2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Towards the analytical computation of time-optimal trajectories for unicycle robots in corridor environments

Sonia De Santis[1*], Alejandro Astudillo[1*], Wilm Decré[1] and Jan Swevers[1]

***Abstract*— This paper presents an analytical formulation for the time-optimal trajectory planning problem for unicycle robots in structured environments. We describe the environment as a sequence of rectangular corridors and compute the trajectory as a sequence of time-optimal motion primitives defined according to Pontryagin's minimum principle. The proposed approach uses a heuristic to place the center of arc maneuvers and, in certain situations, uses a turn on-the-spot primitive as initial or final maneuver to achieve time-optimality and obstacle avoidance. The computation of the trajectory within each corridor is decoupled and, therefore, the results can be applied to an arbitrarily long sequence of corridors while allowing parallel computations. The effectiveness of the approach is demonstrated by means of an example and Monte Carlo simulation: the suboptimality of the analytic motion is less than 1%, while the solution time is two orders of magnitude less than that of an optimal control problem formulation.**

I. Introduction

The use of autonomous mobile robots (AMR) has been the subject of increasing interest in modern society. Applications with AMRs have expanded from industrial environments to the fields of, e.g., medicine, agriculture, space, and logistics. Some AMRs – including differential drive and synchro drive robots – can be represented by the nonholonomic kinematic model of one steerable drive wheel, and are referred to as unicycle robots [1]. This model can be used as a low-fidelity model for complex systems [2], [3], [4] to rapidly plan safe trajectories to, e.g., initialize numerical algorithms. In this paper, the problem of analytically computing minimal motion time trajectories for unicycle robots moving in structured environments is addressed. This study is motivated by the fact that many AMR applications take place in structured environments whose free space can be represented as a set of rectangular corridors, e.g., crop inspection in greenhouses [5], last-mile urban goods transportation along sidewalks [6], service robotics in hospital hallways [7], or logistics in warehouses [8]. Such applications require to find the time-optimal trajectory from an initial pose to a desired final pose while considering environment- and robot-specific constraints, e.g., obstacle avoidance or minimum and maximum speeds.

For unicycle robots subject to a minimum turning radius (maximum curvature) constraint and constant positive translational velocity v, Dubins [9] proposed the use of a sequence of straight lines and arcs with maximum curvature to achieve the *shortest path* between two poses in *free space*. These paths, known as Dubins paths, were proven to be distance-optimal by using Pontryagin's minimum principle [10], [11], and are used in applications ranging from intercepting moving targets [12] to extensions with aerial vehicles in three-dimensional environments [13], [14].

While Dubins paths can be used to define distance-optimal trajectories, these paths do not aim to achieve time-optimality, especially when being applied to unicycle models that have a minimum turning radius equal to zero and can, therefore, rotate on-the-spot. Moreover, additional constraints have to be introduced to find a path that belongs to the free space delimited by rectangular corridors. This kind of problem can be formulated as an optimal control problem (OCP) that accounts for the total time of the trajectory in its performance objective. For real-time motion planning, however, the OCP approach may not be adequate due to high computation time, unpredictable solutions in case of bad initialization and local minima, or nonconvergence of the solution. Due to all these limitations, this work aims at finding a formulation of this time-optimal trajectory planning that (i) is near to optimality, (ii) requires a deterministic computation time that is orders of magnitude faster than OCP solutions and hence allows real-time implementations, and (iii) delivers predictable solutions. Note that the problem of finding a feasible sequence of rectangular corridors is not tackled in this work.

The main contributions of this work are as follows: (i) it proposes a sequence of maneuvers that describe the time-optimal motion of a unicycle robot to reach a point that lays on a given circumference, (ii) it analytically computes near time-optimal motions of unicycle robots within a sequence of rectangular corridors while allowing decoupling the computation of trajectories within each corridor of the sequence, and (iii) it extensively validates the proposed approach via numerical experiments.

The remaining of this paper is organized as follows. Section II introduces the problem formulation and preliminary concepts on optimal motion of unicycle robots. In Section III, the proposed approach is described in detail. Numerical experiments that show the effectiveness of the proposed method are presented in Section IV. Finally, Section V closes the paper with concluding remarks.

A. Notation

Let us define a point as $\boldsymbol{p} \coloneqq [x, y]^\top$ with homogeneous representation $\boldsymbol{p}_h \coloneqq [x, y, 1]^\top$, and the equation of a line on

[1]MECO Research Team, Dept. of Mechanical Engineering, KU Leuven and Flanders Make@KU Leuven, 3001 Leuven, Belgium. {sonia.desantis, alejandro.astudillovigoya, wilm.decre, jan.swevers}@kuleuven.be

This work has been carried out within the framework of the Flanders Make SBO project ARENA: Agile and Reliable Navigation. Flanders Make is the Flemish strategic research centre for the manufacturing industry.

*S. De Santis and A. Astudillo are equal contributors to this work.

a plane as $w_{x_j}x+w_{y_j}y+w_{d_j}=0$ or, compactly, $\boldsymbol{w}_j^\top \boldsymbol{p}_h = 0$, where $\boldsymbol{w}_j \coloneqq [w_{x_j}, w_{y_j}, w_{d_j}]^\top$ with $w_{x_j}, w_{y_j}, w_{d_j} \in \mathbb{R}$. Then, a rectangular area, referred to as *corridor*, can be defined as $\mathcal{C} \coloneqq \{\boldsymbol{p} : \boldsymbol{w}_j^\top \boldsymbol{p}_h \leq 0, j \in [1,4]\}$, where the subscript j is associated with each of the four lines to which the edges of the rectangle belong. The j-th edge of a corridor $\mathcal{C}$ is defined as $\boldsymbol{e}_j^{\mathcal{C}} \coloneqq \{\boldsymbol{p} : \boldsymbol{w}_j^\top \boldsymbol{p}_h = 0, \boldsymbol{p} \in \mathcal{C}\}$, where the index $j \in [1,4]$ is assigned starting from the top edge of the corridor and proceeding clockwise (see Fig. 1).

A sequence of $n_{\mathcal{C}} \in \mathbb{N}$ corridors is represented as $\boldsymbol{\mathcal{S}}_{\mathcal{C}} \coloneqq \{\mathcal{C}_i\}_{i\in[1,n_{\mathcal{C}}]}$. A shrunken corridor $\bar{\mathcal{C}}_i \subset \mathcal{C}_i$ is defined as a corridor whose center and orientation coincide with the original corridor $\mathcal{C}_i$, but its width and length have been reduced by $\epsilon \in \mathbb{R}_{>0}$. A circumference $\mathcal{O}_i$ is defined as $\mathcal{O}_i \coloneqq \{(x,y) : (x-x_{c_i})^2 + (y-y_{c_i})^2 = R_{c_i}^2\}$, where (x_{c_i}, y_{c_i}) and R_{c_i} are the center and radius of the circumference. For a maneuver involving a rotation, let us define a turn direction $\tau \in \{-1, 1\}$ determined by the sign of the angular velocity ω. The operator $\boldsymbol{\delta_\theta}(a,b)$ represents the smallest signed angle between angles a and b.

II. Problem Formulation

This section describes the preliminary aspects of the considered problem. First, formal descriptions of the unicycle kinematics and of the environment are given. Then, a description of the motion planning problem is detailed. Finally, the formulation of the motion planning problem as a time-optimal control problem is presented.

A. Robot kinematics

The unicycle pose is described by the state vector $\boldsymbol{x} \coloneqq [x, y, \theta]^\top \in \mathcal{X} \subset \mathrm{SE}(2)$, where $\mathrm{SE}(2)$ is the Special Euclidean Group in 2 dimensions, $[x,y]^\top \in \mathbb{R}^2$ corresponds to position coordinates and $\theta \in \mathbb{S}^1$ is the heading of the robot. Assuming a control vector $\boldsymbol{u} \coloneqq [v, \omega]^\top \in \mathcal{U} \coloneqq [v_{\min}, v_{\max}] \times [-\omega_{\max}, \omega_{\max}]$, where v is the robot's translational velocity in direction θ, and ω is the robot's rotational velocity, the kinematics of the unicycle robot are expressed by the following motion equations

$$\dot{x} = v\cos(\theta),\ \dot{y} = v\sin(\theta),\ \dot{\theta} = \omega. \tag{1}$$

The considered unicycle robot is characterized by a circular footprint with radius $r \in \mathbb{R}_{>0}$, as illustrated in Fig. 1.

