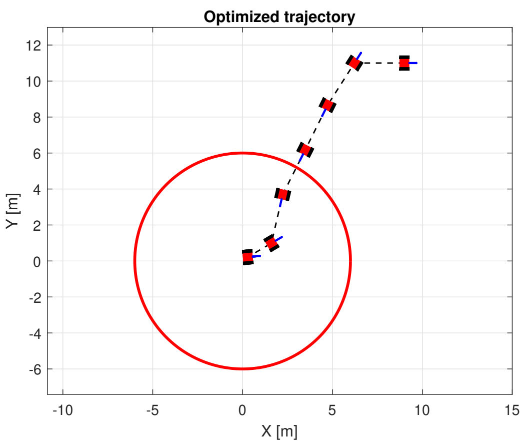
THRESHOLD	CONFIGURATION	TOTAL ERROR	TOTAL TIME	POSITION RANKING	TIME RANKING	TOTAL POINTS	FINAL RANKING
$6\ m$	3	0.373800	2.954746	6	2	8	1
$8\ m$	2	0.375400	2.952475	7	1	8	2
$2\ m$	1	0.128100	3.063850	1	8	9	3
$2\ m$	2	0.137000	3.112999	2	9	11	4
$8\ m$	1	0.422000	2.970222	8	3	11	5
$8\ m$	3	0.431500	2.976055	9	4	13	6
$4\ m$	1	0.204500	3.273598	3	11	14	7
$6\ m$	2	0.211000	3.225610	4	10	14	8
$6\ m$	1	0.439300	2.976979	10	5	15	9
$4\ m$	2	0.216400	3.295152	5	12	17	10
$4\ m$	3	0.490200	2.986207	11	6	17	11
$2\ m$	3	0.490200	2.986207	12	7	19	12

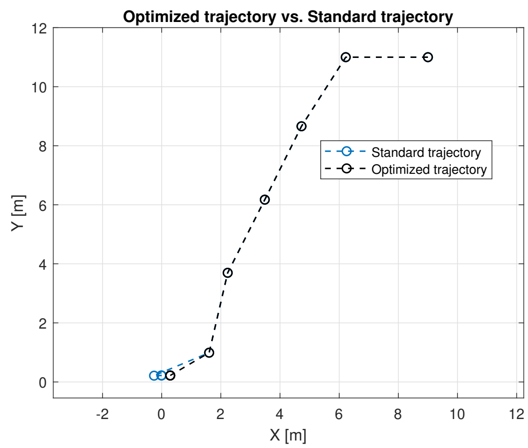
Table 4: Final ranking.

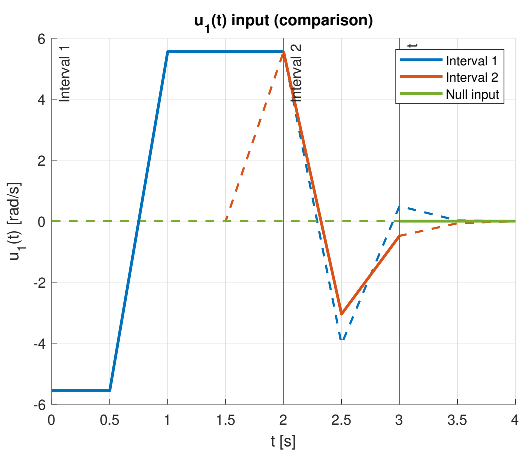
In the light of all the considerations made, the best choice for the setup parameters is the one that provides a threshold value of $\xi = 6\ m$, therefore a sufficiently large threshold for the switching of the problem and the consequent adjustment maneuvers of the vehicle; and as regards the functional weights, the configuration with the highest values ($P_i = 1\ , 1000$ and $K = 10000$), therefore with higher control over the input variables based on the region to which they belong.
Following are the results of the simulation of the best setup.

