Motion-Planning for Unicycles using the Invariant-Set Motion-Planner

Danielson, Claus; Berntorp, Karl; Di Cairano, Stefano; Weiss, Avishai

TR2020-089 July 01, 2020

Abstract

This paper adapts the invariant-set motion-planner to systems with unicycle-like dynamics. The invariant-set motion-planner is a motion-planning algorithm that uses the positive-invariant sets of the closed-loop dynamics to find a collision-free path to a desired target through an obstacle filled environment. The main challenge in applying the invariant-set motion-planner to unicycles is that the positive invariant sets of the unicycle under discontinuous feedback control have complex geometry. Thus, we develop numerically efficient mathematical tools for detecting collisions. We demonstrate the invariant-set motion-planner for unicycles in an automated perpendicular parking case study.

American Control Conference (ACC)

Motion-Planning for Unicycles using the Invariant-Set Motion-Planner

Claus Danielson, Karl Berntorp, Stefano Di Cairano, Avishai Weiss

***Abstract*—This paper adapts the invariant-set motion-planner to systems with unicycle-like dynamics. The invariant-set motion-planner is a motion-planning algorithm that uses the positive-invariant sets of the closed-loop dynamics to find a collision-free path to a desired target through an obstacle filled environment. The main challenge in applying the invariant-set motion-planner to unicycles is that the positive invariant sets of the unicycle under discontinuous feedback control have complex geometry. Thus, we develop numerically efficient mathematical tools for detecting collisions. We demonstrate the invariant-set motion-planner for unicycles in an automated perpendicular parking case study.**

I. Introduction

The invariant-set motion-planner (ISMP) is an algorithm for generating dynamically feasible trajectories from an initial state to a target state through an obstacle-filled environment [1]–[6]. Like many other motion-planning algorithms, the ISMP abstracts the motion-planning problem as a graph search. The defining feature of the ISMP is that knowledge of the closed-loop system dynamics is incorporated into the search graph using obstacle-free positive invariant (PI) sets. These sets describe regions of the state-space where the system can safely track the corresponding references. Thus, the ISMP can safely omit collision checks for the fine-scale trajectories of the system inside these sets. The coarse motion of the system is bisimulated by the edges of the graph, which indicate that the system will enter another safe set without leaving the current safe set. Thus, the ISMP finds a corridor of safe sets that safely guides the system through the obstacle filled environment to the target state, eliminating the need for expensive collision checks.

The ISMP has several advantageous properties. It allows for aggressive but safe maneuvers since the system state will never leave the safe PI sets by construction. It is inherently robust since it incorporates feedback into the design and the PI sets provide a natural buffer that can absorb tracking errors due to model uncertainty and disturbances [6]. It does not require dense sampling since the PI sets can cover large volumes of the state/output-space. It reduces the curse-of-dimensionality by sampling from the output-space instead of the state-space (although for the unicycle considered in this paper the state-space and output-space are identical). It typically has low online computational costs since the PI sets can be pre-computed as they only depend on the time-invariatn closed-loop dynamics, rather than the time varying environment.

This paper applies the ISMP to systems with unicycle dynamics. The unicycle model is a kinematic vehicle model widely used to describe the behavior of mobile robots and other non-holonomic vehicles. We exploit Lyapunov-based controllers for unicycles [7]–[11], since the level-sets of their Lyaponov functions are PI sets that can be used by the ISMP. It is well known that the unicycle cannot be stabilized by smooth state-feedback in the Cartesian state-space [12]. Thus, unicycle controllers typically exploit the polar state-space representation of the dynamics [7]–[11]. The nonlinear and discontinuous transformations between the Cartesian and polar state-spaces are the source of the main challenge for implementing the ISMP for unicycles. To avoid obstacles, the level-sets of the Lyapunov function must be transformed into the Cartesian state-space in which the obstacles are described. Although the level-sets of the Lyapunov functions have beneficial geometry in the polar state-space, in the Cartesian state-space they are non-convex and scale nonlinearly with the parameters of the Lyapunov function. Thus, this paper derives numerically efficient methods for detecting whether these PI sets collide with obstacles. Furthermore, we provide methods for selecting the parameters of the PI sets to expand their volume while guaranteeing safety.

Motion-planning is a well studied area. One popular class of path-planners is sampling-based algorithms. Rapidly-exploring random trees (RRT) abstract the motion-planning problem as a graph search, where the graph vertices are points sampled from the obstacle-free region and the graph edges indicate collision-free geometric paths connecting these vertices [13]–[15]. Kinodynamic RRT [13], [14] and closed-loop RRT [16] are variants of RRT that construct the search graph edges by sampling control inputs and references, respectively, and then simulating the resulting motion of the system over a finite-time horizon. By construction, this produces dynamically feasible trajectories, provided that the model used in the simulations is correct and there are no disturbances. The works [6], [17]–[19] consider sampling-based motion-planning under uncertainty.

Recently, set-based motion-planning algorithms have been growing in popularity [20]–[26]. Like sampling-based algorithms, set-based algorithms abstract the motion-planning problem as a graph search. However, set-based algorithms sample subsets of the state-space or output-space, rather than just points. For the invariant-set motion-planners [1]–[6], the vertices of the graph index equilibrium states as well as a surrounding obstacle-free positive-invariant subset of the state-space, in which the closed-loop system is guaranteed to avoid collisions, i.e., the sampled set is safe. The edges indicate that it is possible to enter another safe-set without leaving the current safe-set. A similar concept is reachable-set based verification methods [20]–[22] in which an edge of the search graph indicates that the target vertex lies in an obstacle-free reachable-set of the current vertex. LQR-trees [23]–[26] is another example of set-based motion-planners. In [26] an edge is added to the search graph if a two-point boundary value problem can be solved to find a trajectory connecting a pair of vertices and a sum-of-squares program can be solved to find a full-dimensional invariant set around this trajectory. Model predictive control has also been used for motion-planning [27]–[29], but has high computational cost and lacks convergence guarantees due its formulation as a non-convex optimization problem.

This paper is organized as follows. In Section II, we define the motion-planning problem and briefly summarize the ISMP algorithm. In Section III, we adapt the ISMP to systems with unicycle dynamics. More specifically, in Section III-A, we present the closed-loop model of the unicycle dynamics. In Section III-B, we provide a numerically efficient method for performing collision checks. In Section III-C, we describe how to select the parameter of the PI sets to expand their volume while maintaining safety. In Section III-D, we describe how to connect the PI sets to form the edges of the search graph.

Finally, in Section IV, we demonstrate the ISMP for unicycles in a perpendicular parking case study.