B. Model of the environment

The environment considered in this work is structured and uncluttered. The free space is modeled as a sequence of rectangular frames – referred to as *corridors* – within which the robot can navigate (see Fig. 1). We assume that a suitable sequence of $n_{\mathcal{C}}$ corridors $\boldsymbol{\mathcal{S}}_{\mathcal{C}}$ is given a priori, such that a feasible trajectory exists to steer the robot from a given initial pose to a given final pose. Each corridor $\mathcal{C}_i \in \boldsymbol{\mathcal{S}}_{\mathcal{C}}$ is fully specified by the following set of parameters: the center point $(c_{x_i}, c_{y_i}) \in \mathbb{R}^2$, the orientation $\psi_i \in [0, 2\pi)$ with respect to the $\mathbf{x}$-axis, the width $w_i \in \mathbb{R}_{>0}$ and the length $l_i \in \mathbb{R}_{>0}$. Figure 1 shows an example of a sequence of two corridors $\mathcal{C}_1,\ \mathcal{C}_2$ with their shrunken version $\bar{\mathcal{C}}_1,\ \bar{\mathcal{C}}_2$, and reports the four edges $\boldsymbol{e}_j^{\mathcal{C}_1}$, $j \in [1,4]$ of $\mathcal{C}_1$. The amount by which the corridors are shrunken can be set depending on the robot's footprint and accounting on an additional safety margin. In this way, by constraining the (x,y) coordinates of the robot to be within the shrunken corridors, no collision with any wall will occur. Each pair of subsequent shrunken corridors has an overlapping area $\bar{\mathcal{C}}_i \cap \bar{\mathcal{C}}_{i+1}$ such that the robot is able to move from one corridor to the other.

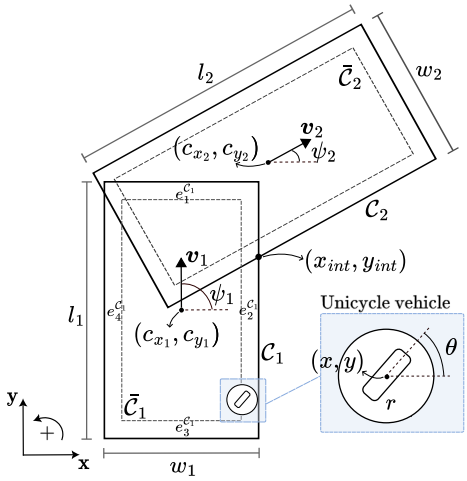


Fig. 1: Representation of two subsequent corridors $\mathcal{C}_1, \mathcal{C}_2$ and a unicycle robot with a circular footprint of radius r.

C. Problem statement

Having introduced the description of the robot and the environment, we now move on to describe the details of the motion planning problem addressed in this work. Let us define a unicycle robot with kinematics defined by (1). We aim to propose a solution to the following problem:

Problem 1: Find the time-optimal trajectory for such unicycle robot from a given initial pose $\bar{\boldsymbol{x}}_0 \coloneqq [x_0, y_0, \theta_0]^\top$ to a given final pose $\bar{\boldsymbol{x}}_f \coloneqq [x_f, y_f, \theta_f]^\top$ within a sequence of $n_{\mathcal{C}}$ corridors $\boldsymbol{\mathcal{S}}_{\mathcal{C}}$ – as defined in Section II-B – while keeping a safety distance r from the lateral walls of $\mathcal{C}_i\ \forall i \in [1, n_{\mathcal{C}}]$.

Assumption 2.1: The dimensions of the corridors are such that $w_i, l_i > 2r\ \forall i \in [1, n_{\mathcal{C}}]$, and $[x_0, y_0]^\top \in \mathcal{C}_1$ and $[x_f, y_f]^\top \in \mathcal{C}_{n_{\mathcal{C}}}$. The unicycle can only have nonnegative translational speed in its direction θ – i.e., $v_{\min} = 0$, $\mathcal{U} \coloneqq [0, v_{\max}] \times [-\omega_{\max}, \omega_{\max}]$.

For the sake of simplicity, but without loss of generality, we formulate the problem with only two corridors – i.e., $\boldsymbol{\mathcal{S}}_{\mathcal{C}} \coloneqq \{\mathcal{C}_1, \mathcal{C}_2\}$. However, the results presented herein can be extended to an arbitrarily long sequence of corridors, as shown in Section III.

D. Formulation as an optimal control problem

The motion problem described in Section II-C can be formulated as an OCP, as this methodology provides optimal solutions with respect to a specific performance objective while easily embedding the dynamics of the robot and constraints due to actuation limits and collision avoidance.

We formulate this problem as a multi-stage OCP with $n_{\mathcal{C}}$ stages, where every corridor $\mathcal{C}_i \in \boldsymbol{\mathcal{S}}_{\mathcal{C}}$ spatially defines each

stage, similarly to the approach in [15]. For the case of $n_{\mathcal{C}} = 2$, such OCP is defined as the following finite-dimensional problem:

$$\min_{\boldsymbol{x}_k^{[i]}, \boldsymbol{u}_k^{[i]}, T_i} \quad \boldsymbol{T} \coloneqq \sum_{i=1}^{n_{\mathcal{C}}=2} \boldsymbol{T}_i \tag{2a}$$

$$\text{s.t.} \quad \boldsymbol{x}_0^{[1]} = \bar{\boldsymbol{x}}_0, \tag{2b}$$

$$\boldsymbol{x}_{k+1}^{[1]} = F(\boldsymbol{x}_k^{[1]}, \boldsymbol{u}_k^{[1]}, \boldsymbol{T}_1), \quad k \in [0, N_1 - 1], \tag{2c}$$

$$\boldsymbol{p}_k^{[1]} \in \bar{\mathcal{C}}_1, \quad k \in [0, N_1 - 1], \tag{2d}$$

$$\boldsymbol{u}_k^{[1]} \in \mathcal{U}, \quad k \in [0, N_1 - 1], \tag{2e}$$

$$\boldsymbol{x}_0^{[2]} = \boldsymbol{x}_{N_1}^{[1]}, \tag{2f}$$

$$\boldsymbol{x}_{k+1}^{[2]} = F(\boldsymbol{x}_k^{[2]}, \boldsymbol{u}_k^{[2]}, \boldsymbol{T}_2), \quad k \in [0, N_2 - 1], \tag{2g}$$

$$\boldsymbol{p}_k^{[2]} \in \bar{\mathcal{C}}_2, \quad k \in [0, N_2 - 1], \tag{2h}$$

$$\boldsymbol{u}_k^{[2]} \in \mathcal{U}, \quad k \in [0, N_2 - 1], \tag{2i}$$

$$\boldsymbol{x}_{N_2}^{[2]} = \bar{\boldsymbol{x}}_f, \tag{2j}$$

where, for the i-th stage, $\boldsymbol{T}_i \in \mathbb{R}_{\geq 0}$ in the objective function (2a) denotes the motion time, N_i denotes the prediction horizon, and the superscript $[i]$ refers to the stage to which a variable belongs, $\forall i \in \{1, 2\}$. The constraints (2b) and (2j) set the initial and terminal poses to be the desired poses $\bar{\boldsymbol{x}}_0$ and $\bar{\boldsymbol{x}}_f$; function F in (2c) and (2g) represents a discretized version of (1) using a numerical integrator – e.g., forward Euler or fourth-order Runge-Kutta – transcribed by using a multiple-shooting scheme; the constraints (2d) and (2h) restrict the position of the center of the unicycle robot to be inside the i-th shrunken corridor, whose width and length have been reduced by $2r$ with respect to the original corridors. Finally, (2f) is a stitching constraint that links together the two stages.