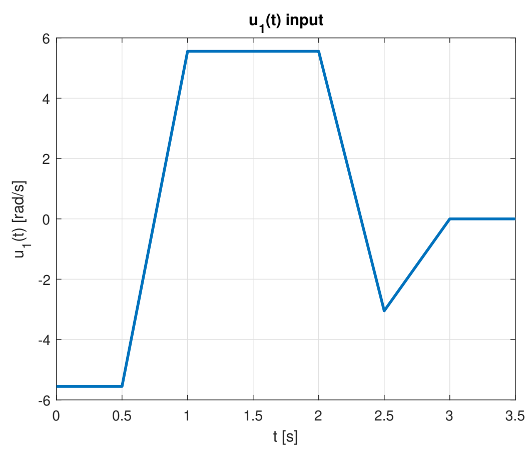
Initial abscissa x_{1i}	$9\ m$
Initial ordinate x_{2i}	$11\ m$
Initial orientation x_{3i}	$0°$
Final abscissa x_{1f}	$0\ m$
Final ordinate x_{2f}	$0\ m$
Final orientation x_{3f}	$0°$
Weights P_i	$1\ , 1000$
Weights K	10000
Threshold ξ	$6\ m$
Weights P_i	$1\ , 1000$
Weights K	10000
Switching instant	$2\ s$
Time of exceeding the threshold	$1.697064\ s$
Reaction time	$0.302936\ s$
Distance traveled between threshold and switch	$1.682980\ m$
Total time taken to reach the goal	$2.954746\ s$
Experiment duration	$3.000000\ s$

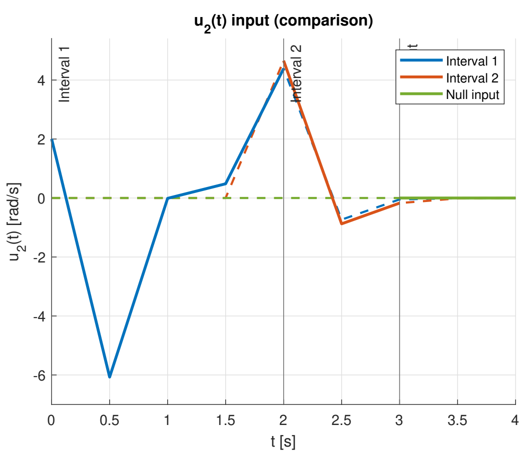
Table 5: Results of the simulation of the best setup.

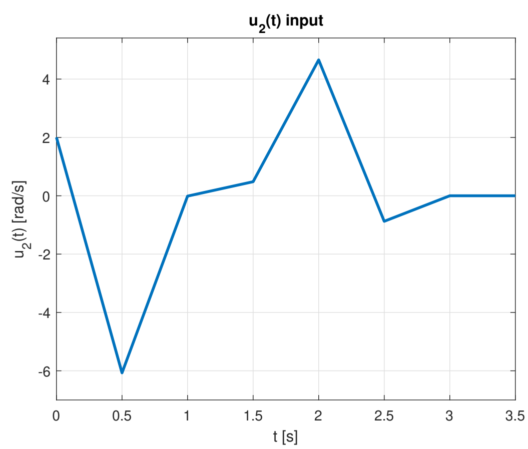

Optimized trajectory
Y [m]
X [m]

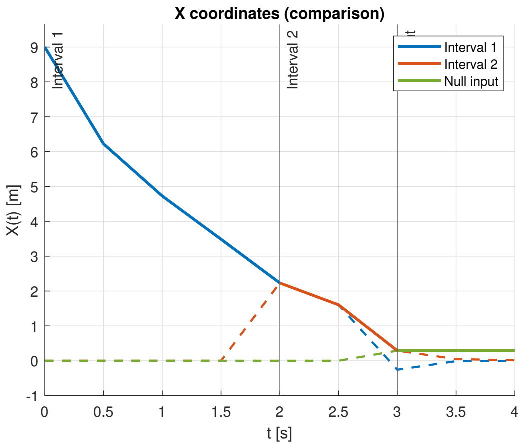

Optimized trajectory vs. Standard trajectory
Standard trajectory
Optimized trajectory
Y [m]
X [m]