Notation: Consider an autonomous dynamic system $\dot{x}=f(x)$. The notation $x(t) \to \bar{x}$ is shorthand for $\lim_{t\to\infty} x(t) = \bar{x}$. A set $\mathcal{O}$ is positive invariant if $x(t_0) \in \mathcal{O}$ implies $x(t) \in \mathcal{O}$ for all $t > t_0$. A (global) Lyapunov function $V(x)$ is a scalar continuously differentiable positive definite function that satisfies $\nabla V(x)^\intercal f(x) < 0$ for all $x \neq 0$. Level-sets $\{x : V(x) \leq l\}$ of Lyapunov functions are positive invariant. $\mathbb{S}^1$ and $\mathbb{SE}(2) = \mathbb{R}^2 \times \mathbb{S}^1$ denote the groups of planar rotations and planar translations and rotations, respectively (or, with a minor abuse of terminology, any isomorphic groups). The cone of a set $\mathcal{S}$ is defined as $\mathrm{cone}(\mathcal{S}) = \left\{ \left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right] : x \in y\mathcal{S}, y \geq 0 \right\}$. The dual of $\mathcal{S}$ is defined as $\mathcal{S}^\circ = \left\{(a,b) : ax + by \leq 0, \forall \left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right] \in \mathrm{cone}(\mathcal{S})\right\}$. A directed graph $\mathcal{G}=(\mathcal{I},\mathcal{E})$ is a set of vertices $\mathcal{I}$ together with a set of ordered pairs $\mathcal{E} \subseteq \mathcal{I} \times \mathcal{I}$ called edges. Vertices $i, j \in \mathcal{I}$ are called adjacent if $(i,j) \in \mathcal{E}$ is an edge. A path is a sequence of adjacent vertices. A graph search is an algorithm for finding a path through a graph. The planar rotation matrix is defined as

$$R(\psi) = \left[\begin{smallmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{smallmatrix}\right].$$

II. Background: Invariant-Set Motion-Planner

This paper applies the ISMP to systems with unicycle dynamics. In this section, we define the motion-planning problem for a generic nonlinear system and summarize the ISMP [1]–[6] used to solve this problem. In the next section, we will adapt the generic ISMP for the unicycle.

A. Motion-Planning Problem

The objective of the motion-planning problem is to plan the trajectory $s(t)$ of a dynamic system from an initial state $s(0) = s_0$ to a target state s_∞ while the position $p(t) = Cs(t)$ avoids $p(t) \notin \mathcal{B}_k$ obstacles $\{\mathcal{B}_k\}_{k\in\mathcal{K}}$ in the environment $\subset \mathbb{R}^{n_p}$. The trajetory $s(t)$ is produced by providing a sequence $\{\bar{s}_i\}_{i=1}^N$ of intermediate references $\bar{s}_i \in \mathbb{R}^{n_p}$, called a *path*, to a generic closed-loop nonlinear system

$$\dot{s}(t) = f(s(t), \bar{s}(t)) \tag{1a}$$
$$p(t) = Cs(t) \tag{1b}$$

where the position $p(t) = Cs(t)$ of the system is a linear function of its state $s(t)$. We assume that the closed-loop system (1) asymptotically tracks $s(t) \to \bar{s}(t) = \bar{s}_i$ constant reference $\bar{s}(t) = \bar{s}_i$ i.e. it is asymptotically stable with unitary dc-gain from the reference to the steady-state state.

The motion-planning problem is stated formally below.

Problem 1 (Motion-Planning)**.** *Construct a path $\{\bar{s}_i\}_{i=1}^N$ such that the resulting trajectory $s(t)$ of the closed-loop system* (1) *avoids obstacles $p(t) \notin \mathcal{B}_k$ for all $k \in \mathcal{K}$ and reaches the target state $s(t) \to \bar{s}_\infty$.*

B. Invariant-Set Motion-Planner

The ISMP described by Algorithm 1 can solve Problem 1. The ISMP abstracts motion-planning as the search for a path $\{\sigma_i\}_{i=0}^N \in \mathcal{I}$ through a graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$. The vertices $i \in \mathcal{I}$ of the graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ index reference states $\bar{s}_i$ that can be tracked by the closed-loop system (1) where the initial $\bar{s}_{\sigma_0} = s(0)$ and target $\bar{s}_{\sigma_N} = \bar{s}_\infty$ states are included $\sigma_0, \sigma_N \in \mathcal{I}$.

The defining feature of the ISMP is that knowledge of the closed-loop system (1) is incorporated into the graph $\mathcal{G}$ using PI sets. Associated with each vertex $i \in \mathcal{I}$ is an obstacle-free PI set $\mathcal{O}_i$. Each set $\mathcal{O}_i$ is safe since it is obstacle-free $C\mathcal{O}_i \cap \mathcal{B}_k = \varnothing$. Furthermore, since the set $\mathcal{O}_i$ is positive invariant, we know that the closed-loop

Algorithm 1 Invariant-Set Motion-Planner

1: use search graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ to find path $\{\bar{s}_{\sigma_0}, \dots, \bar{s}_{\sigma_N}\}$ from $\bar{s}_{\sigma_0} = s(0)$ to $\bar{s}_{\sigma_N} = \bar{s}_\infty$
2: set $k \leftarrow 0$
3: **repeat**
4:   **if** $s(t) \in \mathcal{O}_{\sigma_{k+1}}$ **then**
5:     $k \leftarrow k + 1$
6:   **end if**
7:   track current target state $\bar{s}(t) = \bar{s}_{\sigma_k}$
8: **until** $s(t) = \bar{s}_\infty$

system (1) will remain inside this safe set $s(t) \in \mathcal{O}_i$ as long as it is tracking the i-th reference $\bar{s}_i$. The edges $(i,j) \in \mathcal{E}$ of the graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ indicate that the system (1) will enter the j-th safe-set $\mathcal{O}_j$ while tracking the i-th vertex without leaving the current safe-set $\mathcal{O}_i$. Thus, the ISMP avoids obstacles by moving the system through a corridor of safe-sets $\mathcal{O}_{\sigma_i}$ for $\{\sigma_i\}_{i=0}^N$.

This paper applies the ISMP (Algorithm 1) to systems with unicycle dynamics. The main challenges is that the PI sets $\mathcal{O}_i$ of the unicycle have complex geometry. Thus, we develop numerically efficient mathematical tools for finding large regions $\mathcal{O}_i$ of the state-space that are verifiably safe $C\mathcal{O}_i \cap \mathcal{B}_k = \varnothing$ for each obstacle $k \in \mathcal{K}$.

III. Invariant-Set Motion-Planner for a Unicycle

In this section, we adapt the ISMP (Algorithm 1) for systems with unicycle dynamics.

A. Closed-Loop Unicycle Dynamics

The open-loop dynamics of a unicycle are modeled by the following nonlinear system [12]

$$\dot{x}(t) = v(t)\cos(\psi(t)) \tag{2a}$$
$$\dot{y}(t) = v(t)\sin(\psi(t)) \tag{2b}$$
$$\dot{\psi}(t) = \omega(t) \tag{2c}$$

where the state $s = (x, y, \psi) \in \mathbb{SE}(2)$ is comprised of the Cartesian-position $p = (x, y) \in \mathbb{R}^2$ and orientation (yaw) $\psi \in \mathbb{S}^1$ of the unicycle. The position $p = (x, y) \in \mathbb{R}^2$ and state $s = (x, y, \psi) \in \mathbb{SE}(2)$ are related $p = Cs$ by the matrix $C = \left[\begin{smallmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{smallmatrix}\right]$. The control inputs are the velocity $v \in \mathbb{R}$ and yaw-rate $\omega \in \mathbb{R}$.

