

Robust Motion-Planning for Uncertain Systems with Disturbances using the Invariant-Set Motion-Planner

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

TR2021-038 May 04, 2021

Abstract

The invariant-set motion-planner uses a collection of safesets to find a collision-free path through an obstacle filled environment [1]–[4]. This paper extends the invariant-set motion-planner to systems with persistently varying disturbances and parametric model uncertainty. This is accomplished by replacing the previously used positive invariant sets with robust positive invariant sets. Since the model uncertainty obfuscates the relationship between the invariant-sets in the state-space, and the references and obstacles in the output-space, we reformulate the dynamics in velocity form so that the system output appears directly in the modified system state. Since the persistently varying disturbances will prevent the closed-loop system from converging to the desired reference, we introduce a new robust connection rule where references are connected when the invariant-set of one reference contains the minimal volume robust invariant-set of another. In addition, we bound the time required to transition between invariant-sets to ensure safety when the obstacles are moving. By parameterizing the invariant-sets using a pre-computed input-to-state Lyapunov function, we reduce the real-time computational complexity of our motion-planner. The robust invariantset motion-planner is demonstrated for an automated highway driving case study.

IEEE Transactions on Automatic Control

Robust Motion-Planning for Uncertain Systems with Disturbances using the Invariant-Set Motion-Planner

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

Abstract—The invariant-set motion-planner uses a collection of safe-sets to find a collision-free path through an obstacle filled environment [1]–[4]. This paper extends the invariant-set motion-planner to systems with persistently varying disturbances and parametric model uncertainty. This is accomplished by replacing the previously used positive invariant sets with robust positive invariant sets. Since the model uncertainty obscures the relationship between the invariant-sets in the state-space, and the references and obstacles in the output-space, we reformulate the dynamics in velocity form so that the system output appears directly in the modified system state. Since the persistently varying disturbances will prevent the closed-loop system from converging to the desired reference, we introduce a new robust connection rule where references are connected when the invariant-set of one reference contains the minimal volume robust invariant-set of another. In addition, we bound the time required to transition between invariant-sets to ensure safety when the obstacles are moving. By parameterizing the invariant-sets using a pre-computed input-to-state Lyapunov function, we reduce the real-time computational complexity of our motion-planner. The robust invariant-set motion-planner is demonstrated for an automated highway driving case study.

I. INTRODUCTION

Motion planning is the problem of generating a dynamically feasible collision-free trajectory between an initial state and a goal state in a generally nonconvex environment. Motion-planning is a fundamental problem in a variety of fields, for instance autonomous driving [5], robotics [6], manufacturing [7], and space exploration [1]. The presence of obstacles in the environment, and hence the nonconvexity, renders this problem computationally difficult [8]. Fortunately, many heuristic motion-planning algorithms have been developed to simplify the problem.

Often motion-planning is split into path-planning and path-tracking. One popular class of path-planners is sampling-based algorithms. Rapidly-exploring random trees (RRT) abstract the path-planning problem as a graph search, where the graph nodes are points sampled from the obstacle-free region and the graph edges indicate collision-free geometric paths connecting these nodes [9]–[11]. Kinodynamic RRT [9], [10] and closed-loop RRT [5] 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 [12]–[14] consider sampling-based motion-planning under uncertainty.

Recently, set-based path-planning algorithms have been growing in popularity. Like sampling-based algorithms, set-based algorithms abstract the path-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]–[4], the nodes of the search graph index sampled 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., a safe-set. 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 [15]–[17] in which an edge of the search graph indicates that the target

node lies in an obstacle-free reachable-set of the current node. LQR-trees [18], [19] are another example of set-based motion-planners. In [19] 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 nodes 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 [20], [21], but has high computational cost due to the need to solve a nonconvex optimization problem.

One of the advantages of set-based motion-planners is their robustness. By sampling sets instead of points, set-based motion-planners have a natural buffer that can absorb tracking errors. Furthermore, by incorporating the controller into the motion-planning process, these algorithms inherit the innate robustness provided by feedback. The objective of this paper is to make this implicit robustness explicit for one particular algorithm: the invariant-set motion-planner. This paper modifies the invariant-set motion-planner for systems with parametric model uncertainty and rate-bounded disturbances. This is accomplished by replacing the positive invariant (PI) sets with *robust* positive invariant (RPI) sets that are explicitly designed to guarantee collision avoidance in the presences of uncertainty and disturbances. Model uncertainty is a reoccurring complication since it makes it impossible to precisely relate the system’s inputs, outputs, and states. For instances, it is impossible to know which state the system will reach for a given sequence of inputs. Furthermore, for a sampled output reference, it is impossible to compute the corresponding equilibrium state where the robust positive invariant (RPI) should be centered. Likewise, the model uncertainty makes it difficult to determine whether an RPI subset of the state-space collides with an obstacle in the output space. All of these issues are addressed by reformulating the dynamics in the velocity form so that the system output appears directly in the velocity state due to the presence of integral-action in the closed-loop system. Thus, the relationship between velocity-states and outputs is unambiguous, despite model uncertainty.

The conditions for connecting safe-sets must also be modified to account for model uncertainty and disturbances. Here, disturbances are the main issue since they prevent the closed-loop system from converging to an equilibrium point. Thus, we cannot connect two safe sets when the equilibrium of the first is included in the second invariant set. Instead, we create connections when the entire minimal volume ellipsoidal RPI set associated with a reference is contained inside the safe-set associated with another reference. In practice, this connection rule only requires tightening the previously used connection radius [1]. We show that the computational complexity of forming edges is quadratic in the dimension of the planning space and independent of the dimension of the state-space. Thus, our algorithm has low computational complexity even for systems with high-dimensional dynamics.

This paper is organized as follows. In Section II we define the motion-planning problem and briefly describe the invariant-set motion-planner which we will robustify to handle uncertainty. In Section III we describe regions of the state-space for which it is

possible to safely track a reference without colliding with an obstacle and we provide a method for safely connecting these safe-sets. In Section IV we demonstrate our robust motion-planning algorithm for highway driving scenario.