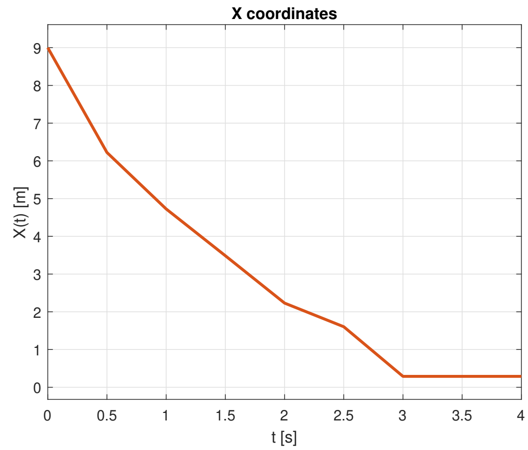

u_1(t) input (comparison)
Interval 1
Interval 2
Interval 1
Interval 2
Null input
u_1(t) [rad/s]
t [s]

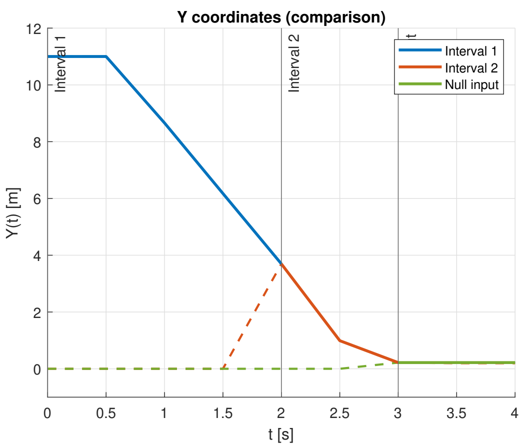

u_1(t) input
u_1(t) [rad/s]
t [s]

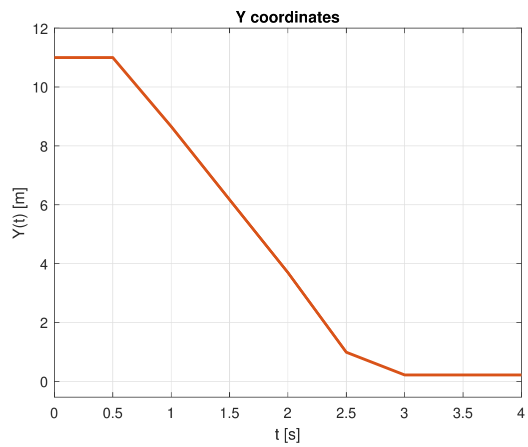

u_2(t) input (comparison)
Interval 1
Interval 2
Interval 1
Interval 2
Null input
u_2(t) [rad/s]
t [s]

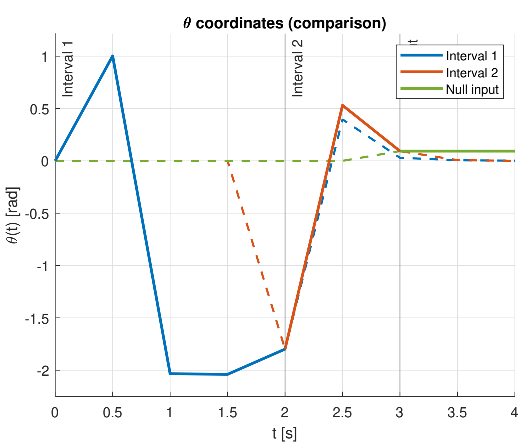

u_2(t) input
u_2(t) [rad/s]
t [s]