Given a reference position $(\bar{x}_i, \bar{y}_i)$ and orientation $\bar{\psi}_i$, the unicycle dynamics (2) can also be expressed in the polar state-space

$$\dot{r}_i(t) = -v(t)\cos\alpha_i(t) \tag{3a}$$
$$\dot{\theta}_i(t) = -v(t)\tfrac{\sin\alpha_i(t)}{r(t)} \tag{3b}$$
$$\dot{\alpha}_i(t) = \omega(t) + v(t)\tfrac{\sin\alpha_i(t)}{r_i(t)} \tag{3c}$$

where $r_i \in \mathbb{R}$ and $\theta_i \in \mathbb{S}^1$ are the magnitude and angle, respectively, of the position error $(x-\bar{x}_i, y-\bar{y}_i)$ and $\alpha_i = \psi - \bar{\psi}_i - \theta_i \in \mathbb{S}^1$ is the heading angle error. The polar-state $(r_i, \theta_i, \alpha_i)$ is defined by the following nonlinear, discontinuous coordinate transformation

$$r_i = \sqrt{(x-\bar{x}_i)^2 + (y-\bar{y}_i)^2} \tag{4a}$$
$$\theta_i = \arctan(y-\bar{y}_i, x-\bar{x}_i) - \bar{\psi}_i \tag{4b}$$
$$\alpha_i = \psi - \arctan(y-\bar{y}_i, x-\bar{x}_i) \tag{4c}$$

where the range of the two-argument arc-tangent is $(-\pi, \pi]$.

The Cartesian-state (x, y, ψ) can be recovered using the inverse of the transformation (4),

$$x = \bar{x}_i + r_i \cos(\bar{\psi}_i + \theta_i) \tag{5a}$$
$$y = \bar{y}_i + r_i \sin(\bar{\psi}_i + \theta_i) \tag{5b}$$
$$\psi = \bar{\psi}_i + \theta_i + \alpha_i. \tag{5c}$$

The following controller from the literature [8] can be used to drive the unicycle (2) to the reference $(\bar{x}_i, \bar{y}_i, \bar{\psi}_i)$

$$v(t) = \quad k_r r_i(t) \cos\alpha_i(t) \tag{6a}$$
$$\omega(t) = -k_\alpha \alpha_i(t) - k_r \frac{\sin\alpha_i(t)\cos\alpha_i(t)}{\alpha_i(t)}(\alpha_i(t) - \theta_i(t)) \tag{6b}$$

where the gains $k_r \in \mathbb{R}_+$ and $k_\alpha \in \mathbb{R}_+$ tune the relative convergence rates of the magnitude of the position error (4a) and heading angle error (4c). The angle of the position error (4b) is irrelevant when the position error magnitude (4a) and heading angle error (4c) are zero. The unicycle moves in reverse $v < 0$ when the target is behind $|\alpha| > \frac{\pi}{2}$.

The following proposition shows that the controller (6) will drive $\big(x(t), y(t), \psi(t)\big) \to (\bar{x}_i, \bar{y}_i, \bar{\psi}_i)$ the unicycle (3) to any desired reference $(\bar{x}_i, \bar{y}_i, \bar{\psi}_i)$ from any initial condition $\big(x(0), y(0), \psi(0)\big)$.

Proposition 1. *The controller* (6) *asymptotically stabilizes the state* $(\bar{x}_i, \bar{y}_i, \bar{\psi}_i)$ *for the unicycle* (3).

Proof Sketch from [8]. The stability of the origin $(r_i, \theta_i, \alpha_i) = (0, 0, 0)$ under these closed-loop dynamics (3) and (6) can be verified using the Lyapunov function

$$V\big(r_i, \theta_i, \alpha_i\big) = \left(\frac{r_i}{p_r}\right)^2 + \left(\frac{\theta_i}{p_\theta}\right)^2 + \left(\frac{\alpha_i}{p_\theta}\right)^2 \tag{7}$$

for any $p_r, p_\theta > 0$. Asymptotic stability can be proven using LaSalle's invariance principle. Thus, $\big(x(t), y(t), \psi(t)\big) \to (\bar{x}_i, \bar{y}_i, \bar{\psi}_i)$ since the origin $(r_i, \theta_i, \alpha_i) = (0, 0, 0)$ of the polar state-space (4) corresponds to the reference state $(\bar{x}_i, \bar{y}_i, \bar{\psi}_i)$ in Cartesian state-space. □

A sketch of the proof of Proposition 1 was included since the Lyapunov function (7) is used throughout the paper.

This paper considers the motion-planning (Problem 1) for the nonlinear system (1) comprised of the unicycle (3) in closed-loop with the controller (6). The obstacles $\{\mathcal{B}_k\}_{k\in\mathcal{K}}$ are modeled by subsets $\mathcal{B}_k \subset \mathbb{R}^2$ of the Cartesian-plane $\mathbb{R}^2$ through which the unicycle moves. The objective of the unicycle motion-planning problem is to construct a path $\{(\bar{x}_\sigma, \bar{y}_\sigma, \bar{\psi}_\sigma)\}_{\sigma=0}^{N}$ that safely guides the closed-loop unicycle (3) and (6) from the initial state $\big(x(0), y(0), \psi(0)\big)$ to the target state $(\bar{x}_\infty, \bar{y}_\infty, \bar{\psi}_\infty)$ while avoiding obstacles $\big(x(t), y(t)\big) \notin \mathcal{B}_k$. For now, we will model the unicycle as a point and later we will consider when the unicycle has a rectangular shape.

B. Collision-free Invariant-Sets

In this section, we describe regions $\mathcal{O}_i \subset \mathbb{SE}(2)$ of the state-space $\mathbb{SE}(2)$ in which the closed-loop system (3) and (6) can safely track the i-th reference $(\bar{x}_i, \bar{y}_i, \bar{\psi}_i)$ without colliding with any obstacles $\mathcal{B}_k$ for $k \in \mathcal{K}$. Since finding the largest PI set $\mathcal{O}_i$ is a non-convex problem, we focus on finding a computationally tractable, rather than optimal, method for computing the PI sets $\mathcal{O}_i$.

The following proposition characterizes the PI sets of the closed-loop system (3) and (6).