Notation and Definitions: A set \mathcal{O} is PI for the autonomous system $\dot{x} = f(x)$ if $x(0) \in \mathcal{O}$ implies that $x(t) \in \mathcal{O}$ for all $t \in \mathbb{R}_+$. If $V(x)$ is a global Lyapunov function for the stable autonomous system $\dot{x} = f(x)$, then any level-set $\mathcal{O} = \{x : V(x) \leq l\}$ is a PI set since $\dot{V} \leq 0$. A set \mathcal{O} is RPI for the disturbed system $\dot{x} = f(x, w)$ if $x(0) \in \mathcal{O}$ implies that $x(t) \in \mathcal{O}$ for all $w(t) \in \mathcal{W}$ and $t \in \mathbb{R}_+$. The notation $x(t) \rightarrow \bar{x}$ is shorthand for $\lim_{t \rightarrow \infty} x(t) = \bar{x}$. For $P \succ 0$ we define the weighted norm $\|x\|_P = \sqrt{x^\top P x}$. The interior of a set \mathcal{Y} is denoted by $\text{int}(\mathcal{Y})$. The image of a set \mathcal{X} through a matrix A is defined as $A\mathcal{X} = \{Ax : x \in \mathcal{X}\}$. The Minkowski sum \oplus of sets \mathcal{X} and \mathcal{Y} is defined as $\mathcal{X} \oplus \mathcal{Y} = \{x + y : x \in \mathcal{X}, y \in \mathcal{Y}\}$. A directed graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ is a set of nodes \mathcal{I} together with a set of ordered pairs $\mathcal{E} \subseteq \mathcal{I} \times \mathcal{I}$ called edges. Nodes $i, j \in \mathcal{I}$ 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 tensor product $\mathcal{G} = \mathcal{G}_1 \times \mathcal{G}_2$ of two graphs $\mathcal{G}_1 = (\mathcal{I}_1, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{I}_2, \mathcal{E}_2)$ is the graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ with nodes $\mathcal{I} = \mathcal{I}_1 \times \mathcal{I}_2$ and edges $((i_1, i_2), (j_1, j_2)) \in \mathcal{E}$ for all $(i_1, j_1) \in \mathcal{E}_1$ and $(i_2, j_2) \in \mathcal{E}_2$.

II. ROBUST MOTION-PLANNING PROBLEM

In this section, we describe the invariant-set motion-planner used to solve motion-planning problems.

A. Motion-Planning Problem

The objective of the motion-planning problem is to plan the motion $y(t) = C(\xi)x(t) \in \mathbb{R}^{n_y}$ of a dynamic system from an initial state $x(0) \in \mathbb{R}^{n_x}$ to a target output $y^\infty \in \mathbb{R}^{n_y}$ through an obstacle filled environment $\bar{\mathcal{Y}} \subseteq \mathbb{R}^{n_y}$ where the physical meaning of $y(t)$ depends on the application e.g. the position of the ego-vehicle in the autonomous driving case-study in Section IV. The obstacle-free region $\mathcal{Y}(t) \subseteq \bar{\mathcal{Y}}$ is modeled by the set-difference of a nominal constraint set $\bar{\mathcal{Y}} \subseteq \mathbb{R}^{n_y}$ and a collection $\{\mathcal{B}_j(t)\}_{j \in \mathcal{J}(t)}$ of obstacle sets $\mathcal{B}_j(t) \subset \mathbb{R}^{n_y}$

$$\mathcal{Y}(t) = \bar{\mathcal{Y}} \setminus \left(\bigcup_{j \in \mathcal{J}(t)} \mathcal{B}_j(t) \right). \quad (1)$$

We assume that the future distribution of obstacles $\{\mathcal{B}_j(t_k)\}_{j \in \mathcal{J}(t_0)}$ is known at regular time intervals $t_k = t_0 + k\Delta\tau$ over a horizon $k = 0, \dots, N$. Furthermore, we assume that the obstacle sets $\mathcal{B}_j(t_k)$ account for the spatial extents of both the obstacle and the system [22] over the entire interval $[t_k, t_{k+1})$, i.e., $\mathcal{B}_j(t) = \mathcal{B}_j(t_k)$ for $t \in [t_k, t_{k+1})$. Note that the number $|\mathcal{J}(t)|$ of obstacles is potentially time-varying.

Without loss of generality, we can assume that the obstacle sets $\mathcal{B}_j(t_k)$ are convex, since any compact set can be covered by a finite union of convex sets [9]. Thus, a nonconvex obstacle can be replaced by a finite collection of convex obstacles. For simplicity, we also assume that the obstacles sets $\mathcal{B}_j(t_k)$ are polyhedral $\mathcal{B}_j(t_k) = \{y : h_{jl}(t_k)y \leq k_{jl}(t_k), l = 1, \dots, \ell\}$ since any convex set can be approximated with arbitrary precision by a polyhedral set [23]. Indeed, in practice the obstacle sets $\mathcal{B}_j(t_k)$ are typically given by a bounding-box or convex-hull of a point-cloud, which are both special-cases of polyhedra.

The motion $y(t)$ is produced by providing a piecewise constant $r(t) = r_k^\infty$ for $t \in [t_k, t_{k+1})$ reference trajectory $r(t)$ a closed-loop

dynamic system, modeled by the following uncertain linear state-space model

$$\dot{x}(t) = A(\xi)x(t) + B(\xi)r(t) + B_w(\xi)w(t) \quad (2a)$$

$$y(t) = C(\xi)x(t) \quad (2b)$$

where $x(t) \in \mathbb{R}^{n_x}$ is the state and $w(t) \in \mathbb{R}^{n_w}$ are disturbances. For simplicity, we update the reference $r(t)$ at the time epochs $t_k = k\Delta\tau$ when the obstacle sets $\mathcal{B}_j(t_k)$ are predicted. The uncertain state-space matrices $(A(\xi), B(\xi), B_w(\xi), C(\xi))$ depend on a constant, but unknown parameter $\xi \in \Xi$ which is contained in the bounded set Ξ . We assume that (2) is a uniform minimal realization of the closed-loop dynamics, i.e., the pair $(A(\xi), B(\xi))$ is controllable and the pair $(A(\xi), C(\xi))$ is observable for all $\xi \in \Xi$.

The dynamics (2) model a physical system in closed-loop with a well-designed controller. Thus, we can assume that the output $y(t)$ of the closed-loop system (2) asymptotically tracks $y(t) \rightarrow r^\infty$ constant references $r(t) = r^\infty$ in the presence of constant disturbances $w(t) = w^\infty$ for all realizations $\xi \in \Xi$ of the model uncertainty. Thus, it is reasonable to assume that the closed-loop system (2) includes integral-action, i.e., the state $x(t)$ of the system (2) includes the integrated tracking error $x_1(t) = \int_0^t y(\tau) - r(\tau) d\tau$. Under these conditions, the closed-loop dynamics (2) can be realized as the following structured state-space model

$$\begin{bmatrix} \dot{x}_p \\ \dot{x}_1 \end{bmatrix} = \underbrace{\begin{bmatrix} A_{11}(\xi) & A_{12}(\xi) \\ C_p(\xi) & 0 \end{bmatrix}}_{A(\xi)} \begin{bmatrix} x_p \\ x_1 \end{bmatrix} + \underbrace{\begin{bmatrix} B_p(\xi) \\ -I \end{bmatrix}}_{B(\xi)} r + \underbrace{\begin{bmatrix} B_{pw}(\xi) \\ 0 \end{bmatrix}}_{B_w(\xi)} w \quad (3)$$

where $x_1(t) \in \mathbb{R}^{n_y}$ is the integrated tracking error and $x_p(t) \in \mathbb{R}^{n_x - n_y}$ is the remainder state, which includes the plant states and potentially other states of a dynamic controller. If the closed-loop system (2) does not include integral-action, then the dynamics can be augmented with integrator dynamics to obtain the structured system (3) with $A_{12}(\xi) = 0$.

The disturbances $w(t)$ are modeled as time-varying, but rate bounded $w(t) \in \mathcal{W} = \{w(t) : \|\dot{w}(t)\|_W^2 \leq 1\}$ where $W \succ 0 \in \mathbb{R}^{n_w \times n_w}$. This model is motivated by the fact that a well designed closed-loop system (3) will include filters that ameliorate noisy measurements. Furthermore, constant disturbances $w(t) = w^\infty$ are a trivial hinderance since they will be rejected by the integral-action in the asymptotically stable closed-loop system (3).

