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)
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 regions. 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, 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.

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 \( \nabla V(x)^T f(x) < 0 \) for all \( x \neq 0 \). Level-sets \( \{ x : V(x) \leq l \} \) of Lyapunov functions are positive invariant. \( \mathcal{S}^1 \) and \( \mathcal{S}^2 \) 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 \( \text{cone}(\mathcal{S}) = \{ (a,b) : a \in y \mathcal{S}, y \geq 0 \} \). The dual of \( \mathcal{S} \) is defined as \( \mathcal{S}^\top = \{ (a,b) : ax + by \leq 0, \forall [a,b] \in \text{cone}(\mathcal{S}) \} \).

A directed graph \( G = (\mathcal{I}, \mathcal{E}) \) 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

\[
\begin{align*}
R(\psi) &= \begin{bmatrix}
\cos \psi & -\sin \psi \\
\sin \psi & \cos \psi
\end{bmatrix}.
\end{align*}
\]

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 \( s_\infty \) such that the position \( p(t) = C s(t) \) avoids \( p(t) \notin \mathcal{B}_k \) obstacles \( \mathcal{B}_k \) \( k \in \mathcal{K} \) in the environment \( \mathcal{E} \) \( \subset \mathbb{R}^{n_p} \). The trajectory \( s(t) \) 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

\[
\begin{align*}
\dot{s}(t) &= f(s(t), \bar{s}(t)) & (1a) \\
p(t) &= C s(t) & (1b)
\end{align*}
\]

where the position \( p(t) = C s(t) \) of the system is a linear function of its state \( s(t) \). We assume that the closed-loop system (1) asymptotically tracks \( s(t) \to \bar{s}(t) \) \( \to \bar{s}_i \) 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 \( G = (\mathcal{I}, \mathcal{E}) \). The vertices \( i \in \mathcal{I} \) of the graph \( G = (\mathcal{I}, \mathcal{E}) \) 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 \( G \) 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 \( \mathcal{C} \mathcal{O}_i \cap \mathcal{B}_k = \emptyset \). Furthermore, since the set \( \mathcal{O}_i \) is positive invariant, we know that the closed-loop 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 \( G = (\mathcal{I}, \mathcal{E}) \) 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 \( \mathcal{O}_k \) for \( \{ \sigma_i \}_{i=0}^N \).

This paper applies the ISMP (Algorithm 1) to systems with unicycle dynamics. The main challenges is that knowledge of 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 \( \mathcal{C} \mathcal{O}_i \cap \mathcal{B}_k = \emptyset \) 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]

\[
\begin{align*}
\dot{x}(t) &= v(t) \cos(\psi(t)) & (2a) \\
\dot{y}(t) &= v(t) \sin(\psi(t)) & (2b) \\
\dot{\psi}(t) &= \omega(t) & (2c)
\end{align*}
\]

where the state \( s = (x,y,\psi) \in \mathbb{S}^2 \) 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 \( s = (x,y,\psi) \in \mathbb{S}^2 \) are related by \( p = C s \) by the matrix \( C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \). The control inputs are the velocity \( v \in \mathbb{R} \) and yaw-rate \( \omega \in \mathbb{R} \).

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

\[
\begin{align*}
\dot{r}(t) &= -v(t) \cos(\alpha_i(t)) & (3a) \\
\dot{\theta}(t) &= -v(t) \sin(\alpha_i(t)) \frac{\sin(\alpha_i(t))}{r_i(t)} & (3b) \\
\dot{\alpha}_i(t) &= \omega(t) + v(t) \frac{\sin(\alpha_i(t)) \sin(\alpha_i(t))}{r_i(t)} & (3c)
\end{align*}
\]

where \( r_i \in \mathbb{R} \) and \( \theta_i \in \mathbb{S}^1 \) are the magnitude and angle, respectively, of the position error \( (x-\bar{x}, y-\bar{y}) \) 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

\[
\begin{align*}
r_i &= \sqrt{(x-\bar{x})^2 + (y-\bar{y})^2} & (4a) \\
\theta_i &= \arctan(y-\bar{y}, x-\bar{x}) - \bar{\psi}_i & (4b) \\
\alpha_i &= \psi - \arctan(y-\bar{y}, x-\bar{x}) & (4c)
\end{align*}
\]