Although the OCP formulation in (2) actively accounts for system- and task-related constraints while minimizing the total time of the trajectory, the numerical solution of such OCP has a high computational complexity – complicating or restricting its use in real-time applications –, can converge to infeasible points (or not converge at all), does not provide guarantees on the time required to deliver a solution, and may return suboptimal and unpredictable solutions due to bad initialization or local minima. Moreover, as will be explained in Section III-A, the solution of the OCP (2) requires the use of a nonuniform time grid – i.e., the N_i control points in each stage are not equally distributed along the time grid. This adds computational complexity to the solution of (2).

III. Proposed Approach

Having discussed the formulation of the problem, this section describes the proposed approach that aims at analytically computing time-optimal trajectories for unicycle robots in corridor environments. This approach is based on the use of time-optimal motion primitives defined by Pontryagin's minimum principle [10], and on heuristic rules to compute the center of the arc maneuvers appearing in the optimal sequence of primitives. The heuristic presented herein was defined based on extensive simulations and assessment of the solution of Problem 1 by using OCP (2). The proposed approach relies on (i) the computation of the time-optimal motion to reach a point on a circumference from a given pose $\bar{\boldsymbol{x}}_\bullet \in \mathcal{X}$, and (ii) the decoupled computation of a sequence of primitives for each corridor $\mathcal{C}_i \in \mathcal{S}_{\mathcal{C}}$ – from an initial pose $\bar{\boldsymbol{x}}_0$ and final pose $\bar{\boldsymbol{x}}_f$ to an arc that is part of a circumference $\mathcal{O}_c$ such that $(\mathcal{O}_c \cap (\mathcal{C}_i \cap \mathcal{C}_{i+1})) \neq \emptyset$, i.e., part of $\mathcal{O}_c$ is located in a region where corridor $\mathcal{C}_i$ intersects corridor $\mathcal{C}_{i+1}$.

A. Time-optimal motion primitives

As explained in detail in [2], Pontryagin's minimum principle can be used to find candidate time-optimal trajectory primitives for the unicycle robot moving in *free space*. By minimizing the Hamiltonian of the unicycle model while accounting for the constraint $\boldsymbol{u} \in \mathcal{U}$, the optimal control inputs $\boldsymbol{u}$ will correspond to (i) the vertices of the set $\mathcal{U}$, (ii) the singular control with $v = 0$ and $\omega = \pm\omega_{\max}$, or (iii) the singular control with $\omega = 0$ and $v = v_{\max}$, leading to a bang-bang control since both inputs should abruptly switch among these optimal values. These control inputs lead to five specific motion primitives: a linear segment S, a circular arc describing a counter-clockwise turn C^+, a circular arc describing a clockwise turn C^-, a counter-clockwise turn on-the-spot T^+, and a clockwise turn on-the-spot T^-. Recall that the turn direction $\tau \in \{-1, 1\}$ is determined by the sign of ω. Such sign determines the superscript of the primitives where $\omega \neq 0$. The description of each primitive is summarized in Table I.

TABLE I: Time optimal motion primitives

Symbol	$v(t)$	$\omega(t)$	τ
S	$v_{\max}$	0	$-$
C^+	$v_{\max}$	$\omega_{\max}$	1
C^-	$v_{\max}$	$-\omega_{\max}$	-1
T^+	0	$\omega_{\max}$	1
T^-	0	$-\omega_{\max}$	-1

Note that, the control inputs defining the primitives $C^\bullet$, i.e., $(v(t) = v_{\max},\ \omega(t) = \pm\omega_{\max})$, make the resulting path correspond to an arc with radius $R = v_{\max}/\omega_{\max}$.

Computing the time-optimal trajectory from any given start pose $\bar{\boldsymbol{x}}_0$ to a final pose $\bar{\boldsymbol{x}}_f$ in free space requires finding a proper sequence of the primitives shown in Table I. We describe a sequence of primitives $\boldsymbol{\mathcal{S}}_{\mathcal{P}}$ by concatenating their symbols – e.g., $T^-C^-SC^+$ represents a trajectory composed of a clockwise turn on-the-spot, a clockwise turn along a circular arc, a linear segment, and a final counter-clockwise turn along a circular arc, in that specific order.

As a final remark, since the control trajectory corresponding to a sequence of time-optimal primitives presents a bang-bang shape, to effectively formulate a time-optimal motion planning OCP for unicycle robots a *nonuniform time grid* must be employed, as it allows to freely select the time instants where to switch from one primitive to another and, therefore, retain the bang-bang control.

B. Time-optimal motion to reach a point on a circumference

In this section an important preliminary result is described that forms the basis for the proposed approach to solve Problem 1. Let us first address the following problem:

Problem 2: For a unicycle robot, find the time-optimal trajectory in free space from an initial pose $\bar{\boldsymbol{x}}_0 := [x_0, y_0, \theta_0]^\top$ to a final pose $\bar{\boldsymbol{x}}_t := [x_t, y_t, \theta_t]^\top$, where (x_t, y_t) belongs to a given target circumference $\mathcal{O}_2$ with center (x_{c_2}, y_{c_2}) and radius $R = v_{\max}/\omega_{\max}$, and θ_t represents an orientation tangential to $\mathcal{O}_2$ according to a (clockwise or counter-clockwise) turn direction $\tau_2 \in \{-1, 1\}$.

The solution to Problem 2 entails the proper selection of a sequence $\boldsymbol{\mathcal{S}}_\mathcal{P}$ of time-optimal motion primitives of the types described in Section III-A. The shape of the solution will depend on the turn directions of the selected primitives, as shown in Fig. 2.

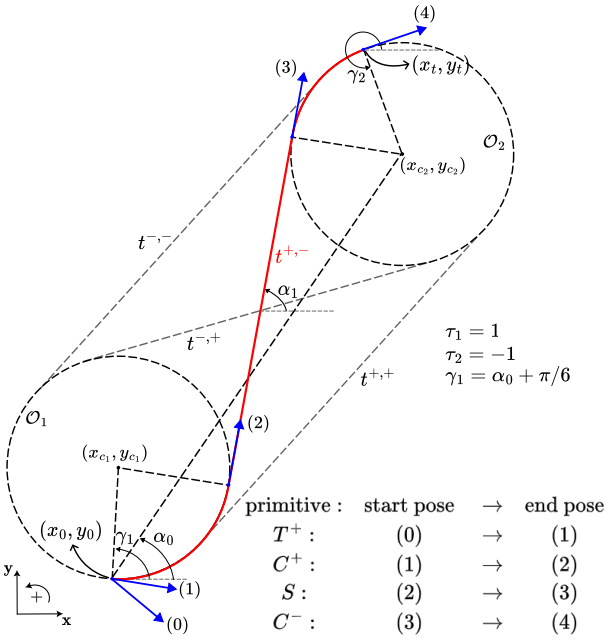


Fig. 2: Overview of a trajectory computed when $\tau_1 = 1$ (counter-clockwise initial turn) and $\tau_2 = -1$ (clockwise final turn). $\boldsymbol{\mathcal{S}}_\mathcal{P}$ is the sequence $T^+C^+SC^-$.

Assumption 3.1: Since $(x_t, y_t) \in \mathcal{O}_2$ and θ_t is an orientation tangential to $\mathcal{O}_2$, we assume that the final primitive in the sequence $\boldsymbol{\mathcal{S}}_\mathcal{P}$ is an arc C^+ if $\tau_2 = 1$ or C^- if $\tau_2 = -1$.

Requiring an initial turn on-the-spot: By using Dubins paths, the solution to Problem 2 would require a sequence $C^{\bullet_1}SC^{\bullet_2}$, where $\bullet_1, \bullet_2 \in \{+, -\}$ depend on initial and final turn directions τ_1 and τ_2 [16]. However, this solution can be generalized to include an initial turn on-the-spot with same turn direction as $C^{\bullet_1}$, in case needed to achieve time-optimality, as stated in the following proposition.