The motion planning problem is summarized below.

Problem 1 (Motion-Planning). *Compute a piecewise constant reference $r(t) = r_k^\infty$ for $t \in [t_k, t_{k+1})$ such that the resulting motion $y(t)$ of the closed-loop system (2) avoids obstacles $y(t) \in \mathcal{Y}(t)$ for all $t \in \mathbb{R}_+$ and reaches the target output $y(t) \rightarrow y^\infty$.*

Problem 1 is challenging since the obstacle-free region (1) is nonconvex and the differential constraints (2) can result in complex motion $y(t)$, i.e., $y(t)$ will not be composed of simple line-segments. More importantly for this paper, we cannot predict the exact trajectory $y(t)$ that the system will follow between waypoints r_k^∞ due to the persistently varying disturbances $w(t) \in \mathcal{W}$ and model uncertainty $\xi \in \Xi$. Thus, it is difficult to guarantee collision-free motion. This is the main issue that we address in this paper.

B. Invariant-Set Motion-Planning Algorithm

One method for constructing a piecewise constant reference $r(t)$ that produces collision-free trajectory $y(t) \in \mathcal{Y}(t)$ is the invariant-set motion-planner described by Algorithm 1 [1]–[4]. Like several other motion-planning algorithms, the invariant-set motion-planner

abstracts motion-planning as a search for a path through a graph. The nodes $i \in \mathcal{I}$ of this graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ index reference outputs $r_i^\infty \in \text{int}(\mathcal{Y})$ sampled from the interior $\text{int}(\mathcal{Y})$ of the obstacle-free region (1) which are tracked by the closed-loop system (2). The references r_i^∞ can be sampled from the obstacle-free region (1) using a variety of methods [22], which are outside the scope of this paper.

The defining feature of the invariant-set motion-planner is that knowledge of the closed-loop system (2) is incorporated into the graph \mathcal{G} using the system's invariant sets $\mathcal{O}_i \subseteq \mathbb{R}^{n_x}$. Associated with each sampled reference output r_i^∞ is an obstacle-free PI set \mathcal{O}_i . Since the set \mathcal{O}_i is PI, we know that the closed-loop dynamics (2) will evolve inside this set, i.e., if $x(0) \in \mathcal{O}_i$ and $r(t) = r_i^\infty$ then $x(t) \in \mathcal{O}_i$ for all $t \in \mathbb{R}_+$. Furthermore, if the invariant set $\mathcal{O}_i \subseteq \mathbb{R}^{n_x}$ is obstacle-free $C(\xi)\mathcal{O}_i \subseteq \mathcal{Y}(t)$ for all $\xi \in \Xi$, then the motion $y(t) = C(\xi)x(t)$ is safe $y(t) \in C(\xi)\mathcal{O}_i \subseteq \mathcal{Y}(t)$ even though the closed-loop system (2) does not perfectly track the reference $y(t) \neq r(t)$. An edge $(i, j) \in \mathcal{E}$ of the search graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ indicates that the system (2) state $x(t)$ will enter the safe-set \mathcal{O}_j of the j -th reference r_j^∞ while tracking the i -th reference r_i^∞ without leaving the current safe-set \mathcal{O}_i . Thus, the invariant-set motion-planner avoids obstacles by moving the system through a series of safe-sets \mathcal{O}_i .

Algorithm 1 Motion-Planning Algorithm

- 1: Find path $\sigma(0), \dots, \sigma(N) \in \mathcal{I}$ through the search graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ where $x(t_0) \in \mathcal{O}_{\sigma(0)}$
 - 2: **for** each time t_k for $k=0, \dots, N$ **do**
 - 3: $r(t) = r_{\sigma(k)}^\infty$
 - 4: **end for**
-

This paper addresses two main issues. First, finding regions $\mathcal{O}_i \subseteq \mathbb{R}^{n_x}$ of the state-space where it is safe $C(\xi)\mathcal{O}_i \subset \mathcal{Y}(t)$ to track the i -th reference $r_i^\infty \in C(\xi)\mathcal{O}_i$ despite model uncertainty $\xi \in \Xi$. Second, connecting $(i, j) \in \mathcal{E}$ safe-sets \mathcal{O}_i and \mathcal{O}_j in the search graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ when the system does not necessarily converge $y(t) \not\rightarrow r_i^\infty$ to the desired reference r_i^∞ due to disturbances $w(t) \in \mathcal{W}$.

III. ROBUST MOTION-PLANNER DESIGN

In this section, we describe a method for finding regions $\mathcal{O}_i \subseteq \mathbb{R}^{n_x}$ where it is safe to track the i -th reference r_i^∞ and provide a method for safely connecting $(i, j) \in \mathcal{E}$ safe-sets \mathcal{O}_i and \mathcal{O}_j .

First, we provide a method for quantifying the robustness of the closed-loop system (2) to model uncertainty $\xi \in \Xi$ and disturbances $w(t) \in \mathcal{W}$, which will guide the construction of the safe sets $\mathcal{O}_i \subseteq \mathbb{R}^{n_x}$ and the search graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$. The robustness of the closed-loop system (2) is quantified by solving

$$\max_{\alpha, X} \log \det X \quad (4a)$$

$$\text{s.t.} \quad \begin{bmatrix} XA(\xi)^\top + A(\xi)X + \alpha X & B_w(\xi) \\ B_w(\xi)^\top & -\alpha W \end{bmatrix} \preceq 0 \quad \forall \xi \in \Xi \quad (4b)$$

where $\alpha > 0 \in \mathbb{R}$ and $X = P^{-1} \succ 0 \in \mathbb{R}^{n_x}$ are the decision variables. Problem (4) is nonconvex, but it can be efficiently solved by alternatively finding $X \succ 0$ and $\alpha \in (0, \bar{\alpha})$, using a semidefinite program and line-search, respectively. Methods for formulating the matrix inequality (4b) for different models of parametric uncertainty $\xi \in \Xi$ can be found in [24, Ch.6]. For example, for systems with polyhedral uncertainty $A(\xi) = \sum_{i=1}^{\ell} \xi_i A_i$ with $\sum_{i=1}^{\ell} \xi_i = 1$ and $\xi_i \geq 0$ the matrix inequality (4b) must be satisfied for each vertex A_i .

Problem (4) produces a quadratic input-to-state Lyapunov function $V(x) = x^\top P x$ that is common to all realizations $\xi \in \Xi$ of the closed-loop system (2) dynamics. The common Lyapunov function allows us to replace the uncertainty of the system state-space matrices (2) with the certainty of the Lyapunov matrix $P \in \mathbb{R}^{n_x \times n_x}$. The asymptotic stability of the closed-loop system (2) does not guarantee that the optimization problem (4) is feasible. However, if the closed-loop system (2) is quadratically stable (i.e. $\exists X \succ 0$ such that $XA(\xi) + A(\xi)^\top X \prec 0 \quad \forall \xi \in \Xi$) then the optimization problem (4) is feasible [25]. The cost (4a) minimizes the volume of the level-sets of the input-to-state Lyapunov function $V(x) = x^\top P x$ [24].