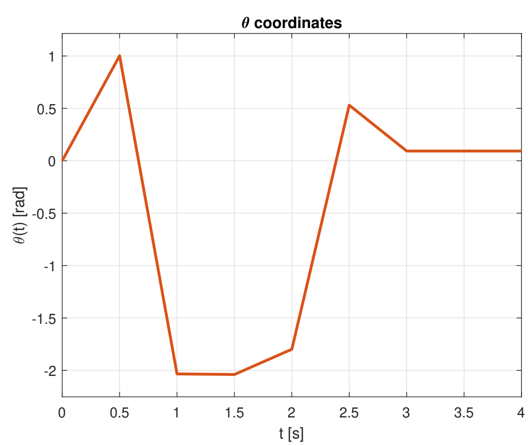

X coordinates (comparison)
Interval 1
Interval 2
Null input
X(t) [m]
t [s]


X coordinates
X(t) [m]
t [s]


Y coordinates (comparison)
Interval 1
Interval 2
Null input
Y(t) [m]
t [s]


Y coordinates
Y(t) [m]
t [s]


θ coordinates (comparison)
Interval 1
Interval 2
Null input
θ(t) [rad]
t [s]

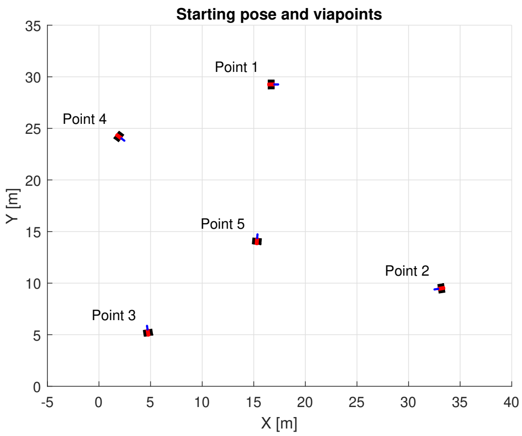

θ coordinates
θ(t) [rad]
t [s]

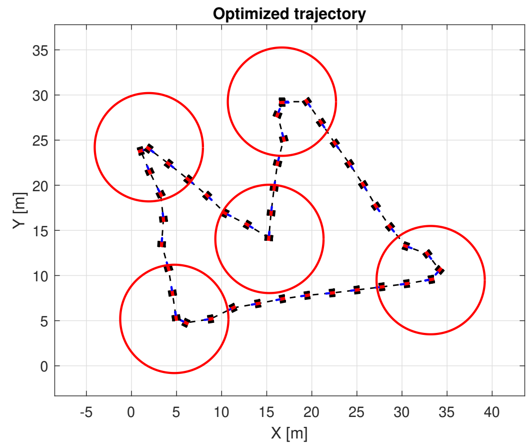
5.4 Motion planning through viapoints and return to base function

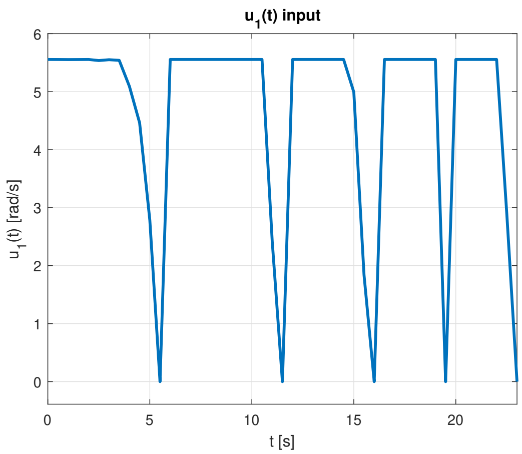
Once the best setup has been chosen to optimize the performance of the device with the optimal control algorithm abundantly described so far, this can be exploited for a more realistic case such as that of calculating the trajectory passing through several viapoints and finally returning to the base of departure, a feature implemented in more and more mobile robotic devices nowadays.
The logic of this general algorithm is to allow the user to choose the initial position (which will eventually also be the final one thanks to the return to base function) and in addition to choose other passage and positioning points that the robot will have to reach during the task.
The optimal control algorithm mentioned so far is then used for each sequence that connects each viapoint to the next. In this case, to make the trajectory feasible with reliable behavior and simultaneously improve both the positioning error and the final time as much as possible, for each sequence the algorithm is applied more than once, increasing the total time of the experiment (and consequently the total number of available samples) at each iteration. Therefore, given a certain number of optimal solutions, each with a corresponding total time taken and positioning error, these are compared with the same ranking criterion described in Section 5.3.5.
At the end of the general algorithm, by concatenating the best optimal solutions of each sequence, we obtain a single solution for the motion planning problem.