Proposition 3.1: The time-optimal trajectory that solves Problem 2 is composed of a sequence $\boldsymbol{\mathcal{S}}_\mathcal{P}$ of motion primitives being either $T^{\bullet_1}C^{\bullet_1}SC^{\bullet_2}$ or $C^{\bullet_1}SC^{\bullet_2}$.

This proposition is directly proved for the case where the initial and final turn directions are equal ($\tau_1 = \tau_2$) by the following lemma.

Let $\alpha_0 := \arctan(y_{c_2} - y_0 / x_{c_2} - x_0)$ describe the direction from (x_0, y_0) to (x_{c_2}, y_{c_2}) (see also Fig. 2).

Lemma 3.1: When the initial and final turn directions are equal ($\tau_1 = \tau_2$), if $|\boldsymbol{\delta_\theta}(\theta_0, \alpha_0)| > \frac{\pi}{2}$, then the first primitive of $\boldsymbol{\mathcal{S}}_\mathcal{P}$ is a turn on-the-spot $T^{\bullet_1}$ and the arc described in the subsequent primitive $C^{\bullet_1}$ has a center (x_{c_1}, y_{c_1}) with $x_{c_1} = x_0 + R\cos(\gamma_1)$ and $y_{c_1} = y_0 + R\sin(\gamma_1)$, where $\gamma_1 = \alpha_0$.

Proof: Refer to the proof in the Appendix. ■

For the case where the initial and final turn directions are different ($\tau_1 \neq \tau_2$), Proposition 3.1 relies on the following heuristic:

Heuristic 1: When the initial and final turn directions are different ($\tau_1 \neq \tau_2$), if $|\boldsymbol{\delta_\theta}(\theta_0, \alpha_0 + \tau_1 \frac{\pi}{6})| > \frac{\pi}{2}$, then the first primitive of $\boldsymbol{\mathcal{S}}_\mathcal{P}$ is a turn on-the-spot $T^{\bullet_1}$ and the arc described in the subsequent primitive $C^{\bullet_1}$ has a center (x_{c_1}, y_{c_1}) with $x_{c_1} = x_0 + R\cos(\gamma_1)$ and $y_{c_1} = y_0 + R\sin(\gamma_1)$, where $\gamma_1 = \alpha_0 + \tau_1 \frac{\pi}{6}$, as depicted in Fig. 2.

Computation of the motion primitives: Based on the aforementioned proposition, we now illustrate how to compute each primitive in $\boldsymbol{\mathcal{S}}_\mathcal{P}$. Each primitive is characterized by its starting pose $\boldsymbol{x}_i^s = [x_i^s, y_i^s, \theta_i^s]^\top$ and ending pose $\boldsymbol{x}_i^e = [x_i^e, y_i^e, \theta_i^e]^\top$, where i is the position of the primitive in $\boldsymbol{\mathcal{S}}_\mathcal{P}$. Note that $\boldsymbol{x}_{i+1}^s = \boldsymbol{x}_i^e,\ \forall i \in [1, n_{\boldsymbol{\mathcal{S}}_\mathcal{P}} - 1]$, where $n_{\boldsymbol{\mathcal{S}}_\mathcal{P}}$ is the number of primitives in $\boldsymbol{\mathcal{S}}_\mathcal{P}$.

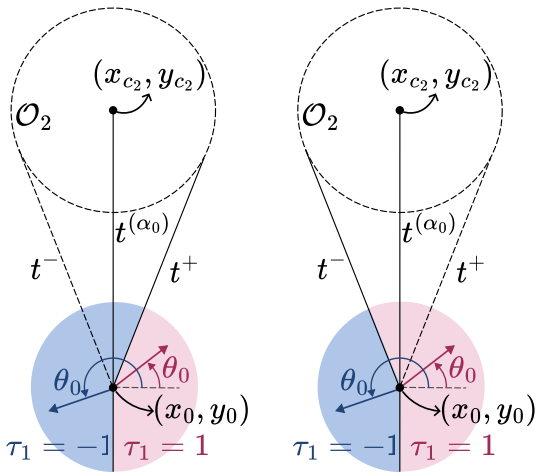
The initial turn direction τ_1 is determined depending on the initial orientation of the robot θ_0, the target circumference $\mathcal{O}_2$ and the turn direction τ_2, through a procedure denoted as $\tau_1 = \texttt{turn\_direction}(\tau_2, \mathcal{O}_2, \theta_0)$. Specifically, by denoting with t^+ and t^- the two lines passing through (x_0, y_0) and tangent to $\mathcal{O}_2$, and with $t^{(\alpha_0)}$ the line passing through (x_0, y_0) and (x_{c_2}, y_{c_2}), as shown in Fig. 3, then:

- when $\tau_2 = 1$, then $\tau_1 = 1$ if θ_0 lies on the region to the right of the half plane identified by the lines t^+ and $t^{(\alpha_0)}$ (green area in Fig. 3a), otherwise $\tau_1 = -1$ (orange area in Fig. 3a).
- when $\tau_2 = -1$, then $\tau_1 = -1$ if θ_0 lies on the region to the left of the half plane identified by the lines t^- and $t^{(\alpha_0)}$ (orange area in Fig. 3b), otherwise $\tau_1 = 1$ (green area in Fig. 3b).

Once τ_1 is computed, starting from $\boldsymbol{x}_1^s = \bar{\boldsymbol{x}}_0$, the computation of each primitive in $\boldsymbol{\mathcal{S}}_\mathcal{P}$ is performed as follows:

1) Turn on-the-spot $T^{\bullet_1}$: The initial turn on-the-spot is required if the condition described in Lemma 3.1 or Heuristic 1 holds. This primitive involves a rotation by an angle Δ_θ, computed as $\Delta_\theta = \boldsymbol{\delta_\theta}(\theta_0, \alpha_0 - \tau_1 \frac{\pi}{2})$ if $\tau_1 = \tau_2$, or $\Delta_\theta = \boldsymbol{\delta_\theta}(\theta_0, \alpha_0 - \tau_1 \frac{\pi}{2} + \tau_1 \frac{\pi}{6})$ if $\tau_1 \neq \tau_2$. Since this primitive does not involve any translation, its final pose is $\boldsymbol{x}_i^e = [x_i^s, y_i^s, \theta_i^s + \Delta_\theta]^\top$.

2) Arc $C^{\bullet_1}$: This arc belongs to a circumference $\mathcal{O}_1$ with radius R and center $(x_{\mathcal{O}_1}, y_{\mathcal{O}_1}) = (x_0 + R\cos(\theta_i^s + \tau_1 \frac{\pi}{2}),\ y_0 + R\sin(\theta_i^s + \tau_1 \frac{\pi}{2}))$, where θ_i^s either corresponds to the final orientation of the previous primitive $T^{\bullet_1}$, i.e., $\theta_i^s = \theta_{i-1}^e$, or $\theta_i^s = \theta_0$ if no turn on-the-spot is performed beforehand. The end pose is determined by computing the intersection point between $\mathcal{O}_1$ and the line $t^{\bullet_1,\bullet_2}$ tangent



(a) $\tau_2 = 1$. (b) $\tau_2 = -1$.

Fig. 3: Selection of τ_1, depending on θ_0, τ_2 and $\mathcal{O}_2$.

to $\mathcal{O}_1$ and $\mathcal{O}_2$, where $\bullet_1, \bullet_2$ refer to the sign of τ_1, τ_2, respectively.

3) Segment S: This segment belongs to the line $t^{\bullet_1,\bullet_2}$, with start pose corresponding to the final pose of the previous arc $C^{\bullet_1}$, and with the end point being the intersection point between $t^{\bullet_1,\bullet_2}$ and $\mathcal{O}_2$.

4) Arc $C^{\bullet_2}$: This arc belongs to the circumference $\mathcal{O}_2$, starting from the final pose of the previous primitive S, and ending on the arbitrary pose $\boldsymbol{x}_t$.

Remark 1: In the following, without loss of generality, the time-optimal sequence solution to Problem 2 will be always denoted as the sequence of four primitives $T^{\bullet_1}C^{\bullet_1}SC^{\bullet_2}$. When an initial turn on-the-spot is not required, $T^{\bullet_1}$ is assigned with a zero rotation $\Delta_\theta = 0$, i.e., $\theta_1^s = \theta_1^e$.