where the range of the two-argument arc-tangent is \( (-\pi, \pi] \).
The Cartesian-state \((x, y, \psi)\) can be recovered using the inverse of the transformation (4),

\[
\begin{align*}
x &= \bar{x} + r \cos(\bar{\psi} + \theta_i), \\
y &= \bar{y} + r \sin(\bar{\psi} + \theta_i), \\
\psi &= \bar{\psi} + \theta_i + \alpha_i.
\end{align*}
\]

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

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

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

\[
V(r, \theta, \alpha) = \left(\frac{r}{p_r}\right)^2 + \left(\frac{\theta}{p_\theta}\right)^2 + \left(\frac{\alpha}{p_\alpha}\right)^2
\]

for any \(p_r, p_\theta > 0\). Asymptotic stability can be proven using LaSalle’s invariance principle. Thus, \((x(t), y(t), \psi(t)) \to (\bar{x}, \bar{y}, \bar{\psi})\) since the origin \((r, \theta, \alpha) = (0, 0, 0)\) of the polar state-space (4) corresponds to the reference state \((\bar{x}, \bar{y}, \bar{\psi})\) 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 \(\{B_k\}_{k \in \mathcal{K}}\) are modeled by subsets \(B_k \subset \mathbb{R}^2\) 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 \(\{(\bar{x}, \bar{y}, \bar{\psi})\}_{t=0}^\infty\) that safely guides the closed-loop unicycle (3) and (6) from the initial state \((x(0), y(0), \psi(0))\) to the target state \((\bar{x}, \bar{y}, \bar{\psi})\) while avoiding obstacles \((x(t), y(t)) \not\in B_k\). 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 \((\bar{x}, \bar{y}, \bar{\psi})\) without colliding with any obstacles \(B_k\) 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 states \(\mathcal{O}_i\).

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

**Proposition 2.** The set

\[
\mathcal{O}_i = \left[\begin{array}{c}
R(\bar{\psi}_i) \\
0
\end{array}\right] + \left[\begin{array}{c}
\bar{x}_i \\
\bar{y}_i
\end{array}\right]
\]

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

\[
\mathcal{O}_0 = \left\{\begin{array}{c}
r \cos \theta \\
r \sin \theta \theta + \alpha
\end{array}\right\} : \frac{r^2}{p_r^2} + \frac{\theta^2}{p_\theta^2} + \frac{\alpha^2}{p_\alpha^2} \leq \ell^2
\]

Proposition 2 says that the PI set corresponding to the \(i\)-th reference \((\bar{x}, \bar{y}, \bar{\psi})\) can be obtained by rotating and translating (8) the base PI set \(\mathcal{O}_0\) corresponding to the origin \((x, y, \psi) = (0, 0, 0)\). 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 \(CO_i \subset \mathbb{R}^2\) of the PI set \(\mathcal{O}_i \subset \mathbb{SE}(2)\) onto the Cartesian-plane \(\mathbb{R}^2\) does not intersect \(CO_i \cap B_k = \emptyset\) any obstacles \(B_k \subset \mathbb{R}^2\).

**Fig. 1:** Projection \(CO_0 \subset \mathbb{R}^2\) of the origin PI set (9) with parameters \(p_r = 1\) and \(p_\theta = \frac{\pi}{2}\). The parameter \(p_r = 1\) means that the level-sets have length \(\ell\) i.e. \(|x| \leq \ell\). For \(\ell = 1\), the parameter \(p_r\) determines the length of the set \(\mathcal{O}_0\) i.e. \(|x| = r \leq p_r\) for \(\theta = 0\).

The parameter \(p_\theta\) 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.
that there are no collisions \( CO_i \cap B_k = \emptyset \) by checking whether the obstacles \( B_k \) intersect \( \mathcal{R}(\bar{x}_i, \bar{y}_i, \bar{\psi}_i) \cap B_k = \emptyset \) a rectangle \( \mathcal{R}(\bar{x}_i, \bar{y}_i, \bar{\psi}_i) = R(\bar{\psi}_i) \mathcal{R} + \binom{\bar{x}_i}{\bar{y}_i} \).