Proposition 2. *The set*

$$\mathcal{O}_i = \begin{bmatrix} R(\bar{\psi}_i) & 0 \\ 0 & 1 \end{bmatrix} \mathcal{O}_0 + \begin{bmatrix} \bar{x}_i \\ \bar{y}_i \\ \bar{\psi}_i \end{bmatrix} \tag{8}$$

is a PI *set for the closed-loop unicycle* (3) *and* (6) *where* $\bar{s} = (\bar{x}_i, \bar{y}_i, \bar{\psi}_i) \in \mathbb{SE}(2)$ *and* $\mathcal{O}_0 \in \mathbb{SE}(2)$ *is the* PI *set* (8) *corresponding to the origin* $(\bar{x}, \bar{y}, \bar{\psi}) = (0, 0, 0)$

$$\mathcal{O}_0 = \left\{ \begin{bmatrix} r\cos\theta \\ r\sin\theta \\ \theta + \alpha \end{bmatrix} : \frac{r_i^2}{p_r^2} + \frac{\theta_i^2}{p_\theta^2} + \frac{\alpha_i^2}{p_\theta^2} \le \ell^2 \right\}. \tag{9}$$

Proposition 2 says that the PI set corresponding to the i-th reference $(\bar{x}_i, \bar{y}_i, \bar{\psi}_i)$ can be obtained by rotating and translating (8) the base PI set $\mathcal{O}_0$ corresponding to the origin $(\bar{x}, \bar{y}, \bar{\psi}) = (0, 0, 0)$. The parameters $p_r, p_\theta, \ell > 0$ can be chosen to shape the base invariant-set $\mathcal{O}_0(p_r, p_\theta, \ell)$, and thus any other PI sets $\mathcal{O}_i$.

Since the controller (6) will keep the unicycle (3) inside the PI set (8), we can prevent collisions by ensuring that the PI set does not intersect an obstacle. More precisely, the projection $C\mathcal{O}_i \subset \mathbb{R}^2$ of the PI set $\mathcal{O}_i \subset \mathbb{SE}(2)$ onto the Cartesian-plane $\mathbb{R}^2$ does not intersect $C\mathcal{O}_i \cap \mathcal{B}_k = \varnothing$ any obstacles $\mathcal{B}_k \subset \mathbb{R}^2$.

Fig. 1 shows the projection $C\mathcal{O}_0 \subset \mathbb{R}^2$ of the origin PI set (9) with parameters $p_r = 1$ and $p_\theta = \frac{\pi}{4}$. The parameter $p_r = 1$ means that the level-sets have length ℓ i.e. $|x| \le \ell$. For $\ell = 1$, the parameter p_r determines the length of the set $\mathcal{O}_0$ i.e. $|x| = r \le p_r$ for $\theta = 0$. The parameter p_θ determines the angle the set makes near the origin $r \approx 0$. Increasing the scaling $\ell > 1$ increases the length and angle of the set.

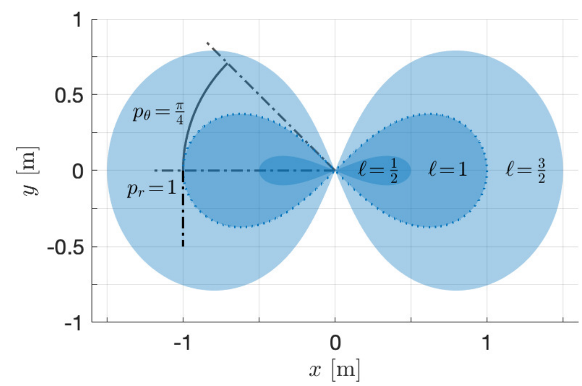


Fig. 1: Projection $C\mathcal{O}_0 \subset \mathbb{R}^2$ of the PI set $\mathcal{O}_0 \subset \mathbb{SE}(2)$ for different levels $\ell > 0$.

The non-convexity of the projected PI sets $C\mathcal{O}_i$ complicates the collision checking $C\mathcal{O}_i \cap \mathcal{B}_k = \varnothing$. To remedy this issue, we outer-approximate the set $C\mathcal{O}_i$ using a convex set. The following proposition outer-approximates the projected PI set $C\mathcal{O}_i$ by a rectangle.

Proposition 3. *The* PI *set* (8) *satisfies*

$$C\mathcal{O}_i \subset R(\bar{\psi}_i)\mathcal{R} + \begin{bmatrix} \bar{x}_i \\ \bar{y}_i \end{bmatrix} \tag{10}$$

where $\mathcal{R} \subset \mathbb{R}^2$ *is a rectangle defined by*

$$\mathcal{R} = \left\{ \begin{bmatrix} x \\ y \end{bmatrix} : \begin{array}{l} |x| \le p_r \ell \\ |y| \le \frac{1}{2} p_r p_\theta \ell^2 \end{array} \right\}. \tag{11}$$

Proposition 3 says that we can approximate the projected PI sets $C\mathcal{O}_i$ by translating and rotating (10) a rectangle (11), which is parametrized by the parameters p_r, p_θ, ℓ of the Lyapunov function (7). Thus, we can efficiently (but conservatively) guarantee that there are no collisions $C\mathcal{O}_i \cap \mathcal{B}_k = \varnothing$ by checking whether the obstacles $\mathcal{B}_k$ intersect $\mathcal{R}(\bar{x}_i, \bar{y}_i, \bar{\psi}_i) \cap \mathcal{B}_k = \varnothing$ a rectangle $\mathcal{R}(\bar{x}_i, \bar{y}_i, \bar{\psi}_i) = R(\bar{\psi}_i)\mathcal{R} + \left[\begin{smallmatrix}\bar{x}_i \\ \bar{y}_i\end{smallmatrix}\right]$.

C. Scaling the Invariant-Sets

Next, we consider how to select the parameters p_r, p_θ, ℓ of the invariant-sets (8) to increase their volume, while still ensuring that they are collision-free. There are several advantages to having large PI sets. Larger PI sets cover more of the state-space $\mathbb{SE}(2)$ meaning that the search graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ needs fewer vertices $|\mathcal{I}|$ to find a path to the target $(\bar{x}_\infty, \bar{y}_\infty, \bar{\psi}_\infty)$. Furthermore, the controller (6) is more aggressive inside larger PI set $\mathcal{O}_i$, which means that the unicycle will reach the target $(\bar{x}_\infty, \bar{y}_\infty, \bar{\psi}_\infty)$ more quickly.

Previously [1]–[6], the PI sets were scaled by taking different level-sets of the Lyapunov function. However, the PI sets (8) for the unicycle scale nonlinearly with the level ℓ of the Lyapunov function (7) i.e. different level-sets of the Lyapunov function (7) not only have different sizes, but also different shapes, see Fig. 1. This nonlinear scaling is also reflected in Proposition 3. The length $p_r\ell$ of the rectangle (11) grows linearly with the level ℓ, while the width $\frac{1}{2}p_r p_\theta \ell^2$ grows quadratically. Thus, for the unicycle we fix the level $\ell = 1$ and scale the PI sets (8) using the parameter $p_r > 0$ which effects both the length and width linearly. The choice of $p_\theta > 0$ is discussed later in this section.

The following theorem shows that the maximum scaling p_r can be determined by solving a convex optimization problem.