Note that the optimization problem (4) only depends on the state-space parameters $(A(\xi), B_w(\xi))$ of the closed-loop dynamics (2), which are time-invariant, and not on the time-varying environment (1). Thus, the robustness analysis (4) can be conducted offline and will not adversely affect the online complexity of the motion-planner.

In terms of the integral-action state-space realization (3), the Lyapunov matrix $P \succ 0$ obtained from (4) can be partitioned as

$$P = \begin{bmatrix} P_{xx} & P_{xy} \\ P_{yx} & P_{yy} \end{bmatrix} \succ 0 \quad (5)$$

where the partition $P_{yy} \in \mathbb{R}^{n_y \times n_y}$ corresponds to integrated tracking error $x_1(t) \in \mathbb{R}^{n_y}$ and the partition $P_{xx} \in \mathbb{R}^{n_x - n_y \times n_x - n_y}$ corresponds remainder $x_p(t) \in \mathbb{R}^{n_x - n_y}$ of the state $x(t)$. The partitioning (5) of the Lyapunov matrix P is important for determining when a set \mathcal{O}_i is safe and for safely connecting $(i, j) \in \mathcal{E}$ references r_i^∞ and r_j^∞ .

A. Robust Safe Sets

In this section, we provide a method for finding regions $\mathcal{O}_i \subseteq \mathbb{R}^{n_x}$ of the state-space \mathbb{R}^{n_x} in which the closed-loop system (3) can safely track the i -th reference r_i^∞ without colliding with any obstacles $\mathcal{B}_j(t)$ for $j \in \mathcal{J}$. Finding the largest safe-set $\mathcal{O}_i \subseteq \mathbb{R}^{n_x}$ is a nonconvex problem since the obstacle-free region (1) is generally nonconvex. Thus, we focus on finding a computationally tractable, rather than optimal, method for computing the safe-sets \mathcal{O}_i . This includes restricting the safe-sets \mathcal{O}_i to ellipsoidal sets.

Previously [1]–[4], the safe-sets \mathcal{O}_i were obstacle-free ellipsoidal PI sets, centered at an equilibrium state $x_i^\infty \in \mathbb{R}^{n_x}$ corresponding to the reference $r_i^\infty \in \mathbb{R}^{n_y}$ where

$$\begin{bmatrix} A(\xi) \\ C(\xi) \end{bmatrix} x_i^\infty = \begin{bmatrix} -B_w(\xi)w(t) \\ r_i^\infty \end{bmatrix} \quad (6)$$

for a constant disturbance $w(t) \approx \bar{w}$. Unfortunately, we cannot solve (6) for an equilibrium state $x_i^\infty(\xi, w)$ corresponding to the reference r_i^∞ since it depends on the unknown parameter $\xi \in \Xi$ and disturbance $w(t) \in \mathcal{W}$. Furthermore, even if $\xi \in \Xi$ and $w(t) \in \mathcal{W}$ were estimated online, the equilibrium (6) must be recomputed online for each new reference $r_i^\infty \in \text{int}(\mathcal{Y}(t))$ sampled for the time-varying obstacle-free region (1). This requires a matrix inversion or factorization, which could prevent real-time computation. To overcome this issue, we reformulate the closed-loop dynamics (3) in velocity-form [26]

$$\frac{d}{dt} \begin{bmatrix} \dot{x}_p(t) \\ y(t) \end{bmatrix} = \underbrace{\begin{bmatrix} A_{11}(\xi) & A_{12}(\xi) \\ C(\xi) & 0 \end{bmatrix}}_{A(\xi)} \underbrace{\begin{bmatrix} \dot{x}_p(t) \\ y(t) \end{bmatrix}}_{\dot{x}(t)} + \underbrace{\begin{bmatrix} \hat{B}_{pw}(\xi) \\ 0 \end{bmatrix}}_{B_w(\xi)} \dot{w}(t) \quad (7)$$

where $\dot{y}(t) = \dot{c}(t) - \dot{r}(t)$ and $\dot{r}(t) = 0$ for almost all $t \in \mathbb{R}_+$ since the reference $r(t) = r_{\sigma(k)}^\infty$ is piecewise constant on the planning intervals $[t_k, t_{k+1})$. The velocity-dynamics (7) have two main advantages; the equilibrium states $\dot{x}_i^\infty = \begin{bmatrix} 0 \\ r_i^\infty \end{bmatrix}$ corresponding to a reference r_i^∞ do not depend on the model uncertainty $\xi \in \Xi$ or disturbances $w(t) \in \mathcal{W}$

and there is no uncertainty in the output matrix $[0 \ I_{n_y}]$ that relates the system (3) state $\dot{x}(t) = \begin{bmatrix} \dot{x}_p(t) \\ \dot{y}(t) \end{bmatrix}$ to the output $y(t)$.

The following proposition uses the input-to-state Lyapunov matrix (5) to characterize a family of RPI sets for the velocity-dynamics (7).

Proposition 1. *The set $\mathcal{O}_i = \dot{x}_i^\infty + \rho_i \underline{\mathcal{O}}$ is an RPI set of the system (7) for all rate-bounded disturbances $\|\dot{w}\|_W^2 \leq 1$ and model uncertainty $\xi \in \Xi$ if $\rho_i \geq 1$ where*

$$\underline{\mathcal{O}} = \left\{ \begin{bmatrix} \dot{x}_p \\ y \end{bmatrix} : \left\| \begin{bmatrix} \dot{x}_p \\ y \end{bmatrix} \right\|_P^2 \leq 1 \right\} \quad (8)$$

is the minimal volume ellipsoidal RPI set centered at the origin $x^\infty = 0$ and $\dot{x}_i^\infty = \begin{bmatrix} 0 \\ r_i^\infty \end{bmatrix}$ is the equilibrium state-velocity associated with the reference $r_i^\infty \in \mathbb{R}^{n_y}$.

Proof. Pre- and post-multiplying (4) by $\begin{bmatrix} P_{\dot{x}} \\ \dot{w} \end{bmatrix}$ shows that the vector-field $f(\dot{x}, \dot{w}) = A\dot{x} + B_w\dot{w}$ on the boundary $\{\dot{x} : \|\dot{x}\|_P^2 = \rho^2\}$ of the set $\mathcal{O}_i = \rho_i \underline{\mathcal{O}}$ satisfies

$$\begin{aligned} f(\dot{x}, \dot{w})^\top \left(\frac{\partial \|\dot{x}\|_P^2}{\partial \dot{x}} \right) + \left(\frac{\partial \|\dot{x}\|_P^2}{\partial \dot{x}} \right)^\top f(\dot{x}, \dot{w}) &= \\ = (A(\xi)\dot{x} + B_w(\xi)\dot{w})^\top P\dot{x} + \dot{x}^\top P(A(\xi)\dot{x} + B_w(\xi)\dot{w}) & \\ \leq \alpha \dot{w}^\top W\dot{w} - \alpha \dot{x}^\top P\dot{x} \leq -\alpha(\rho^2 - 1) \leq 0 & \end{aligned} \quad (9)$$

for $\alpha > 0$ where $\|\dot{w}\|_W^2 \leq 1$ and $\dot{x}^\top P\dot{x} = \rho^2$. Since $\rho \geq 1$, the vector-field prevents the state \dot{x} from leaving the set \mathcal{O}_i . Thus, if $\rho \geq 1$ then the set (8) is RPI for the system (7) for all $\xi \in \Xi$ and $w(t) \in \mathcal{W}$. \square