C. Scaling the Invariant-Sets

Next, we consider how to select the parameters \( p_r, p_u, \ell \) 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 \( \mathcal{E}(2) \) meaning that the search graph \( G = (\mathcal{T}, \mathcal{E}) \) needs fewer vertices \( |\mathcal{E}| \) 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 \( O_i \), 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 \( \ell \) 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 \( \ell \), while the width \( \frac{\sqrt{2}}{2}p_r p_u \ell^2 \) 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 \( p_u > 0 \) 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 \( B_k \subset \mathbb{R}^2 \) be convex. The PI set (8) corresponding to the \( i \)-th reference \( (\bar{x}_i, \bar{y}_i, \bar{\psi}_i) \) does not intersect \( CO_i \cap B_k = \emptyset \) the obstacle \( B_k \) if

\[
 p_{rik} \leq \max_{a,b} \quad a \bar{x}_i + a b \bar{y}_i - b \quad \text{s.t.} \quad (a, b) \in B_k^a
\]

\[
 \left\| \begin{bmatrix} 1 & 0 \\ 0 & p_u/2 \end{bmatrix} R(\bar{\psi}_i) a \right\|_1 = 1
\]

(12c)

where \( p_u \) is fixed, \( B_k^a \) is the dual of the set \( B_k \), and \( \| \cdot \|_1 \) is the 1-norm.

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

\[
 \max_{p_r} \quad p_r \\
 \text{s.t.} \quad \mathcal{R}(\bar{\psi}_i) \mathcal{R}(p_r, p_u) + \binom{\bar{x}_i}{\bar{y}_i} \cap B_k = \emptyset
\]

where \( p_u > 0 \) is fixed and \( CO_i \subset \mathcal{R}(\bar{\psi}_i) \mathcal{R}(p_r, p_u) + \binom{\bar{x}_i}{\bar{y}_i} \) according to Proposition 3. Since the rectangle (11) and obstacles \( B_k \) are both convex, the optimization problem above is equivalent to finding a separating hyper-plane for the sets \( \mathcal{R}(p_r, p_u) \) and \( B_k \)

\[
 \max_{p_r} \quad p_r \\
 \text{s.t.} \quad B_k \subseteq H(a,b) \\
 R(\bar{\psi}_i) \mathcal{R}(p_r, p_u) + \binom{\bar{x}_i}{\bar{y}_i} \subseteq H^c(a,b)
\]

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

By convexity, \( R(\bar{\psi}_i) \mathcal{R}(p_r, p_u) + \binom{\bar{x}_i}{\bar{y}_i} \subseteq H^c(a,b) \) holds if and only if it holds at each of the vertices of the rectangle (11)

\[
a \left( \begin{bmatrix} \bar{x}_i \\ \bar{y}_i \end{bmatrix} + R(\bar{\psi}_i) \begin{bmatrix} \pm 1 \\ \pm \frac{p_u}{2} \end{bmatrix} \right) \geq b.
\]

Rearranging terms yields

\[
a_x \bar{x}_i + a_y \bar{y}_i - b \geq a R(\bar{\psi}_i) \begin{bmatrix} 1 & 0 \\ 0 & p_u/2 \end{bmatrix} \begin{bmatrix} \pm 1 \\ \pm 1 \end{bmatrix} p_r
\]

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

\[
a_x \bar{x}_i + a_y \bar{y}_i - b \geq \left\| \begin{bmatrix} 1 & 0 \\ 0 & p_u/2 \end{bmatrix} R(\bar{\psi}_i) a \right\|_1 p_r
\]

(13)

where \( p_r > 0 \) and \( R(\bar{\psi}_i) a \) are the hyper-planes that define the obstacle set. Since half-spaces are scale invariant (i.e. \( H(a,b) = H(\mu a, \mu b), \forall \mu > 0 \)), 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).