Theorem 1. *Let the obstacle $\mathcal{B}_k \subset \mathbb{R}^2$ be convex. The PI set (8) corresponding to the i-th reference $(\bar{x}_i, \bar{y}_i, \bar{\psi}_i)$ does not intersect $C\mathcal{O}_i \cap \mathcal{B}_k = \varnothing$ the obstacle $\mathcal{B}_k$ if*

$$p_{rik} \leq \max_{a,b} \quad a_x\bar{x}_i + a_y\bar{y}_i - b \tag{12a}$$

$$\text{s.t.} \quad (a, b) \in \mathcal{B}_k^\circ \tag{12b}$$

$$\left\| \begin{bmatrix} 1 & 0 \\ 0 & {p_\theta}/{2} \end{bmatrix} R(-\bar{\psi}_i) a \right\|_1 = 1 \tag{12c}$$

where p_θ is fixed, $\mathcal{B}_k^\circ$ is the dual of the set $\mathcal{B}_k$, and $\|\cdot\|_1$ is the 1-norm.

Proof. The largest safe scaling p_r of the rectangle $\mathcal{R}(p_r, p_\theta)$ would be the solution of

$$\begin{aligned} \max \quad & p_r \\ \text{s.t.} \quad & \left(R(\bar{\psi}_i)\mathcal{R}(p_r, p_\theta) + \left[\begin{smallmatrix}\bar{x}_i \\ \bar{y}_i\end{smallmatrix}\right] \right) \cap \mathcal{B}_k = \varnothing \end{aligned}$$

where $p_\theta > 0$ is fixed and $C\mathcal{O}_i \subset R(\bar{\psi}_i)\mathcal{R}(p_r, p_\theta) + \left[\begin{smallmatrix}\bar{x}_i \\ \bar{y}_i\end{smallmatrix}\right]$ according to Proposition 3. Since the rectangle (11) and obstacles $\mathcal{B}_k$ are both convex, the optimization problem above is equivalent to finding a separating hyper-plane for the sets $\mathcal{R}(p_r, p_\theta)$ and $\mathcal{B}_k$

$$\begin{aligned} \max \quad & p_r \\ \text{s.t.} \quad & \mathcal{B}_k \subseteq \mathcal{H}(a, b) \\ & R(\bar{\psi}_i)\mathcal{R}(p_r, p_\theta) + \left[\begin{smallmatrix}\bar{x}_i \\ \bar{y}_i\end{smallmatrix}\right] \subseteq \mathcal{H}^c(a, b) \end{aligned}$$

where $\mathcal{H}^c(a, b) = \{(x, y) : a_x x + a_y y > b\}$ is the complement of the half-space $\mathcal{H}(a, b) = \{(x, y) : a_x x + a_y y \leq b\}$. Note that the condition $\mathcal{B}_k \subseteq \mathcal{H}(a, b)$ is equivalent to (12b).

By convexity, $R(\bar{\psi}_i)\mathcal{R}(p_r, p_\theta) + \left[\begin{smallmatrix}\bar{x}_i \\ \bar{y}_i\end{smallmatrix}\right] \subseteq \mathcal{H}^c(a, b)$ holds if and only if it holds at each of the vertices of the rectangle (11)

$$a\left(\begin{bmatrix} \bar{x}_i \\ \bar{y}_i \end{bmatrix} + R(\bar{\psi}_i) \begin{bmatrix} \pm 1 \\ \pm \frac{p_\theta}{2} \end{bmatrix} p_r \right) \geq b.$$

Rearranging terms yields

$$a_x\bar{x}_i + a_y\bar{y}_i - b \geq aR(\bar{\psi}_i) \begin{bmatrix} 1 & 0 \\ 0 & {p_\theta}/{2} \end{bmatrix} \begin{bmatrix} \pm 1 \\ \pm 1 \end{bmatrix} p_r$$

which must hold for each of the four combinations of signs ($\pm$) that correspond to the four vertices of (11). For the worst-case choice of signs ($\pm$)

$$a_x\bar{x}_i + a_y\bar{y}_i - b \geq \left\| \begin{bmatrix} 1 & 0 \\ 0 & {p_\theta}/{2} \end{bmatrix} R(-\bar{\psi}_i) a^\intercal \right\|_1 p_r \tag{13}$$

where $p_r > 0$ and $R(\bar{\psi}_i)^\intercal = R(-\bar{\psi}_i)$. Since half-spaces are scale invariant (i.e. $\mathcal{H}(a, b) = \mathcal{H}(\mu a, \mu b)$, $\forall \mu > 0$), we can select $a \in \mathbb{R}^2$ such that the 1-norm has unit length i.e. (12c) holds without loss of generality. Then, the cost (12a) directly follows from (12c) and the bound (13). $\square$

Theorem 1 means that we can guarantee that the PI set $\mathcal{O}_i$ does not collide $C\mathcal{O}_i \cap \mathcal{B}_k = \varnothing$ with the k-th obstacle $\mathcal{B}_k$ by solving a convex optimization problem. Thus, we can guarantee that the PI set $\mathcal{O}_i$ is safe by taking worst-case scaling $p_{ri} = \min_{k \in \mathcal{K}} p_{rik}$ over all the obstacles $\{\mathcal{B}_k\}_{k \in \mathcal{K}}$.

For polyhedral obstacles $\mathcal{B}_k$, the convex program (12) reduces to a linear program, as shown in the following corollary.

Corollary 1. *Let $\mathcal{B}_k = \{(x, y) : A(x, y) \leq b\}$. Then PI set (8) corresponding to the i-th reference $(\bar{x}_i, \bar{y}_i, \bar{\psi}_i)$ does not intersect $C\mathcal{O}_i \cap \mathcal{B}_k = \varnothing$ the obstacle $\mathcal{B}_k$ if*

$$p_{ri} \leq \max_{z \geq 0} \quad \big(A(\bar{x}_i, \bar{y}_i) - b\big)^\intercal z \tag{14a}$$

$$\text{s.t.} \quad \left\| \begin{bmatrix} 1 & 0 \\ 0 & {p_\theta}/{2} \end{bmatrix} R(-\bar{\psi}_i) A^\intercal z \right\|_1 = 1. \tag{14b}$$

Although linear programs can be solved efficiently, solving (14) for hundreds or thousands of references $(\bar{x}_i, \bar{y}_i, \bar{\psi}_i)$ would be computationally prohibitive. Thus, we use a heuristic to approximate (14). Any feasible solution of the optimization problem (12) provides a safe scaling $p_{ri} = \min_{k \in \mathcal{K}} p_{rik}$ of the PI set $\mathcal{O}_i(p_{ri}, p_\theta)$. In other words, the scaling

$$p_{rik} = \min_{k \in \mathcal{K}} \frac{a_x\bar{x}_i + a_y\bar{y}_i - b}{\left\| \left[\begin{smallmatrix} 1 & 0 \\ 0 & p_\theta/2 \end{smallmatrix}\right] R(-\bar{\psi}) a \right\|_1}$$

is safe for any $(a, b) \in \mathcal{B}_k^\circ$ where a_x and a_y are the x and y components of a, respectively. Thus, our heuristic approximates (14) by evaluating the suboptimal solution above for a finite set of pre-selected hyper-planes $(a_j, b_j) \in \mathcal{B}_k^\circ$

$$p_{ri} = \min_{k \in \mathcal{K}} \max_j \frac{a_{xj}\bar{x}_i + a_{yj}\bar{y}_i - b_j}{\left\| \left[\begin{smallmatrix} 1 & 0 \\ 0 & p_\theta/2 \end{smallmatrix}\right] R(-\bar{\psi}) a \right\|_1}. \tag{15}$$

For polyhedral obstacles $\mathcal{B}_k$ a natural choice for the hyper-planes are the hyper-planes that define the obstacle set.