According to Proposition 1, the state $\dot{x}(t) = [\dot{x}_p(t)^\top \ y(t)^\top]^\top$ of the dynamics (7) will remain inside the set (8). Thus, the invariant set (8) bounds the motion $y(t)$ of the original closed-loop system (3) since $y(t)$ is a component of the state of the velocity-dynamics (7) due to presence of integral-action. If the Lyapunov matrix (5) was hypothetically block-diagonal, then for states $\dot{x}(t)$ inside the RPI set (8) the resulting motion $y(t)$ of the system (2) remains in a neighborhood $\|y(t) - r_i^\infty\|_{P_{yy}} \leq \rho_i$ of the reference r_i^∞ and the state-velocity $\dot{x}_p(t)$ is low $\|\dot{x}_p(t)\| \leq \rho_i$. The cross-term P_{xy} in the Lyapunov matrix (5) allows the tracking-error $y(t) - r_i^\infty$ to be larger provided that the state-velocity $\dot{x}_p(t)$ is in a direction $\dot{x}_p(t)^\top P_{xy}(y(t) - r_i^\infty) < 0$ that decreases the tracking error $y(t) - r_i^\infty$. The full state $x(t) = \int_0^t \dot{x}(\tau) d\tau$ of the original system (2) is bounded since it is observable. Any feasible solution of the optimization problem (4) provides an RPI set (8), while the optimal solution provides the minimal volume ellipsoidal RPI [24].

Proposition 1 provides a lower-bound $\rho_i \geq 1$ on the radius ρ_i of the neighborhoods \mathcal{O}_i of the equilibrium state $\dot{x}_i^\infty = \begin{bmatrix} 0 \\ r_i^\infty \end{bmatrix}$ in which the system (7) will remain given the uncertainty about the plant dynamics $\xi \in \Xi$ and the presence of disturbances $w(t) \in \mathcal{W}$. Next, we provide a condition on the reference r_i^∞ that ensures that the set \mathcal{O}_i is safe, i.e., $[0 \ I_{n_y}] \mathcal{O}_i \cap \mathcal{B}_j(t) = \emptyset$ during a planning interval $t \in [t_k, t_{k+1})$ where $[0 \ I_{n_y}]$ is the output matrix of the velocity dynamics (7).

Theorem 1. *The set $\mathcal{O}_i = \dot{x}_i^\infty + \rho_i \underline{\mathcal{O}}$ will not intersect $[0 \ I_{n_y}] \mathcal{O}_i \cap \mathcal{B}_j(t) = \emptyset$ the obstacle $\mathcal{B}_j(t)$ at time $t \in \mathbb{R}_+$ if and only if $r_i^\infty \notin \mathcal{B}_j(t) \oplus \rho_i \mathcal{S}$ where $\mathcal{S} = \{y \in \mathbb{R}^{n_y} : y^\top S^{-1} y \leq 1\}$ and*

$$S^{-1} = P_{yy} - P_{yx} P_{xx}^{-1} P_{xy} \quad (10)$$

is the Schur complement of the Lyapunov matrix (5) obtained from robustness analysis (4).

Proof. First, we prove the if portion of Theorem 1. Suppose there exists $x \in \mathbb{R}^{n_x}$ and $y \in \mathbb{R}^{n_y}$ such that $y + r_i^\infty \in \mathcal{B}_j(t)$ and $\begin{bmatrix} x \\ y \end{bmatrix} \in \rho_i \underline{\mathcal{O}}$.

Then, using Schur complements we can show $y \in \rho_i \mathcal{S}$. Likewise, $-y \in \rho_i \mathcal{S}$ since all ellipsoidal sets are centrally symmetric $\mathcal{S} = -\mathcal{S}$. Thus,

$$r_i^\infty \in \mathcal{B}_j(t) \oplus \rho_i \mathcal{S} = \{y_1 + y_2 : y_1 \in \mathcal{B}_j(t), y_2 \in \rho_i \mathcal{S}\}$$

where $y_1 = y + r_i^\infty \in \mathcal{B}_j(t)$ and $y_2 = -y \in \rho_i \mathcal{S}$. Summarizing: if there exists $y' = y + r_i^\infty \in [0 \ I_{n_y}] \mathcal{O}_i \cap \mathcal{B}_j(t)$, then $r_i^\infty \in \mathcal{B}_j(t) \oplus \rho_i \mathcal{S}$. The contrapositive of that statement is: if $r_i^\infty \notin \mathcal{B}_j(t) \oplus \rho_i \mathcal{S}$, then there does not exist $y' \in \mathcal{B}_j(t) \cap [0 \ I_{n_y}] \mathcal{O}_i$, i.e., $\mathcal{B}_j(t) \cap [0 \ I_{n_y}] \mathcal{O}_i = \emptyset$.

Next, we prove the only if portion of Theorem 1. Suppose $r_i^\infty \in \mathcal{B}_j(t) \oplus \rho_i \mathcal{S}$. Then, there exists $y \in \mathcal{B}_j(t)$ such that $y - r_i^\infty \in \rho_i \mathcal{S}$ by definition of the Minkowski sum $\mathcal{B}_j(t) \oplus \rho_i \mathcal{S}$. Define $x = -P_{xx}^{-1} P_{xy}(y - r_i^\infty)$. Then,

$$\begin{aligned} \begin{bmatrix} x \\ y - r_i^\infty \end{bmatrix}^\top \begin{bmatrix} P_{xx} & P_{xy} \\ P_{yx} & P_{yy} \end{bmatrix} \begin{bmatrix} x \\ y - r_i^\infty \end{bmatrix} &= \\ = (y - r_i^\infty)^\top \underbrace{(P_{yy} - P_{yx} P_{xx}^{-1} P_{xy})}_{S^{-1}} (y - r_i^\infty) &\leq \rho_i^2 \end{aligned}$$

where $(y - r_i^\infty)^\top S^{-1} (y - r_i^\infty) \leq \rho_i^2$ since $y - r_i^\infty \in \rho_i \mathcal{S}$. Thus, $y \in [0 \ I_{n_y}] \mathcal{O}_i$ since $\begin{bmatrix} x \\ y - r_i^\infty \end{bmatrix} \in \rho_i \underline{\mathcal{O}}$. Therefore, $y \in \mathcal{B}_j(t) \cap [0 \ I_{n_y}] \mathcal{O}_i \neq \emptyset$. \square

Theorem 1 uses the robustness (4) of the closed-loop system (3) to determine whether an invariant-set \mathcal{O}_i is obstacle-free $[0, I] \mathcal{O}_i \cap \mathcal{B}_j(t) = \emptyset$. Since the obstacle-sets are piecewise constant $\mathcal{B}_j(t) = \mathcal{B}_j(t_k)$ for $t \in [t_k, t_{k+1})$, Theorem 1 means that the invariant-set \mathcal{O}_i is obstacle-free and therefore safe over the time interval $[t_k, t_{k+1})$. Theorem 1 uses the Schur complement (10) rather than the output component P_{yy} of the Lyapunov matrix (5). This more restrictive condition $S^{-1} \succeq P_{yy}^{-1}$ ensures, not only that the system is initially outside the obstacle $y(t_k) \notin \mathcal{B}_j(t_k)$, but also that the state-velocity $\dot{x}_p(t)$ is small enough so that the dynamics (7) will prevent the output $y(t)$ from colliding with an obstacle $y(t) \notin \mathcal{B}_j(t) = \mathcal{B}_j(t_k)$ in the future $t \in [t_k, t_{k+1})$.