Theorem 1 means that we can guarantee that the PI set \( O_i \) does not collide \( CO_i \cap B_k = \emptyset \) with the \( k \)-th obstacle \( B_k \) by solving a convex optimization problem. Thus, we can guarantee that the PI set \( O_i \) is safe by taking worst-case scaling \( p_{r_i} = \min_{k \in K} p_{rik} \) over all the obstacles \( \{B_k\}_{k \in K} \).

For polyhedral obstacles \( B_k \), the convex program (12) reduces to a linear program, as shown in the following corollary.

**Corollary 1.** Let \( B_k = \{(x,y) : A(x,y) \leq b\} \). Then PI set (8) corresponding to the \( i \)-th reference \( (\bar{x}_i, \bar{y}_i, \bar{\psi}_i) \) does not intersect \( CO_i \cap B_k = \emptyset \) the obstacle \( B_k \) if

\[
p_{r_i} \leq \max_{x \geq 0} \quad (A(\bar{x}_i, \bar{y}_i) - b)^\top z
\]

\[
\text{s.t.} \quad \left\| \begin{bmatrix} 1 & 0 \\ 0 & p_u/2 \end{bmatrix} R(\bar{\psi}_i) A^\top z \right\|_1 = 1.
\]

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 \( p_{r_i} = \min_{k \in K} p_{rik} \) of the PI set \( O_i(p_{r_i}, p_u) \). In other words, the scaling

\[
p_{rik} = \min_{k \in K} \quad a_x \bar{x}_i + a_y \bar{y}_i - b \quad \text{s.t.} \quad \left\| \begin{bmatrix} 1 & 0 \\ 0 & p_u/2 \end{bmatrix} R(\bar{\psi}_i) a \right\|_1 = 1.
\]

is safe for any \( (a,b) \in B_k^a \) 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 \( \{(a_j, b_j) \in B_k^a\} \). For polyhedral obstacles \( B_k \) a natural choice for the hyper-planes are the hyper-planes that define the obstacle set.

Next, we will use the parameter \( p_u > 0 \) 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

\[
B_0 = \left\{ \begin{bmatrix} x \\ y \end{bmatrix} : \begin{bmatrix} |x| & |y| \end{bmatrix} \leq \begin{bmatrix} l/2 \\ w/2 \end{bmatrix} \right\}
\]

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 \( p_0 = 2\omega_l/\ell \). Then, the PI set \( C_i = R(\bar{\psi}_i)C_0(p_{ri} - \frac{1}{2}, p_0) + [\bar{x}_i^j, y_i^j] \) is safe \( C_{O_i} \cap R(\psi)B_0 \cap B_k = \emptyset \) for all \( k \in K \).

Proof. The PI set \( C_i(p_{ri} - \frac{1}{2}, p_0) \) is safe if, for each possible position \((x, y) \in C_{O_i} \) of the unicycle, the unicycle body \( B_0 \) does not intersect any obstacle \( B_k \)

\[
\left( R(\bar{\psi}_i)B_0 + [\bar{x}_i^j, y_i^j] \right) \cap B_k = \emptyset
\]

for all \((x, y) \in C_{O_i} \) and \( k \in K \). Or equivalently

\[
\left( R(\bar{\psi}_i)B_0 \oplus R(\bar{\psi}_i)C_{O_i} + [\bar{x}_i^j, y_i^j] \right) \cap B_k = \emptyset
\]

where \( \oplus \) is the Minkowski sum. Note that

\[
R(\bar{\psi}_i)B_0 \oplus R(\bar{\psi}_i)C_{O_i} + [\bar{x}_i^j, y_i^j] = R(\bar{\psi}_i)\left( \bigcup_{p \in B_0} C_{O_i} + p \right) + [\bar{x}_i^j, y_i^j].
\]

For \( p_0 = 2\omega_l/\ell \) we have \( C_{O_i} \subseteq R(p_{ri} - \frac{1}{2}, 2\omega_l/\ell) \). Thus, \( C_{O_i} + B_0 \subseteq R(p_{ri} - \frac{1}{2}, 2\omega_l/\ell) + B_0 = R(p_{ri}, p_0) \). Therefore,

\[
R(\bar{\psi}_i)(C_{O_i} + B_0) + [\bar{x}_i^j, y_i^j] \subseteq R(\bar{\psi}_i)R(p_{ri}, p_0) + [\bar{x}_i^j, y_i^j].
\]

Since \( p_{ri} \) satisfies (12), we have \( R(\bar{\psi}_i)R(p_{ri}, p_0) + [\bar{x}_i^j, y_i^j] \cap B_k = \emptyset \) for all \( k \in K \) by Theorem 1.