Next, we will use the parameter $p_\theta > 0$ to account for the spatial extent of the unicycle. We will assume that the spatial extent of the unicycle can be covered by a rectangle

$$\mathcal{B}_0 = \left\{ \begin{bmatrix} x \\ y \end{bmatrix} : \begin{matrix} |x| \leq l/2 \\ |y| \leq w/2 \end{matrix} \right\} \tag{16}$$

with length $l > 0$ and width $w > 0$. The following theorem shows that we can combine the scaling of the rectangle (11) used to outer-approximate the PI sets (8) with the collision check for a rectangular (16) unicycle into a single operation.

Theorem 2. *Let p_{ri} satisfy* (12) *and $p_\theta = {}^{2w}/_{l}$. Then, the* PI *set $\mathcal{O}_i = R(\bar{\psi}_i)\mathcal{O}_0(p_{ri} - {}^{l}/_{2}, p_\theta) + \left[\begin{smallmatrix}\bar{x}_i\\ \bar{y}_i\end{smallmatrix}\right]$ is safe $C\mathcal{O}_i \oplus R(\psi)\mathcal{B}_0 \cap \mathcal{B}_k = \varnothing$ for all $k \in \mathcal{K}$.*

Proof. The PI set $\mathcal{O}_i(p_r - {}^{l}/_{2}, p_\theta)$ is safe if, for each possible position $(x, y) \in C\mathcal{O}_i$ of the unicycle, the unicycle body $\mathcal{B}_0$ does not intersect any obstacle $\mathcal{B}_k$

$$\left(R(\bar{\psi}_i)\mathcal{B}_0 + \left[\begin{smallmatrix}x\\ y\end{smallmatrix}\right]\right) \cap \mathcal{B}_k = \varnothing.$$

for all $(x, y) \in C\mathcal{O}_0$ and $k \in \mathcal{K}$. Or equivalently

$$\Big(R(\bar{\psi}_i)\mathcal{B}_0 \oplus \big(R(\bar{\psi}_i)C\mathcal{O}_0 + \left[\begin{smallmatrix}\bar{x}_i\\ \bar{y}_i\end{smallmatrix}\right]\big)\Big) \cap \mathcal{B}_k = \varnothing$$

where $\oplus$ is the Minkowski sum. Note that

$$R(\bar{\psi}_i)\mathcal{B}_0 \oplus \Big(R(\bar{\psi}_i)C\mathcal{O}_0 + \left[\begin{smallmatrix}\bar{x}_i\\ \bar{y}_i\end{smallmatrix}\right]\Big) = \\ = R(\bar{\psi}_i)\Big(\underbrace{\bigcup_{p \in \mathcal{B}_0} C\mathcal{O}_0 + p}_{C\mathcal{O}_0 \oplus \mathcal{B}_0}\Big) + \left[\begin{smallmatrix}\bar{x}_i\\ \bar{y}_i\end{smallmatrix}\right].$$

For $p_\theta = {}^{2w}/_{l}$ we have $C\mathcal{O}_0 \subset \mathcal{R}(p_r - {}^{l}/_{2}, {}^{2w}/_{l})$. Thus, $C\mathcal{O}_0 \oplus \mathcal{B}_0 \subset \mathcal{R}(p_r - {}^{l}/_{2}, {}^{2w}/_{l}) \oplus \mathcal{B}_0 = \mathcal{R}(p_r, p_\theta)$. Therefore,

$$R(\bar{\psi}_i)(C\mathcal{O}_0 \oplus \mathcal{B}_0) + \left[\begin{smallmatrix}\bar{x}_i\\ \bar{y}_i\end{smallmatrix}\right] \subset R(\bar{\psi}_i)\mathcal{R}(p_r, p_\theta) + \left[\begin{smallmatrix}\bar{x}_i\\ \bar{y}_i\end{smallmatrix}\right].$$

Since p_{ri} satisfies (12), we have $\big(R(\bar{\psi}_i)\mathcal{R}(p_r, p_\theta) + \left[\begin{smallmatrix}\bar{x}_i\\ \bar{y}_i\end{smallmatrix}\right]\big) \cap \mathcal{B}_k = \varnothing$ for all $k \in \mathcal{K}$ by Theorem 1. $\square$

Theorem 2 says that if the rectangle (11) bounding the PI set and the rectangle $\mathcal{B}_0$ bounding the unicycle have the same aspect ratio ${}^{2p_r}/_{p_r p_\theta} = {}^{l}/_{w}$ then we can guarantee that no collision occurs by shrinking $p_{ri} - {}^{l}/_{2}$ the PI sets (8) by half the length ${}^{l}/_{2}$ of the unicycle. If $p_{ri} < {}^{l}/_{2}$ then the reference $(\bar{x}_i, \bar{y}_i, \bar{\psi}_i)$ collides with an obstacle when the spatial-extend $\mathcal{B}_0$ of the unicycle is taken into account.

D. Connecting the Invariant Sets

Next, we describe how the PI sets $\mathcal{O}_i$ are used to construct the edge list $\mathcal{E}$ for the search graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$.

The edges $(i, j) \in \mathcal{E}$ of the search graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ indicate that the trajectory $(x(t), y(t), \psi(t))$ of the closed-loop unicycle (3) and (6) will enter the safe set $\mathcal{O}_j$ while tracking the i-th reference $(\bar{x}_i, \bar{y}_i, \bar{\psi}_i)$ without leaving the current safe set $\mathcal{O}_i$. Since $(x(t), y(t), \psi(t)) \to (\bar{x}_i, \bar{y}_i, \bar{\psi}_i)$ according to Proposition 1, the trajectory $(x(t), y(t), \psi(t))$ will enter the j-th invariant-set $\mathcal{O}_j$ if $(\bar{x}_i, \bar{y}_i, \bar{\psi}_i) \in \mathcal{O}_j$. From the definition (8) of the set $\mathcal{O}_j$ and the polar transformation (4), we have $(\bar{x}_i, \bar{y}_i, \bar{\psi}_i) \in \mathcal{O}_j$ if

$$\frac{\left(\bar{x}_i - \bar{x}_j\right)^2 + \left(\bar{y}_i - \bar{y}_j\right)^2}{p_{rj}^2} + \frac{\left(\bar{\psi}_j - \bar{\theta}_{ij}\right)^2}{p_\theta^2} + \frac{\left(\bar{\psi}_i - \bar{\theta}_{ij}\right)^2}{p_\theta^2} \leq 1 \tag{17}$$

where $\ell = 1$ for the PI set (8) and $\bar{\theta}_{ij}$ is the angle of the vector from $(\bar{x}_j, \bar{y}_j)$ to $(\bar{x}_i, \bar{y}_i)$

$$\bar{\theta}_{ij} = \arctan\left(\frac{\bar{y}_i - \bar{y}_j}{\bar{x}_i - \bar{x}_j}\right).$$

The first term of the connection rule (17) requires that the i-th and j-th references are close (relative to p_r) for the edge $(i, j) \in \mathcal{E}$ to be included in the search graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$. The second and third terms of (17) require that the beginning $\bar{\psi}_i$ and final $\bar{\psi}_j$ orientations must be closely aligned with the straight-line path $(\bar{x}_j - \bar{x}_i, \bar{y}_j - \bar{y}_i)$ connecting the i-th and j-th references.