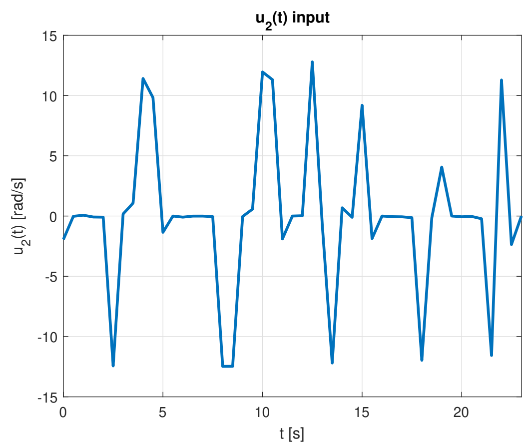
Algorithm 1 Motion planning algorithm

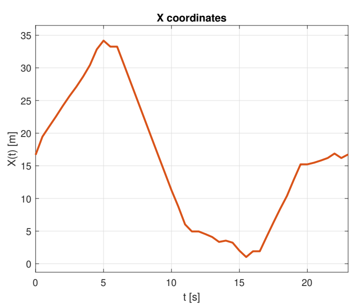
```
maxPoses ← int ≠ 0
i ← 1
while i ≤ maxPoses do
    viaPoint(i) ← pickPositionAlgorithm
for sequence ← 1 to maxPoses −1 do
    x_i ← viaPoint(sequence)
    x_f ← viaPoint(sequence+1)
    attempt ← 1
    totExperiments ← int ≥ 1
    totTimeInput ← 0
    timeStep ← int ≥ 0
    while attempt ≤ totExperiments do
        time ← totTimeInput+timeStep
        evaluationMatrix(attempt) ← optimalControlAlgorithm
        attempt ← attempt+1
    optimalSubsequenceResult(sequence)← overallRankingAlgorithm
```

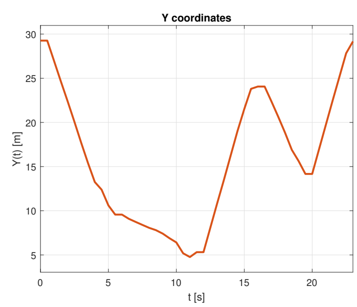

Starting pose and viapoints
Point 1
Point 4
Point 5
Point 2
Point 3
Y [m]
X [m]

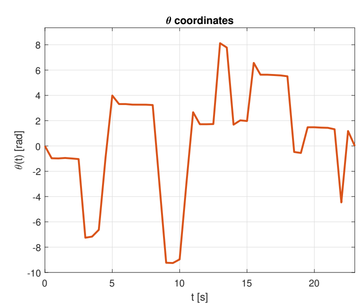

Optimized trajectory
Y [m]
X [m]


u_1(t) input
u_1(t) [rad/s]
t [s]


u_2(t) input
u_2(t) [rad/s]
t [s]


X coordinates
X(t) [m]
t [s]


Y coordinates
Y(t) [m]
t [s]


θ coordinates
θ(t) [rad]
t [s]