Theorem 2 says that if the rectangle (11) bounding the PI set and the rectangle \( B_0 \) bounding the unicycle have the same aspect ratio \( 2p_{ri}/p_{ri}p_0 = 1/\ell \) then we can guarantee that no collision occurs by shrinking \( p_{ri} - \frac{1}{2} \) the PI sets (8) by half the length \( 1/2 \) of the unicycle. If \( p_{ri} < \frac{1}{2} \) then the reference \((\bar{x}_i^j, \bar{y}_i^j, \bar{\psi}_i)\) collides with an obstacle when the spatial-extend \( B_0 \) of the unicycle is taken into account.

D. Connecting the Invariant Sets

Next, we describe how the PI sets \( O_i \) are used to construct the edge list \( E \) for the search graph \( G = (I, E) \).

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

\[
\frac{(\bar{x}_i - \bar{x}_j)^2}{p_{ij}^2} + \frac{(\bar{y}_i - \bar{y}_j)^2}{p_{ij}^2} + \frac{(\bar{\psi}_i - \bar{\psi}_j)^2}{p_{ij}^2} \leq 1
\]

(17)

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

\[
\bar{\psi}_j = \arctan\left( \frac{\bar{y}_i - \bar{y}_j}{\bar{x}_i - \bar{x}_j} \right).
\]

The first term of the connection rule (17) requires that the \( i \)-th and \( j \)-th references are close (relative to \( p_{ij} \)) for the edge \((i, j) \in E \) to be included in the search graph \( G = (I, E) \). 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 \( G = (I, E) \) is directed (i.e., \((i, j) \in E \neq (j, i) \in E \)) since the connection rule (17) uses the scaling (15) parameter \( p_{ij} \) of the \( j \)-th invariant-set \( O_j \). This reflects the intuition that the path-planner will be more cautious when \( p_{ij} \) 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 \( \{ B_k \}_{k=1}^{12} \). The first three sets \( B_1, B_2 \) and \( B_3 \) represent the boundary of the parking garage. The remaining nine sets \( \{ B_k \}_{k=4}^{12} \) 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 \((\bar{x}^1_{\infty}, \bar{y}^1_{\infty}) = (1.5, 3) \) meters. The unicycle must back into this parking space \( \bar{\psi}^1_{\infty} = 0 \) rad. The second target is the open parking space located at \((\bar{x}^2_{\infty}, \bar{y}^2_{\infty}) = (10, 15) \) meters. Again the unicycle must back into this space \( \bar{\psi}^2_{\infty} = -\frac{\pi}{2} \) radians.

![Fig. 2: Automated perpendicular parking scenario.](image-url)
Indeed, the trajectories are more curved in open spaces since the trajectory remains inside the corridor of PI, but nonetheless, the unicycle does not collide with an obstacle B. In contrast, after the unicycle has turned and is backing into the parking spot its maximum velocity is, resulting in the trajectories of the unicycle. In ISMP, references that have a total linear length of 22m, which is approximately 22% of total obstacle-free area of 189.6m^2.

The controller (6) was used to track the paths shown in Fig. 3a resulting in the trajectories (x(t), y(t)) shown in Fig. 3b. Again, the trajectories (x(t), y(t), ψ(t)) of the unicycle (2) do not perfectly track their corresponding paths (x̃(t), ỹ(t), ψ̃(t))Nk=1, but nonetheless, the unicycle does not collide with an obstacle B_k since the trajectory remains inside the corridor of PI sets σ_gk. 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 (x(t), y(t)) = (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 (x(t), y(t)) = (10.5, 11.9) where the unicycle enters the first of 3 progressively smaller PI sets, each with a lower maximum velocity.

REFERENCES