The search graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ is directed (i.e. $(i, j) \in \mathcal{E} \not\Rightarrow (j, i) \in \mathcal{E}$) since the connection rule (17) uses the scaling (15) parameter p_{rj} of the j-th invariant-set $\mathcal{O}_j$. This reflects the intuition that the path-planner will be more cautious when p_{rj} is small since this means an obstacle is nearby. In other words, it is easier to safely move away from an obstacle than towards one.

IV. Case Study: Perpendicular Parking

In this case study, the unicycle must perpendicular park in a crowded parking garage, shown in Fig. 2. There are twelve obstacle sets $\{\mathcal{B}_k\}_{k=1}^{12}$. The first three sets $\mathcal{B}_1$, $\mathcal{B}_2$ and $\mathcal{B}_3$ represent the boundary of the parking garage. The remaining nine sets $\{\mathcal{B}_4\}_{k=4}^{12}$ represent parking spaces occupied by cars. The unicycle is initially at the entrance to the garage located at $(\bar{x}_0, \bar{y}_0) = (1.5, 17)$ meters and is pointed inside $\bar{\psi}_0 = -\frac{\pi}{2}$. We consider two target positions. The first is the open parking space located directly in front of the entrance at $(\bar{x}_\infty^1, \bar{y}_\infty^1) = (1.5, 3)$ meters. The unicycle must back into this parking space $\bar{\psi}_\infty^1 = +\frac{\pi}{2}$ radians. The second target is the open parking space located at $(\bar{x}_\infty^2, \bar{y}_\infty^2) = (10, 15)$ meters. Again the unicycle must back into this space $\bar{\psi}_\infty^2 = -\frac{\pi}{2}$ radians.

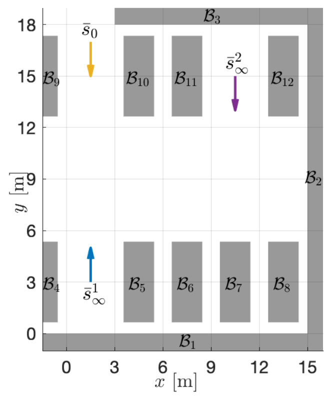


Fig. 2: Automated perpendicular parking scenario.

Reference states $\{(\bar{x}_i, \bar{y}_i, \bar{\psi}_i)\}_{i \in \mathcal{I}}$ were obtained by gridding the plane $\mathbb{R}^2$ with a resolution of 0.5 meters. Sixteen discrete orientations $\bar{\psi}_i \in \mathbb{S}^1$ were chosen

$$\arctan\Bigg\{\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}2\\1\end{bmatrix}, \begin{bmatrix}1\\1\end{bmatrix}, \begin{bmatrix}1\\2\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}-1\\2\end{bmatrix}, \begin{bmatrix}-1\\1\end{bmatrix}, \begin{bmatrix}-2\\1\end{bmatrix}, \\ \begin{bmatrix}-1\\0\end{bmatrix}, \begin{bmatrix}-2\\-1\end{bmatrix}, \begin{bmatrix}-1\\-1\end{bmatrix}, \begin{bmatrix}-1\\-2\end{bmatrix}, \begin{bmatrix}0\\-1\end{bmatrix}, \begin{bmatrix}1\\-2\end{bmatrix}, \begin{bmatrix}1\\-1\end{bmatrix}, \begin{bmatrix}2\\-1\end{bmatrix}\Bigg\} \tag{18}$$

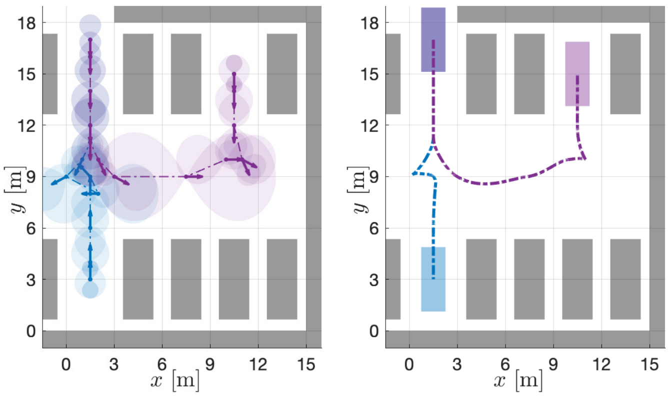
where $\arctan(\left[\begin{smallmatrix}x\\ y\end{smallmatrix}\right])$ is the angle of the vector $\left[\begin{smallmatrix}x\\ y\end{smallmatrix}\right]$. The orientations (18) were not sampled uniformly, but rather selected to align with paths between grid-points.

For each reference $\{(\bar{x}_i, \bar{y}_i, \bar{\psi}_i)\}_{i \in \mathcal{I}}$, the corresponding PI set (8) was scaled (15) such that it does not collide with any obstacle $C\mathcal{O}_i \oplus \mathcal{B}_0 \cap \mathcal{B}_k = \varnothing$. Scaling the $|\mathcal{I}| = 2,268$ PI sets required 1.3 seconds. The references $\{(\bar{x}_i, \bar{y}_i, \bar{\psi}_i)\}_{i \in \mathcal{I}}$ were connected $(i, j) \in \mathcal{E}$ using (17). Constructing the $|\mathcal{E}| = 42,760$ edges required 1.9 seconds. The prototype code was executed on a 2014 MacBook Pro with a 2.5 GHz i7 processor and 16 GB of RAM. No effort was made to optimize the execution time of the prototype code.

The graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ was searched for two paths $\sigma_0^{1,2}, \dots, \sigma_N^{1,2} \in \mathcal{I}$ from the invariant-set $\mathcal{O}_{\sigma_0}$ corresponding to the initial state $s_0 = (x_0, y_0, \psi_0)$ to the PI sets $\mathcal{O}_{\sigma_{N^1}}$ and $\mathcal{O}_{\sigma_{N^2}}$ corresponding to the two target states $\bar{s}_\infty^{1,2} = (\bar{x}_\infty^{1,2}, \bar{y}_\infty^{1,2}, \bar{\psi}_\infty^{1,2})$. The resulting paths $\{\bar{s}_{\sigma_i^{1,2}}\}_{i=1}^{N^{1,2}}$ are shown in Fig. 3a.