Remark 2: Since the computation of $\boldsymbol{\mathcal{S}}_{\mathcal{P}}$ involves the computation of tangent lines between circumferences, we assume that the distance between the center of every pair of subsequent circumferences is always bigger than $2R$.

C. Trajectory within a sequence of corridors

In view of the previous result, the problem of navigating through a sequence of corridors (Problem 1) is addressed by reformulating it as the problem of reaching predefined *intermediate circumferences* that are strategically placed at the intersection between each pair of subsequent corridors.

Trajectory within two corridors: As an example, consider that the sequence of corridors $\boldsymbol{\mathcal{S}}_{\mathcal{C}} := \{\mathcal{C}_1, \mathcal{C}_2\}$ is provided. In this case, it is possible to divide the problem of finding a trajectory from an initial configuration $\bar{\boldsymbol{x}}_0 \in \mathcal{C}_1$ to a final configuration $\bar{\boldsymbol{x}}_f \in \mathcal{C}_2$ in two parts. This division is achieved by properly positioning an *intermediate circumference* $\mathcal{O}_2$ such that $(\mathcal{O}_2 \cap (\mathcal{C}_1 \cap \mathcal{C}_2)) \neq \emptyset$, and constructing the overall trajectory as the connection of the optimal sequences (i) $T_1^{\bullet}C_1^{\bullet}S_1C_2^{\bullet}$ from $\bar{\boldsymbol{x}}_0$ to $\mathcal{O}_2$ (as in Problem 2), and (ii) $C_2^{\bullet}S_2C_3^{\bullet}T_2^{\bullet}$ from $\mathcal{O}_2$ to $\bar{\boldsymbol{x}}_f$.

Note that the latter sequence is obtained by first computing the sequence $\widetilde{T}_2^{\bullet}\widetilde{C}_3^{\bullet}\widetilde{S}_2\widetilde{C}_2^{\bullet}$ from the inverted final pose $\widetilde{\boldsymbol{x}}_f := [x_f,\ y_f,\ \theta_f + \pi]^\top$ (with a modified final orientation) to $\mathcal{O}_2$ – where $\widetilde{\bullet}$ represents an alternative primitive that is later inverted according to Remark 3 –, and then inverting the resulting sequence through an operator denoted as `invert_seq` such that $C_2^{\bullet}S_2C_3^{\bullet}T_2^{\bullet} = \texttt{invert\_seq}(\widetilde{T}_2^{\bullet}\widetilde{C}_3^{\bullet}\widetilde{S}_2\widetilde{C}_2^{\bullet})$.

Remark 3: The inversion of a sequence of primitives through the operator `invert_seq` involves inverting the order of the primitives and for each primitive (i) inverting the sign of ω and, therefore, τ, (ii) switching the initial and final poses $\boldsymbol{x}_i^s$ and $\boldsymbol{x}_i^e$, (iii) rotating θ_i^s and θ_i^e by π.

Since both the final pose of $\boldsymbol{\mathcal{S}}_{\mathcal{P}}^{[1]} := T_1^{\bullet}C_1^{\bullet}S_1C_2^{\bullet}$ and the initial pose of $\boldsymbol{\mathcal{S}}_{\mathcal{P}}^{[2]} := C_2^{\bullet}S_2C_3^{\bullet}T_2^{\bullet}$ are *arbitrary* poses located on $\mathcal{O}_2$, it is possible to select them as the final pose of S_2 and the final pose of S_1, respectively. By doing so, $C_2^{\bullet}$ can be redefined as the arc connecting S_1 and S_2 via the circumference $\mathcal{O}_2$, ensuring the continuity of the trajectory. As a result, $\boldsymbol{\mathcal{S}}_{\mathcal{P}} := T_1^{\bullet}C_1^{\bullet}S_1C_2^{\bullet}S_2C_3^{\bullet}T_2^{\bullet}$ represents a candidate solution for Problem 1. However, the effectiveness of the approach resides in the selection of the location of $\mathcal{O}_2$, which is performed according to the heuristic explained in the following.

Recall that the *intermediate circumference* $\mathcal{O}_2$ is defined by its center (x_{c_2}, y_{c_2}) and its radius $R = v_{\max}/\omega_{\max}$. The turn direction τ_2 is selected based on the relative orientation between the two corridors. Let $\boldsymbol{v}_1 = [\cos(\psi_1),\ \sin(\psi_1),\ 0]^\top$ and $\boldsymbol{v}_2 = [\cos(\psi_2),\ \sin(\psi_2),\ 0]^\top$ be the unit vectors in three-dimensional space representing the orientation of $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively. The turn direction τ_2 is computed as the sign of the third element of the cross product $\boldsymbol{v}_1 \times \boldsymbol{v}_2$, as it determines whether a counter-clockwise turn (if $\tau_2 = 1$) or a clockwise turn (if $\tau_2 = -1$) is needed to align $\mathcal{C}_1$ with $\mathcal{C}_2$. This procedure is denoted as $\tau_2 = \texttt{main\_turn\_direction}(\mathcal{C}_1, \mathcal{C}_2)$.

The center (x_{c_2}, y_{c_2}) of $\mathcal{O}_2$ is placed according to the following heuristic, which was determined following the assessment of extensive simulation results obtained by using OCP (2) to solve Problem 1:

Heuristic 2: We place (x_{c_2}, y_{c_2}) at a distance $R - r$ from one of the intersection points between the edges of the two corridors, which is selected based on τ_2. By expressing the coordinates of the intersection points between the edges of the two corridors *relatively* to a frame attached at the center of (and aligned with) $\mathcal{C}_1$, the leftmost intersection point is selected if $\tau_2 = 1$ (characterized by the lower relative x coordinate), or the rightmost intersection point is selected if $\tau_2 = -1$ (characterized by the larger relative x coordinate). The coordinates of the selected intersection point in the world frame are denoted as $(x_{\text{int}}, y_{\text{int}}) \in \mathcal{X}$ (see Fig. 1 for an example). After finding $(x_{\text{int}}, y_{\text{int}})$, the center of the circle (x_{c_2}, y_{c_2}) is placed at a distance $R - r$ from it, along the bisector of the convex angle defined by the two edges to which the intersection point belongs (see also Fig. 4). This procedure is denoted as $\mathcal{O}_2 = \texttt{circ\_center}(\tau_2, R, \text{r}, \mathcal{C}_1, \mathcal{C}_2)$.

Avoiding collisions with corridor walls: Once $\mathcal{O}_2$ is in place, the sequence $\boldsymbol{\mathcal{S}}_{\mathcal{P}} := T_1^{\bullet}C_1^{\bullet}S_1C_2^{\bullet}S_2C_3^{\bullet}T_2^{\bullet}$ could be computed as described earlier in this section.

However, this sequence could cross the edges of the

Algorithm 1 Computation of trajectory within corridors

Inputs: $\boldsymbol{\mathcal{S}}_{\mathcal{C}} \coloneqq \{\mathcal{C}_i\}_{i\in[1,n_{\mathcal{C}}]}$, $\bar{\boldsymbol{x}}_0$, $\bar{\boldsymbol{x}}_f$, r, $R = v_{\max}/\omega_{\max}$