Time [s]	Interval	x_1 [m]	x_2 [m]	x_3 [rad]	x_3 [°]	u_1	u_2
0	1	16.682	29.2587	0	0	5.5554	-1.9466
0.5	1	19.4597	29.2587	-0.97328	304.2351	5.5547	-0.022919
1	1	21.0222	26.9625	-0.98474	303.5785	5.5535	0.070182
1.5	1	22.558	24.6491	-0.94965	305.5891	5.5543	-0.080364
2	1	24.1742	22.3907	-0.98983	303.2868	5.5555	-0.092754
2.5	1	25.6987	20.0687	-1.0362	300.6296	5.5369	-12.4362
3	1	27.1092	17.6865	-7.2543	304.36	5.5514	0.17558
3.5	1	28.6758	15.3952	-7.1665	309.39	5.5405	1.074
4	2	30.4338	13.2542	-6.6295	340.1577	5.0913	11.4122
4.5	2	32.8283	12.3901	-0.92339	307.0937	4.4622	9.824
5	2	34.1739	10.6105	3.9886	228.5303	2.7851	-1.351
5.5	3	33.2517	9.567	3.3131	189.826	0	0
6	1	33.2517	9.567	3.3131	189.826	5.5556	-0.090978
6.5	1	30.5147	9.093	3.2676	187.2197	5.5556	-0.0061264
7	1	27.759	8.7439	3.2645	187.0442	5.5556	-0.0041857
7.5	1	25.0021	8.4032	3.2624	186.9242	5.5556	-0.051732
8	1	22.2446	8.0684	3.2366	185.4422	5.5556	-12.4743
8.5	1	19.4794	7.8049	-3.0006	188.0794	5.5556	-12.468
9	1	16.7292	7.4145	-9.2346	190.8975	5.5556	-0.040073
9.5	1	14.0015	6.8894	-9.2546	189.7495	5.5556	0.57744
10	1	11.2638	6.419	-8.9659	206.2918	5.5556	11.9555
10.5	2	8.7734	5.1886	-2.9882	188.7915	5.5556	11.3132
11	2	6.0283	4.764	2.6684	152.8909	2.422	-1.8992
11.5	3	4.9503	5.3159	1.7188	98.4822	0	0
12	1	4.9503	5.3159	1.7188	98.4822	5.5556	0.026101
12.5	1	4.5406	8.0632	1.7319	99.23	5.5556	12.792
13	1	4.095	10.8051	8.1279	105.6943	5.5556	-0.69183
13.5	1	3.3436	13.4793	7.782	85.875	5.5556	-12.1961
14	1	3.5434	16.2499	1.6839	96.4824	5.5556	0.68455
14.5	2	3.2298	19.0099	2.0262	116.0934	5.5556	-0.10678
15	2	2.0081	21.5045	1.9728	113.0342	4.9921	9.1903
15.5	2	1.0314	23.8016	6.568	16.318	1.8401	-1.8676
16	3	1.9144	24.0601	5.6342	322.8143	0	0
16.5	1	1.9144	24.0601	5.6342	322.8143	5.5556	-0.051624
17	1	4.1274	22.3812	5.6084	321.3354	5.5556	-0.062848
17.5	1	6.2963	20.6457	5.5769	319.535	5.5556	-0.13764
18	1	8.4097	18.843	5.5081	315.5918	5.5556	-11.9668
18.5	2	10.394	16.8992	-0.47529	332.7676	5.5556	-0.1552
19	2	12.8639	15.6281	-0.5529	328.3213	5.5556	4.0646
19.5	3	15.2278	14.1693	1.4794	84.7648	0	0
20	1	15.2278	14.1693	1.4794	84.7648	5.5556	-0.059566
20.5	1	15.4813	16.9355	1.4496	83.0583	5.5556	-0.031353
21	1	15.817	19.6929	1.434	82.1601	5.5556	-0.22556
21.5	1	16.1959	22.4448	1.3212	75.6983	5.5556	-11.5671
22	2	16.8821	25.1364	-4.4624	104.3245	5.5556	11.2893
22.5	2	16.1948	27.8279	1.1823	67.74	2.8891	-2.3636
23	3	16.742	29.1647	0.00047275	0.027087	0	0

Table 6: Complete simulation results.