The first path $\{\sigma_i^1\}_{i=1}^{N^1}$ consists of $N^1 = 14$ references and has a total linear length of 17.9m. The second path $\{\sigma_i^2\}_{i=1}^{N^2}$ consists of

$N^1 = 15$ references that have a total linear length of 21.8m. This demonstrates that the ISMP does not require dense sampling of the state-space since the PI sets $\mathcal{O}_i$ cover large areas in which the motion does not need to be micro-managed to provide safety. Indeed, for the second path $\{\sigma_i^2\}_{i=1}^{N^2}$ the PI set (8) corresponding to the reference $(\bar{x}_i, \bar{y}_i, \bar{\psi}_i) = (7.5, 9, 0)$ has a radius $p_{ri} = 5.62\text{m}$ and covers an area of approximately 42.1m^2, which is approximately 22% of total obstacle-free area of 189.6m^2.



(a) Path

(b) Trajectory

Fig. 3: (a) Paths $\{\bar{s}_{\sigma_i^j}\}_{i=1}^{N^j}$ constructed by the ISMP and (b) the resulting trajectories $p(t) = \big(x(t), y(t)\big)$ of the unicycle.

The controller (6) was used to track the paths shown in Fig. 3a resulting in the trajectories $\big(x(t), y(t)\big)$ shown in Fig. 3b. Again, the trajectories $(x(t), y(t), \psi(t))$ of the unicycle (2) do not perfectly track their corresponding paths $\{(\bar{x}_{\sigma_k^{1,2}}, \bar{y}_{\sigma_k^{1,2}}, \bar{\psi}_{\sigma_k^{1,2}})\}_{k=1}^{N^{1,2}}$, but nonetheless, the unicycle does not collide with an obstacle $\mathcal{B}_k$ since the trajectory remains inside the corridor of PI sets $\mathcal{O}_{\sigma_k}$. Indeed, the trajectories are more curved in open spaces since the ISMP allows the unicycle to follow its own natural trajectories. In addition, larger PI sets allow the controller (6) to be more aggressive. For the second trajectory, the unicycle has a maximum velocity of 3.9m/s which occurs at the location $\big(x(t), y(t)\big) = (2.8, 9.0)$ where the unicycle enters the previously mentioned largest PI set. In contrast, after the unicycle has turned and is backing into the parking spot its maximum velocity is 2.1m/s which occurs at the location $\big(x(t), y(t)\big) = (10.5, 11.9)$ where the unicycle enters the first of 3 progressively smaller PI sets, each with a lower maximum velocity.

References

[1] A. Weiss, C. Petersen, M. Baldwin, R. Erwin, and I. Kolmanovsky, "Safe positively invariant sets for spacecraft obstacle avoidance," *Journal of Guidance, Control, and Dynamics*, 2015.

[2] C. Danielson, A. Weiss, K. Berntorp, and S. Di Cairano, "Path planning using positive invariant sets," in *Conference on Decision and Control*, 2016.

[3] A. Weiss, C. Danielson, K. Berntorp, I. Kolmanovsky, and S. Di Cairano, "Motion planning with invariant set trees," in *Conference on Control Technology and Applications*, 2017.

[4] K. Berntorp, A. Weiss, C. Danielson, S. Di Cairano, and I. Kolmanovsky, "Automated driving: Safe motion planning using positive-invariant sets," in *Intelligent Transportation Systems Conference*, 2017.

[5] K. Berntorp, R. Bai, K. F. Erliksson, C. Danielson, A. Weiss, and S. D. Cairano, "Positive invariant sets for safe integrated vehicle motion planning and control," *IEEE Transactions on Intelligent Vehicles*, 2020.

[6] C. Danielson, K. Berntorp, A. Weiss, and S. Di Cairano, "Robust motion-planning for uncertain systems with disturbances using the invariant-set motion-planner," *Accepted to Transactions on Automatic Control*, 2019.

[7] C. C. de Wit and O. J. Sordalen, "Exponential stabilization of mobile robots with nonholonomic constraints," *IEEE Transactions on Automatic Control*, 1992.

[8] M. Aicardi, G. Casalino, A. Bicchi, and A. Balestrino, "Closed loop steering of unicycle like vehicles via lyapunov techniques," *IEEE Robotics Automation Magazine*, 1995.

[9] A. Astolfi, "Exponential Stabilization of a Wheeled Mobile Robot Via Discontinuous Control," *Journal of Dynamic Systems, Measurement, and Control*, 1999.

[10] A. P. Aguiar, A. N. Atassi, and A. Pascoal, "Regulation of a nonholonomic dynamic wheeled mobile robot with parametric modeling uncertainty using lyapunov functions," *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No.00CH37187)*, 2000.

[11] F. Pourboghrat, "Exponential stabilization of nonholonomic mobile robots," *Computers and Electrical Engineering*, 2002.

[12] R. Brockett, "Asymptotic stability and feedback stabilization," *Differential Geometric Control Theory*, 1983.

[13] S. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[14] S. LaValle and J. Kuffner, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, 2001.

[15] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, 2011.

[16] J. Leonard, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, S. Karaman *et al.*, "A perception-driven autonomous urban vehicle," *Journal of Field Robotics*, 2008.

[17] B. D. Luders, S. Karaman, and J. P. How, "Robust sampling-based motion planning with asymptotic optimality guarantees," in *AIAA Guid., Nav., and Ctrl. Conf., Boston, MA*, 2013.

[18] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011.

[19] N. A. Melchior and R. Simmons, "Particle rrt for path planning with uncertainty," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 2007.

[20] M. Althoff, D. Althoff, D. Wollherr, and M. Buss, "Safety verification of autonomous vehicles for coordinated evasive maneuvers," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2010.

[21] M. Althoff and J. Dolan, "Online verification of automated road vehicles using reachability analysis," *IEEE Transactions on Robotics*, 2014.

[22] M. Koschi and M. Althoff, "SPOT: A tool for set-based prediction of traffic participants," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2017.

[23] O. Arslan, K. Berntorp, and P. Tsiotras, "Sampling-based algorithms for optimal motion planning using closed-loop prediction," in *International Conference on Robotics and Automation*, 2017.

[24] W. McConley, B. Appleby, M. Dahleh, and E. Feron, "A computationally efficient lyapunov-based scheduling procedure for control of nonlinear systems with stability guarantees," *Transactions on Automatic Control*, 2000.

[25] F. Blanchini, F. Pellegrino, and L. Visentini, "Control of manipulators in a constrained workspace by means of linked invariant sets," *Journal of Robust and Nonlinear Control*, 2004.

[26] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts, "LQR-trees: Feedback motion planning via sums-of-squares verification," *The International Journal of Robotics Research*, 2010.

[27] Y. Gao, A. Gray, H. Tseng, and F. Borrelli, "A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles," *Vehicle System Dynamics*, 2014.

[28] P. Falcone, F.Borrelli, J.Asgari, H. Tseng, and D.Hrovat, "Predictive active steering control for autonomous vehicle systems," *Transactions on Control Systems Technology*, 2007.

[29] S. Di Cairano, U. Kalabi, and K. Berntorp, "Vehicle tracking control on piecewise-clothoidal trajectories by mpc with guaranteed error bounds," *Conference on Decision and Control*, 2016.