Theorem 1 is applicable to any class of obstacle-sets $\mathcal{B}_j(t)$. For the polyhedral obstacle-sets $\mathcal{B}_j(t)$ considered in this paper, the Minkowski sum $\mathcal{B}_j(t) \oplus \rho_i \mathcal{S}$ in Theorem 1 is troublesome for three reasons. First, computing the Minkowski sum of polyhedra described by their half-spaces is known to be computationally expensive, in general. Second, the Minkowski sum of ellipsoidal sets is not necessarily ellipsoidal. And third, the Minkowski sum of an ellipsoidal set and a polyhedron is neither an ellipsoid nor a polyhedron. The following corollary provides a polyhedral outer-approximation of the Minkowski sum $\mathcal{B}_j(t) \oplus \rho_i \mathcal{S}$ that is both tight and inexpensive to compute.

Corollary 1. *Let $\mathcal{B}_j(t) = \{y : h_{jl}(t)^\top y \leq k_{jl}, l = 1, \dots, \ell\}$. Then $\mathcal{B}_j(t) \oplus \rho_i \mathcal{S} \subseteq \bar{\mathcal{B}}_j(t)$ where*

$$\bar{\mathcal{B}}_j(t) = \{y : h_{jl}(t)^\top y \leq k_{jl} + \rho_i \|h_{jl}\|_S, l = 1, \dots, \ell\} \quad (11)$$

and $S > 0$ is the inverse of the Schur complement (10) of (5) obtained by solving (4).

Proof. For notational simplicity, we will drop the indices i, j and time t from ρ_i and $\mathcal{B}_j(t)$.

The set $\mathcal{B} = \bigcap_{l=1}^\ell \mathcal{H}_l$ is the finite $\ell < \infty$ intersection of half-spaces $\mathcal{H}_l = \{y : h_l^\top y \leq k_l\}$. Thus,

$$\begin{aligned} \mathcal{B} \oplus \rho \mathcal{S} &= \bigcup_{y \in \rho \mathcal{S}} \bigcap_{l=1}^\ell \mathcal{H}_l + y \\ &\subseteq \bigcap_{l=1}^\ell \bigcup_{y \in \rho \mathcal{S}} \mathcal{H}_l + y = \bigcap_{l=1}^\ell \mathcal{H}_l \oplus \rho \mathcal{S} \end{aligned}$$

by the properties of unions and intersections. Therefore, we can outer-approximate the Minkowski sum $\mathcal{B} \oplus \rho\mathcal{S}$ by the finite intersection of Minkowski sums $\mathcal{H}_l \oplus \rho\mathcal{S}$ of a half-space \mathcal{H}_l and an ellipsoid $\rho\mathcal{S}$. Note that

$$\begin{aligned}\mathcal{H}_l \oplus \rho\mathcal{S} &= S^{\frac{1}{2}} S^{-\frac{1}{2}} (\mathcal{H}_l \oplus \rho\mathcal{S}) \\ &= S^{\frac{1}{2}} (\hat{\mathcal{H}}_l \oplus \rho\mathbb{S})\end{aligned}\quad (12)$$

where $S^{-\frac{1}{2}}$ is the matrix-square-root of the Schur complement (10), $\hat{\mathcal{H}}_l = S^{-\frac{1}{2}} \mathcal{H}_l = \{y : \hat{h}^\top y \leq k\}$ with $\hat{h}_l = (S^{\frac{1}{2}})^\top h_l$, and $\mathbb{S} = \{y : \|y\|_2 \leq 1\} \subset \mathbb{R}^{n_y}$ is the unit-sphere in \mathbb{R}^{n_y} . The Minkowski sum $\hat{\mathcal{H}}_l \oplus \rho\mathbb{S}$ has a closed-form solution

$$\hat{\mathcal{H}}_l \oplus \rho\mathbb{S} = \hat{\mathcal{H}}_l + \rho \frac{\hat{h}_l}{\|\hat{h}_l\|}. \quad (13)$$

For verification, note that if $y \in \hat{\mathcal{H}}_l \oplus \rho\mathbb{S}$ then $y = y_1 + y_2$ where $h_l^\top y_1 \leq k$ and $y_2^\top y_2 \leq \rho^2$. Thus, $\hat{h}_l^\top y = \hat{h}_l^\top y_1 + \hat{h}_l^\top y_2 \leq k + \rho \|\hat{h}_l\|$, i.e., $y \in \hat{\mathcal{H}}_l + \rho \frac{\hat{h}_l}{\|\hat{h}_l\|}$. Therefore, $\hat{\mathcal{H}}_l \oplus \rho\mathbb{S} \subseteq \hat{\mathcal{H}}_l + \rho \frac{\hat{h}_l}{\|\hat{h}_l\|}$.

Conversely, if $y \in \hat{\mathcal{H}}_l + \rho \frac{\hat{h}_l}{\|\hat{h}_l\|}$ then define $y_2 = \rho \frac{\hat{h}_l}{\|\hat{h}_l\|} \in \rho\mathbb{S}$ so that $y_1 = y - y_2 \in \hat{\mathcal{H}}_l$ since $\hat{h}_l^\top y_1 = \hat{h}_l^\top y - \hat{h}_l^\top y_2 \leq k + \rho \|\hat{h}_l\| - \rho \|\hat{h}_l\| = k$. Thus, $\hat{\mathcal{H}}_l \oplus \rho\mathbb{S} \supseteq \hat{\mathcal{H}}_l + \rho \frac{\hat{h}_l}{\|\hat{h}_l\|}$. Therefore, we can conclude (13) holds.