6 Conclusions and future developments

This work aims to develop a control strategy for a positioning problem of a mobile robotic platform modeled as a non-linear system. The problem in question can be considered an optimal control problem, where a suitable non-linear cost functional is assumed, which weighs the control through a piecewise state-dependent constant function. This approach can also be applied in other sectors such as, for example, in the medical, mechanical, economic, and telecommunications fields. The one used is a strategy that generates the control action by solving an optimization problem in a finite time interval, starting from a known initial condition, and managing, according to the current state, the state and the control inputs autonomously based on the parameters chosen in the cost functional, reaching the target position and minimizing the final positioning error at the same time.
Section 2 provides an adequate presentation of the theoretical notions necessary on the topic of optimal control, in particular, it illustrates the structure of the cost functional that will be used and provides a basic idea of the strategy that will be implemented in the following sections. Section 3 presents a non-linear model. The chosen mobile robotic platform, a Segway device, is described through the kinematic model of a unicycle. Section 4 definitively presents the formulation of the problem that makes use of the obtained mathematical model, describes its structure, and defines the cost functional appropriate to the solution of the optimization problem. Finally, Section 5 contains all the simulations, tests and practical considerations on the algorithm adopted. The main problem faced is that of the selection of the parameters, starting from the simulation times up to the choice of the values to be used in the cost functional to be minimized. The choice of parameters is a crucial aspect of the success of the tests and the solidity of the considerations made in the design phase. Although not having real specifications to satisfy, all the performances were in any case excellent in the simulation phase, a reason that highlights the correctness of the formulation of the problem and the effectiveness of a strategy that implies a switching of the cost functional based on the current position of the device. The consequence of this choice on a practical level is a deviation from the trajectory traveled up to the instant of switch, to then travel through a new optimal one, depending on the new parameters set at the time of the switching time. By testing in the simulation all the possible combinations of parameters it was thus possible to obtain the best setup among all the optimal solutions obtained. The results suggest that the control strategy that involves switching the problem according to the state of the system is much better than a simple strategy of achieving the goal with fixed parameters. The difference between the adopted strategy and the other is given by the precision and regularity of the trajectory to arrive at the point of arrival, accompanied both by the absence of abrupt maneuvers (although moving at considerable speeds), and by the savings in terms of distance traveled and time of arrival at the goal, details that in a real scenario can be relevant, especially if compared to the case of long distances traveled and energy (or fuel) savings.
The type of solution adopted also allows to easily adapt to the case in which there are ob-

stacles in the simulation (or real) environment. This can be an interesting development of this paper, considering for example also a different type of model of mobile devices, with the relative differences in the constraints that this choice would entail. And then conclude with the comparison of the results obtained on the simulator with the performance of mobile robotic platforms in the real world.

References

[1] Paolo Di Giamberardino, Daniela Iacoviello
An Optimal Control Problem Formulation for a State Dependent Resource Allocation Strategy
2017

[2] Daniel Liberzon - Princeton University Press
Calculus of Variations and Optimal Control Theory: A Concise Introduction
2012

[3] Lawrence C. Evans
An introduction to mathematical optimal control theory

[4] Silvio Simani, Marcello Bonfè
Tecniche di controllo
2011

[5] Claudio Melchiorri
Introduzione al controllo ottimo
2008

[6] Marco Gabiccini
Dinamica e Controllo dei Veicoli Robotici
2006

[7] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, Giuseppe Oriolo
Robotics: Modelling, Planning and Control
2009