1: $\boldsymbol{\mathcal{S}}_{\mathcal{P}} \leftarrow \emptyset$
2: **for all** $i \in [2, n_{\mathcal{C}}]$ **do in parallel**
3: $\quad \tau_i \leftarrow$ `main_turn_direction`$(\mathcal{C}_{i-1}, \mathcal{C}_i)$
4: $\quad \mathcal{O}_i \leftarrow$ `circ_center`$(\tau_i, R, r, \mathcal{C}_{i-1}, \mathcal{C}_i)$
5: **end for**
6: $\tau_1 \leftarrow$ `turn_direction`$(\tau_2, \mathcal{O}_2, \theta_0)$
7: $\boldsymbol{\mathcal{S}}_{\mathcal{P}_0} \leftarrow$ `get_subseq`$(\mathcal{C}_1, \boldsymbol{x}_0, R, r, \mathcal{O}_2, \tau_1, \tau_2)$
8: $\tau_{n_{\mathcal{C}}+1} \leftarrow$ `turn_direction`$(\tau_{n_{\mathcal{C}}}, \mathcal{O}_{n_{\mathcal{C}}}, \theta_f)$
9: $\widetilde{\boldsymbol{\mathcal{S}}}_{\mathcal{P}_f} \leftarrow$ `get_subseq`$(\mathcal{C}_{n_{\mathcal{C}}}, \boldsymbol{x}_f, R, r, \mathcal{O}_{n_{\mathcal{C}}}, \tau_{n_{\mathcal{C}}}, \tau_{n_{\mathcal{C}+1}})$
10: $\boldsymbol{\mathcal{S}}_{\mathcal{P}_f} \leftarrow$ `invert_seq`$(\widetilde{\boldsymbol{\mathcal{S}}}_{\mathcal{P}_f})$
11: **for all** $i \in [2, n_{\mathcal{C}} - 1]$ **do in parallel**
12: $\quad S_i \leftarrow$ `compute_tangent`$(\mathcal{O}_i, \mathcal{O}_{i+1}, \tau_i, \tau_{i+1})$
13: **end for**
14: **for all** $i \in [2, n_{\mathcal{C}}]$ **do in parallel**
15: $\quad C_i^{\bullet} \leftarrow$ `compute_arc`$(S_i, S_{i+1}, \mathcal{O}_{i+1})$
16: **end for**
17: Append $\boldsymbol{\mathcal{S}}_{\mathcal{P}_0}$ to $\boldsymbol{\mathcal{S}}_{\mathcal{P}}$
18: **for all** $i \in [1, n_{\mathcal{C}} - 1]$ **do**
19: $\quad$ Append $C_{i+1}^{\bullet} S_{i+1}$ to $\boldsymbol{\mathcal{S}}_{\mathcal{P}}$
20: **end for**
21: Append $C_{n_{\mathcal{C}}}$ to $\boldsymbol{\mathcal{S}}_{\mathcal{P}}$
22: Append $\boldsymbol{\mathcal{S}}_{\mathcal{P}_f}$ to $\boldsymbol{\mathcal{S}}_{\mathcal{P}}$
23: **return** $\boldsymbol{\mathcal{S}}_{\mathcal{P}}$

shrunken corridors $\bar{\mathcal{C}}_1$ and $\bar{\mathcal{C}}_2$, leading to a collision between the robot and the corridor walls. In particular, the primitives $C_1^{\bullet}$ and $C_3^{\bullet}$ could cross the lateral edges of $\bar{\mathcal{C}}_1$ and $\bar{\mathcal{C}}_2$, respectively. Therefore, it is necessary to check whether there are intersection points between (i) the path corresponding to $C_1^{\bullet}$ and the edges $\boldsymbol{e}_2^{\bar{\mathcal{C}}_1}$, $\boldsymbol{e}_4^{\bar{\mathcal{C}}_1}$ and (ii) the path corresponding to $C_3^{\bullet}$ and the edges $\boldsymbol{e}_2^{\bar{\mathcal{C}}_2}$, $\boldsymbol{e}_4^{\bar{\mathcal{C}}_2}$.

If an intersection is detected, a turn on-the-spot primitive is required in such a way that the robot reaches an orientation that allows the remaining primitives to entirely lie within the corridors. In the case that this rotation for collision avoidance is required, $\mathcal{O}_1$ is tangent to the edge with which the collision was detected. If a turn on-the-spot primitive was already required in the original sequence, the additional rotation for collision avoidance just results in a prolonged turn on-the-spot. For the sake of clarity, consider an example where an intersection point is detected between the arc $C_1^{\bullet}$ and the edge $\boldsymbol{e}_4^{\bar{\mathcal{C}}_1}$. In this case, an additional Δ_θ is required such that $\mathcal{O}_1$ touches $\boldsymbol{e}_4^{\bar{\mathcal{C}}_1}$. Note that there exist two circumferences tangent to a given line and passing through a given point. The selected one is the closest to $\mathcal{O}_2$.

Note that the solution to avoid collisions with a corridor wall is not to decrease the radius of $\mathcal{O}_1$ to fit the shrunken corridor, as it would imply a solution that is not time-optimal since an arc motion primitive with a decreased radius would not be part of the optimal selection presented in Table I.

Complete algorithm: The procedure to build a trajectory in a sequence of $n_{\mathcal{C}}$ corridors is described in Algorithm 1, which performs the subroutines explained in the following.

After initializing the sequence $\boldsymbol{\mathcal{S}}_{\mathcal{P}}$ in Line 1, the algorithm computes the turn direction τ_i and the corresponding intermediate circumference $\mathcal{O}_i$ for all pairs of subsequent corridors $\mathcal{C}_{i-1}, \mathcal{C}_i,\ i \in [2, n_{\mathcal{C}}]$ in Lines 2-5. Then, it computes the initial turn direction τ_1 in Line 6 and the motion primitives $\boldsymbol{\mathcal{S}}_{\mathcal{P}_0} \coloneqq T_1^{\bullet} C_1^{\bullet} S_1$ required to reach $\mathcal{O}_2$ from $\bar{\boldsymbol{x}}_0$ in Line 7 with the function `get_subseq`. Next, it computes the final turn direction $\tau_{n_{\mathcal{C}}+1}$ and the sequence of primitives $\boldsymbol{\mathcal{S}}_{\mathcal{P}_f} \coloneqq S_{n_{\mathcal{C}}} C_{n_{\mathcal{C}}+1}^{\bullet} T_2^{\bullet}$ required to reach the final pose $\bar{\boldsymbol{x}}_f$ from $\mathcal{O}_{n_{\mathcal{C}}}$ in Lines 8-10. Since all the circumferences $\mathcal{O}_i$ are known, the algorithm proceeds by computing all segments S_i, $i \in [2, n_{\mathcal{C}} - 1]$, in Lines 11-13, by using the procedure `compute_tangent`, which computes the line segment $t^{\bullet_i \bullet_{i+1}}$ tangent to both $\mathcal{O}_i$ and $\mathcal{O}_{i+1}$ according to the desired turn directions τ_i and τ_{i+1}, $i \in [2, n_{\mathcal{C}} - 1]$.

Then, in Lines 14-16, the arc maneuvers $C_i^{\bullet}$, $i \in [2, n_{\mathcal{C}}]$ are computed by using the procedure `compute_arc` to connect together the two segments S_i and S_{i+1} through the circumference $\mathcal{O}_{i+1}$. Finally, the overall trajectory is obtained by properly appending the computed primitives in Lines 17-23.

Note that, the for-loops starting in Lines 2, 11 and 14 can each be executed in parallel since their internal computations are independent and decoupled from each other. In fact, such decoupling allows the replanning of time-optimal motions by simply updating relevant subsequences – e.g., the initial sequence from the robot's current pose to the first intermediate circumference – while keeping the remaining elements in the the sequence untouched.

IV. Numerical experiments

In this section we demonstrate the effectiveness of the approach presented in Section III by means of two numerical experiments: an illustrative example and an extensive evaluation of the optimality of the approach by using Monte Carlo simulation.

A. Illustrative example

The considered scenario is that of a unicycle robot moving within two corridors. We consider a Clearpath Jackal differential drive robot with $v_{\max} = 0.5$ m/s, $\omega_{\max} = 0.5$ rad/s, $r = 0.215$ m. The two corridors are defined as: (i) $\mathcal{C}_1$ with $\psi_1 = \frac{\pi}{2}$, $l_1 = 5$, $w_1 = 2$, center $(0,\ 2.5)$, and (ii) $\mathcal{C}_2$ with $\psi_2 = \frac{\pi}{6}$, $l_2 = 5$, $w_2 = 2$, center $(2.165,\ 6.25)$. The shrunken corridors $\bar{\mathcal{C}}_1$ and $\bar{\mathcal{C}}_2$ are characterized by a length $\bar{l}_i = l_i - 2r$ and width $\bar{w}_i = w_i - 2r$, with $i = 1, 2$. The time-optimal trajectories computed by using the proposed approach and OCP (2) are shown in Fig. 4, while their corresponding control inputs are shown in Fig. 5.