Substituting (13) into (12) yields

$$\begin{aligned}\mathcal{H}_l \oplus \rho\mathcal{S} &= S^{\frac{1}{2}} \left(\hat{\mathcal{H}}_l + \rho \frac{S^{\frac{1}{2}} h_l}{\|h_l\|_S} \right) = \hat{\mathcal{H}}_l + \rho \frac{S h_l}{\|h_l\|_S} \\ &= \{y : h_l^\top y \leq k_l + \rho \|h_l\|_S\}\end{aligned}$$

where $S^{\frac{1}{2}} = (S^{\frac{1}{2}})^\top$. Taking the intersection over $l = 1, \dots, \ell$ yields the enlarged set (11). \square

In most motion-planning algorithms, collision checking is the most computationally expensive operation. According to Corollary 1, the collision check can be performed by checking whether a reference r_i^∞ is contained in the expanded polyhedron (11), which has the same number of constraints ℓ as the original obstacle set $\mathcal{B}_j(t)$. Thus, the only additional computation is the evaluation of a series of weighted 2-norm $\|h_{jl}(t)\|_S$ for $j \in \mathcal{J}(t)$ to expand the set (11) where the Schur complement (10) of the Lyapunov function (5) can be computed offline since it only depend on the time-invariant system dynamics (7), rather than the time-varying environment (1). Thus, the total computational cost $O(|\mathcal{I}||\mathcal{J}(t)|n_y^2)$ of the collision check $[0 \ I_{n_y}] \mathcal{O}_i \cap \mathcal{B}_j(t) = \emptyset$ is linear in both the number of references $|\mathcal{I}|$ and number of obstacles $|\mathcal{J}(t)|$, and quadratic in the output dimension, which is typically small $n_y \leq n_x \ll |\mathcal{I}|, |\mathcal{J}(t)|$. Therefore, the collision check $r_i^\infty \notin \tilde{\mathcal{B}}_j(t_k)$ can be computed in real-time for a large number of references $|\mathcal{I}| \gg 1$ and obstacles $|\mathcal{J}(t)| \gg 1$, even for high-dimensional system $n_x \gg 1$ provided the output dimension is low $n_y \ll n_x$.

B. Robust Connection Rules

In this section, we describe how to connect $(i, j) \in \mathcal{E}$ safe-sets \mathcal{O}_i and \mathcal{O}_j in the search graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ used by Algorithm 1.

The edges $(i, j) \in \mathcal{E}$ of the search graph are meant to indicate that it is possible to safely transition from tracking the i -th reference r_i^∞ to tracking the j -th reference r_j^∞ . Previously [1]–[4], an edge $(i, j) \in \mathcal{E}$ was included in the search graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ if an equilibrium state x_i^∞ corresponding to the i -th reference r_i^∞ was contained $x_i^\infty \in \mathcal{O}_j$ in the j -th safe-set \mathcal{O}_j . But again, the equilibrium states $x_i^\infty(\xi, w)$ of the uncertain system (3) depend on the unknown disturbance $w(t) \in \mathcal{W}$ and model uncertainty $\xi \in \Xi$. As before, this issue can be resolved by using the velocity-dynamics (7). However, the persistently varying disturbance $\dot{w}(t)$ presents a new issue since it prevents the system

from converging $x(t) \rightarrow x_i^\infty$ to an equilibrium x_i^∞ . Thus, we need a new rule for connecting $(i, j) \in \mathcal{E}$ the graph nodes $i, j \in \mathcal{I}$ that is robust to persistently varying disturbances $\dot{w}(t) \neq 0$.

First, we consider the simpler case in which the obstacles are static $\mathcal{B}_j(t) = \mathcal{B}_j$ for all $t \in \mathbb{R}_+$. For static obstacles, we only need to show that the system is guaranteed to enter the safe set \mathcal{O}_j of the reference r_j^∞ while tracking the i -th reference r_i^∞ . This is shown in the following proposition.

Proposition 2. *If $\dot{x}_i^\infty + \mathcal{Q} \subseteq \text{int}(\mathcal{O}_j)$ then the state $\dot{x}(t)$ of the velocity-dynamics (7) will enter and remain inside the j -th safe-set \mathcal{O}_j while tracking the i -th reference r_i^∞ for all $w(t) \in \mathcal{W}$ and $\xi \in \Xi$.*

Proof. Consider the Lyapunov-like function

$$V_i(\dot{x}) = \begin{cases} (\dot{x} - \dot{x}_i^\infty)^\top P (\dot{x} - \dot{x}_i^\infty) - 1 & \text{if } \dot{x} \notin \dot{x}_i^\infty + \mathcal{Q} \\ 0 & \text{if } \dot{x} \in \dot{x}_i^\infty + \mathcal{Q} \end{cases} \quad (14)$$

which is zero inside of the safe-set $\dot{x}_i^\infty + \mathcal{Q}$ and positive elsewhere since $\|\dot{x} - \dot{x}_i^\infty\|_P^2 = 1$ on the boundary of \mathcal{Q} by definition (8). Thus, we can show $\dot{x}(t) \rightarrow \dot{x}_i^\infty + \mathcal{Q}$ if $V(\dot{x}(t)) \rightarrow 0$. From (9) we have

$$\begin{aligned}\frac{d}{dt} V_i(\dot{x}) &\leq -\alpha (\dot{x} - \dot{x}_i^\infty)^\top P (\dot{x} - \dot{x}_i^\infty) + \alpha \dot{w}^\top W \dot{w} \\ &\leq -\alpha (\dot{x} - \dot{x}_i^\infty)^\top P (\dot{x} - \dot{x}_i^\infty) + \alpha = -\alpha V_i(\dot{x})\end{aligned}$$

where $\|\dot{w}(t)\|_W^2 \leq 1$ and $\alpha > 0$. Thus, $V_i(\dot{x}) \rightarrow 0$ exponentially converges to zero which implies the state of the velocity-dynamics (7) converges $\dot{x}(t) \rightarrow \dot{x}_i^\infty + \mathcal{Q} \subseteq \mathcal{O}_j$ to the j -th safe-set \mathcal{O}_j while tracking the i -th reference r_i^∞ . \square

Proposition 2 says that the references r_i^∞ and r_j^∞ can be connected if the smallest neighborhood $\dot{x}_i^\infty + \mathcal{Q}$ to which the state $\dot{x}(t)$ of the closed-loop dynamics (7) is guaranteed to converge is contained in the next safe-set \mathcal{O}_j . The following corollary provides a computationally tractable method for testing the set-inclusion $\dot{x}_i^\infty + \mathcal{Q} \subseteq \text{int}(\mathcal{O}_j)$.

Corollary 2. *The reference outputs r_i^∞ and r_j^∞ can be connected by a directed edge $(i, j) \in \mathcal{E}$ if and only if*

$$\|r_i^\infty - r_j^\infty\|_{P_{yy}} < \rho_j - 1 \quad (15)$$

where P_{yy} tracking error partition of the Lyapunov matrix (5).