As shown in Fig. 5, and as anticipated in Section III-A, the required control inputs to achieve time-optimality have a bang-bang behaviour. Note that, an initial turn on-the-spot is required. As a consequence, $v = 0$ for the first 1.37 s and the center of $\mathcal{O}_1$ is along the direction identified by $\gamma_1 = \alpha_0 + \tau_1 \pi/6$, according to Heuristic 1, since $\tau_1 \neq \tau_2$.

The analytical solution is extremely close to the time-optimal solution, with a motion time difference between the two of only 0.03%. Nevertheless, the introduced sub-optimality represents only a small disadvantage, especially

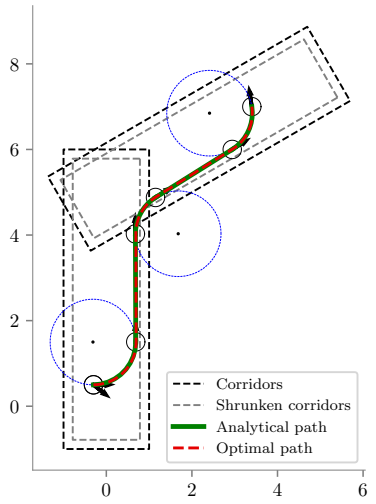


Fig. 4: Spatial trajectories computed for the scenario considered in the illustrative example.

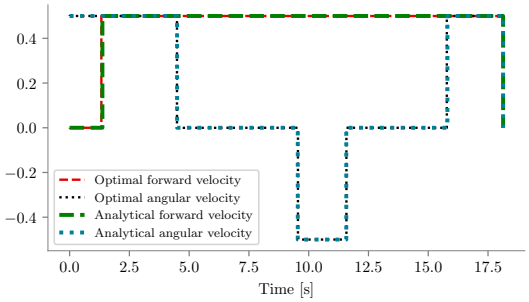


Fig. 5: Control input trajectories for the considered scenario.

considering the low computational time of the analytical planner with respect to the computational time of the optimal planner, which in this case are 1.64 ms and 139.72 ms, respectively.

B. Monte Carlo simulation

A Monte Carlo simulation was performed to evaluate the proposed approach under a wide range of scenarios involving the computation of a trajectory for a unicycle robot with $r = 0.215$ m from a given initial pose to a final one within a sequence of two corridors.

To perform this simulation, a large number of problems were set up, each one defined by a different set of parameters, and then the solution to the problem was obtained by (i) solving an OCP formulated as in (2) and (ii) running Algorithm 1. The resulting trajectories are compared in terms of total motion time and time required to compute the solution. In the tested problems, $v_{\max}$ and $\omega_{\max}$ are set such that $R = v_{\max}/\omega_{\max} = 1$. Without loss of generality, the orientation of $\mathcal{C}_1$ is fixed at $\psi_1 = \frac{\pi}{2}$. The parameters used to define the motion problems and their domain are reported in Table II. Then, with the set of parameters presented in Table II, a trajectory planning problem is defined in the following way. First, the two corridors, $\mathcal{C}_1$ and $\mathcal{C}_2$, are constructed using two consecutive vectors, $v_i, i \in [1, 2]$,

TABLE II: Parameters Monte Carlo simulation

Variable	Symbol	Value	Unit
Corridor length	l	10	m
Corridor width	w	$[0.3l,\ 0.6l]$	m
Tilt of $\mathcal{C}_2$	ψ_2	$[\frac{\pi}{6},\ \frac{5\pi}{6}]$	rad
Max speed	$v_{\max}$	$[0.5,\ 2]$	m/s
x_0 relative to $\mathcal{C}_1$	x_0^{rel}	$[-0.5\frac{w}{2},\ 0.5\frac{w}{2}]$	m
y_0 relative to $\mathcal{C}_1$	y_0^{rel}	$[-0.7\frac{l}{2},\ -0.2\frac{l}{2}]$	m
θ_0 relative to $\mathcal{C}_1$	θ_0^{rel}	$[\frac{\pi}{3},\ \frac{5\pi}{6}]$	rad
x_f relative to $\mathcal{C}_2$	x_f^{rel}	$[-0.5\frac{w}{2},\ 0.5\frac{w}{2}]$	m
y_f relative to $\mathcal{C}_2$	y_f^{rel}	$[0.2\frac{l}{2},\ 0.7\frac{l}{2}]$	m
θ_f relative to $\mathcal{C}_2$	θ_f^{rel}	$[\frac{\pi}{3},\ \frac{5\pi}{6}]$	rad

which are determined by the coordinates of their tail (x_i^t, y_i^t) and their head (x_i^h, y_i^h), defined as $(x_1^t, y_1^t) = (0, 0)$, $(x_1^h, y_1^h) = (l\cos(\psi_1),\ l\sin(\psi_1))$, $(x_2^t, y_2^t) = (x_1^h, y_1^h)$, and $(x_2^h, y_2^h) = (x_2^t + l\cos(\psi_2),\ y_2^t + l\sin(\psi_2))$. Then, the centers of the two corridors correspond to the midpoints of v_1 and v_2, respectively. Their widths w_1, w_2 are both equal to w, and their lengths are augmented by a quantity equal to the width w to account for the necessary overlapping area between the two corridors, thus $l_1, l_2 = l + w$. The shrunken corridors $\bar{\mathcal{C}}_1$ and $\bar{\mathcal{C}}_2$ are then computed, with $\bar{l}_i = l_i - 2r$ and $\bar{w}_i = w_i - 2r$, with $i = 1, 2$. The initial pose $\bar{\boldsymbol{x}}_0 = [x_0, y_0, \theta_0]^T$ is computed by evaluating the relative pose $[x_0^{\text{rel}}, y_0^{\text{rel}}, \theta_0^{\text{rel}}]^\top$ with respect to a reference frame centered in (c_{x_1}, c_{y_1}) and rotated by ψ_1 clockwise, while the final pose $\bar{\boldsymbol{x}}_f = [x_f, y_f, \theta_f]^T$ is computed by evaluating the relative pose $[x_f^{\text{rel}}, y_f^{\text{rel}}, \theta_f^{\text{rel}}]^\top$ with respect to a reference frame centered in (c_{x_2}, c_{y_2}) and rotated by ψ_2 clockwise. With the remaining parameters it is possible to instantiate a unicycle robot, and start a simulation. Denoting as n_p the number of varying parameters, the total number of simulations is given by $4^{n_p} = 4^9 = 262144$.

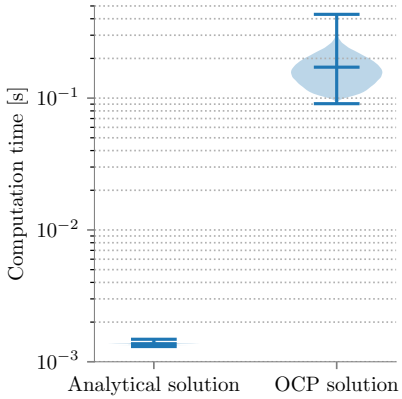


Fig. 6: Comparison of the time required to solve the motion problems within the Monte Carlo simulation.

As can be seen in Fig. 6, the proposed approach – labeled in the figure as Analytical Solution – is about two orders of magnitude faster than the numerical optimization-based approach. In fact, the analytical solution has a mean computation time of 1.29 ms with a standard deviation of 2.05×10^{-2} ms, while the OCP solution has a mean computation time of 174.19 ms with a standard deviation of

49.01 ms. Due to its numerical behaviour, the OCP solution has high variability in its computation time, as shown by its high standard deviation. On the other side, the analytical solution has low variability in its computation time due to the limited number of operations it requires.

In terms of optimality of the solution, the results of the Monte Carlo simulation showed that the difference between the motion time of the OCP-based solution and that of the proposed approach is less than 1% in 97.43% of the cases, having a maximum difference of 3.46% after 262144 simulations. An explanation for this larger difference might be the use of Heuristics 1 and 2 to position the center of the first arc (when $\tau_1 \neq \tau_2$) and the center of $\mathcal{O}_2$, respectively.

V. Conclusions and Outlook

In this paper we presented an analytical approach that aims to compute time-optimal solutions to motion problems of unicycle robots moving in corridor environments. The approach relies on time-optimal motion primitives and heuristics to position such primitives. The results of this study indicate that, under certain conditions, reaching time-optimality and satisfying obstacle avoidance require a turn on-the-spot maneuver, in addition to the arc and straight line maneuvers used in Dubins paths. The proposed approach was shown to be only 1% away from optimality in 97.43% of the simulated cases while being two orders of magnitude faster than a numerical optimization-based approach.

The presented heuristics exclude the scenarios where the time-optimal trajectory within a pair of corridors does not involve an arc maneuver along an intermediate circumference. Considering such scenarios is subject of future work, as well as extending the heuristics such that a nearly optimal solution can be provided for a wider range of parameters in the problem formulation. For this reason, additional Monte Carlo simulations should be performed to validate the approach in an extended domain of scenarios, including large sequences of corridors.

Appendix

A. Proof of Lemma 3.1

Proof: The time $\boldsymbol{T}_{\mathcal{P}}$ required to move along a primitive $\mathcal{P} \in \{S, C^+, C^-, T^+, T^-\}$ is given by: (i) $\boldsymbol{T}_S = \frac{c_i}{v_{\max}}$ where d_i is the Euclidean distance between the starting point (x_i^s, y_i^s) and the ending point (x_i^e, y_i^e), (ii) $\boldsymbol{T}_{C^\bullet} = \frac{\iota_i R}{v_{\max}}$, and (iii) $\boldsymbol{T}_{T^\bullet} = \frac{\phi_i}{\omega_{\max}}$. Let us denote with the superscript $\bullet^*$ the variables corresponding to a solution that is assumed to be time-optimal. The total motion time without a turn on-the-spot maneuver is given by $\boldsymbol{T} \coloneqq \frac{\iota_1 R}{v_{\max}} + \frac{c}{v_{\max}} + \frac{\iota_2 R}{v_{\max}}$, while the time-optimal motion time is given by $\boldsymbol{T}^* \coloneqq \frac{\psi_0^* R}{v_{\max}} + \frac{\iota_1^* R}{v_{\max}} + \frac{c^*}{v_{\max}} + \frac{\iota^*{}_2 R}{v_{\max}}$. We want to prove that $\boldsymbol{T} > \boldsymbol{T}^*$ when the conditions proposed in Lemma 3.1 hold, meaning that a turn on-the-spot is required for time-optimality. Proving this inequality is equivalent to prove that $\iota_1 R + c + \iota_2 R > \psi_0 R + \iota^*{}_1 R + c^* + \iota^*{}_2 R$ which, in turn, is equivalent to

$$\tau_1 \alpha_1 R + c - \tau_2 \alpha_1 R > \tau_1 \alpha_1^* R + c^* - \tau_2 \alpha_1^* R, \qquad (3)$$

since, based on simple geometry, $\iota_1 = \tau_1(\alpha_1 - \theta_0)$, $\iota_2 = \tau_2(\theta_f - \alpha_1)$, $\psi_0^* + \iota_1^* = \tau_1(\alpha_1^* - \theta_0)$, and $\iota_2^* = \tau_2(\theta_f - \alpha_1^*)$. Recall that $\alpha_0 \coloneqq \arctan(y_{c_2} - y_0 / x_{c_2} - x_0)$, $x_{c_1} = x_0 + R\cos(\gamma_1)$, $y_{c_1} = y_0 + R\sin(\gamma_1)$, and $c^2 = (x_{c_2} - x_{c_1})^2 + (y_{c_2} - y_{c_1})^2$. Without loss of generality assume $(x_0, y_0) = (0, 0)$.

If $\tau_1 = \tau_2$, Inequality (3) leads to the inequality $c > c^*$. Since $c, c^* \in \mathbb{R}_{>0}$, $c > c^* \equiv c^2 > c^{*2}$. As we assume that $\gamma_1^* = \alpha_0$ if $\tau_1 = \tau_2$ and $|\boldsymbol{\delta_\theta}(\theta_0, \alpha_0)| > \pi/2$, in this case the center (x_{c_2}, y_{c_2}) of $\mathcal{O}_2$ can be described as $x_{c_2} = (R + c^*)\cos(\gamma_1^*)$ and $y_{c_2} = (R + c^*)\sin(\gamma_1^*)$.

Thus, $c^2 > c^{*2} \equiv (R + c^*)(1 - \cos(\gamma)\cos(\alpha_0) - \sin(\gamma)\sin(\alpha_0)) > 0$. Since $R + c^* > 0$, $c^2 > c^{*2}$ holds if $\cos(\gamma)\cos(\alpha_0) + \sin(\gamma)\sin(\alpha_0) < 1$, leading to $c > c^*\ \forall \gamma \neq \alpha_0$. Therefore, $\gamma_1 = \alpha_0$ leads to a time-optimal trajectory with an initial turn on-the-spot maneuver, and an arc maneuver with $\iota_1^* = \pi/2$. ■

References

[1] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics*. Springer London, 2009.

[2] J. Z. Ben-Asher and E. D. Rimon, "Time optimal trajectories for a car-like mobile robot," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 421–432, Feb. 2022.

[3] S. Kousik, S. Vaskov, M. Johnson-Roberson, and R. Vasudevan, "Safe trajectory synthesis for autonomous driving in unforeseen environments," in *ASME 2017 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers, Oct. 2017.

[4] C. Liu, S. Lee, S. Varnhagen, and H. E. Tseng, "Path planning for autonomous vehicles using model predictive control," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Jun. 2017.

[5] D. Mengoli, "A robotic platform for precision agriculture and applications," Ph.D. dissertation, Università di Bologna, Jun. 2023.

[6] M. Reske, M. Daher, M. Hötter, and C. Brockmeier, "Last-mile robot for urban goods transport," *ATZ worldwide*, vol. 123, no. 12, pp. 42–47, Nov. 2021.

[7] X. Huang, Q. Cao, and X. Zhu, "Mixed path planning for multi-robots in structured hospital environment," *The Journal of Engineering*, vol. 2019, no. 14, pp. 512–516, Feb. 2019.

[8] P. Prabhu and A. R. Chowdhury, "Feasibility study of multi autonomous mobile robots (AMRs) motion planning in smart warehouse environment," in *2021 18th International Conference on Ubiquitous Robots (UR)*. IEEE, Jul. 2021.

[9] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, no. 3, p. 497, Jul. 1957.

[10] L. Pontryagin, *Mathematical Theory of Optimal Processes*, ser. Classics of Soviet Mathematics. Taylor & Francis, 1987.

[11] H. H. Johnson, "An application of the maximum principle to the geometry of plane curves," *Proceedings of the American Mathematical Society*, vol. 44, no. 2, pp. 432–435, 1974.

[12] Y. Zheng, Z. Chen, X. Shao, and W. Zhao, "Time-optimal guidance for intercepting moving targets by dubins vehicles," *Automatica*, vol. 128, p. 109557, Jun. 2021.

[13] Z. Fathi, B. Bidabad, and M. Najafpour, "Time optimal control in presence of moving obstacles for a dubins airplane," 2019.

[14] K. Kučerová, P. Váň, and J. Faigl, "On finding time-efficient trajectories for fixed-wing aircraft using dubins paths with multiple radii," in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*. ACM, Mar. 2020.

[15] T. Mercy, E. Hostens, and G. Pipeleers, "Online motion planning for autonomous vehicles in vast environments," in *2018 IEEE 15th International Workshop on Advanced Motion Control (AMC)*. IEEE, Mar. 2018.

[16] S. G. Manyam, D. Casbeer, A. L. V. Moll, and Z. Fuchs, "Shortest dubins path to a circle," in *AIAA Scitech 2019 Forum*. American Institute of Aeronautics and Astronautics, Jan. 2019.