Proof. First, we show that if (15) holds then $\dot{x}_i^\infty + \mathcal{Q} \subseteq \text{int}(\mathcal{O}_j)$. The elements of the ellipsoidal set $\dot{x}_i^\infty + \mathcal{Q}$ can be parameterized as $\dot{x} = P^{-\frac{1}{2}} z + \dot{x}_i^\infty$ where $\|z\|_2^2 \leq 1$, $P^{-\frac{1}{2}}$ is the matrix square-root of $P^{-1} \succ 0$, and $\dot{x}_i^\infty = [r_i^\infty]$ is the equilibrium state-velocity. Thus, the set-inclusion $\dot{x}_i^\infty + \mathcal{Q} \subseteq \text{int}(\mathcal{O}_j)$ holds if the following inequality is satisfied

$$\left(\dot{x}_i^\infty - \dot{x}_j^\infty + P^{-\frac{1}{2}} z \right)^\top P \left(\dot{x}_i^\infty - \dot{x}_j^\infty + P^{-\frac{1}{2}} z \right) < \rho_j^2$$

for all $\|z\|_2^2 \leq 1$. The worst-case z can be found by solving the following optimization problem

$$\max_z \quad \|\dot{x}_i^\infty - \dot{x}_j^\infty\|_P^2 + 2(\dot{x}_i^\infty - \dot{x}_j^\infty)^\top P^{\frac{1}{2}} z + \|z\|_2^2 \quad (16a)$$

$$\text{s.t.} \quad \|z\|_2^2 \leq 1. \quad (16b)$$

This optimization problem has a closed-form solution, namely $z^* = \frac{P^{\frac{1}{2}} (\dot{x}_i^\infty - \dot{x}_j^\infty)}{\|\dot{x}_i^\infty - \dot{x}_j^\infty\|_P}$ which yields the following quadratic equation

$$\|\dot{x}_i^\infty - \dot{x}_j^\infty\|_P^2 + 2\|\dot{x}_i^\infty - \dot{x}_j^\infty\|_P + 1 - \rho_j^2 < 0.$$

Since this quadratic equation is strictly convex, it is negative between its roots, which are $-1 \pm \rho_j$. Since the norm $\|\dot{x}_i^\infty - \dot{x}_j^\infty\|_P \geq 0$ is already positive and $-\rho_j - 1 < 0$ is negative, the inequality $\|\dot{x}_i^\infty -$

$\dot{x}_j^\infty \|_P \geq -\rho_j - 1$ is redundant. This leaves the inequality $\|\dot{x}_i^\infty - \dot{x}_j^\infty\|_P < \rho_j - 1$. Substituting $\dot{x}_i^\infty = \begin{bmatrix} 0 \\ r_i^\infty \end{bmatrix}$ we obtain (15).

Next, we show that if (15) does not hold then $\dot{x}_i^\infty + \underline{\mathcal{Q}} \not\subseteq \mathcal{O}_j$. Again, consider the worst-case element $\dot{x} = \dot{x}_i^\infty + \frac{(\dot{x}_i^\infty - \dot{x}_j^\infty)}{\|\dot{x}_i^\infty - \dot{x}_j^\infty\|_P} \in \dot{x}_i^\infty + \underline{\mathcal{Q}}$ of the minimal RPI set $\dot{x}_i^\infty + \underline{\mathcal{Q}}$. This state \dot{x} satisfies

$$\begin{aligned} (\dot{x} - \dot{x}_j^\infty)^\top P (\dot{x} - \dot{x}_j^\infty) &= \|r_i^\infty - r_j^\infty\|_{P_{yy}}^2 + 2\|r_i^\infty - r_j^\infty\|_{P_{yy}} + 1 \\ &\geq (\rho_j - 1)^2 + 2(\rho_j - 1) + 1 = \rho_j^2 \end{aligned}$$

where $\|r_i^\infty - r_j^\infty\|_{P_{yy}} \geq \rho_j - 1$ since (15) does not hold. Thus, $\dot{x} \in \dot{x}_i^\infty + \underline{\mathcal{Q}}$ but $\dot{x} \notin \text{int}(\mathcal{O}_j)$. \square

The connection rule (15) is necessarily more restrictive $\rho_j - 1 < \rho_j$ than the previously [1]–[4] used connection rule $\|r_i^\infty - r_j^\infty\| < \rho_j$ due to the model uncertainty $\xi \in \Xi$ and persistently varying disturbances $\|\dot{w}(t)\|_V \leq 1$. For static obstacles $\mathcal{B}_j(t) = \mathcal{B}_j$, we add an edge $(i, j) \in \mathcal{E}$ to the search graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ for every pair of references r_i^∞, r_j^∞ for $i, j \in \mathcal{I}$ that satisfy (15).

We now return to the case of moving obstacles $\mathcal{B}_j(t)$. For time-varying obstacles, it is important that we complete the transition from safe-set \mathcal{O}_i to the next safe-set \mathcal{O}_j within a single planning interval $[t_k, t_{k+1})$ since the current safe-set \mathcal{O}_i may become unsafe in the future $t > t_{k+1}$. The transition time is uncertain due to the model uncertainty and presence of disturbances. Nonetheless, the following corollary bounds the worst-case transition time.

Corollary 3. *Suppose the closed-loop system (7) is tracking $y(t) \rightarrow r(t) = r_i^\infty$ the i -th reference r_i^∞ . Then the state $\dot{x}(t) \in \mathcal{O}_j$ is contained in the j -th safe-set \mathcal{O}_j for all times $t \geq \tau_{ij} \in \mathbb{R}_+$ after*

$$\tau_{ij} = \frac{1}{\alpha} \log \left[\frac{\rho_i^2 - 1}{(\rho_j - \|r_i^\infty - r_j^\infty\|_{P_{yy}})^2 - 1} \right] \quad (17)$$

where α was obtained from (4).

Proof. Since $\dot{x}_i^\infty + \underline{\mathcal{Q}} \subseteq \text{int}(\mathcal{O}_j)$, we can enlarge $\rho \dot{x}_i^\infty + \underline{\mathcal{Q}}$ the set $\dot{x}_i^\infty + \underline{\mathcal{Q}}$ such that $\rho \dot{x}_i^\infty + \underline{\mathcal{Q}} \subseteq \mathcal{O}_i$. The maximum radius ρ can be found by solving an optimization problem similar to (16). Again, this optimization problem has a closed-form solution which yields the following quadratic equation

$$\|r_i^\infty - r_j^\infty\|_{P_{yy}}^2 + 2\rho\|r_i^\infty - r_j^\infty\|_{P_{yy}} + \rho^2 - \rho_j^2 \leq 0.$$

This inequality holds between the roots $\|r_i^\infty - r_j^\infty\|_{P_{yy}} = -\rho \pm \rho_j$. Since the distance $\|r_i^\infty - r_j^\infty\|_{P_{yy}} \geq 0$ is positive, this produces the maximum radius $\rho = \rho_j - \|r_i^\infty - r_j^\infty\|_{P_{yy}} > 1$ such that $\rho \dot{x}_i^\infty + \underline{\mathcal{Q}} \subseteq \mathcal{O}_i$.

The maximum amount of time τ_{ij} spent to reach the safe-set \mathcal{O}_j is given by the amount of time required for the Lyapunov-like function (14) to decay from the initial value $\rho_i^2 - 1$ to final value $\rho^2 - 1 = (\rho_j - \|r_i^\infty - r_j^\infty\|_{P_{yy}})^2 - 1$. Since the Lyapunov-like function (14) is exponentially decaying, we have

$$(\rho_i^2 - 1)e^{-\alpha t} \leq (\rho_j - \|r_i^\infty - r_j^\infty\|_{P_{yy}})^2 - 1.$$

The upper-bound (17) is obtained by solving for $t = \tau_{ij}$. \square

The transition time (17) from each safe-set \mathcal{O}_i to itself is zero $\tau_{ii} = 0$ since the safe-sets are PI. Also, for all pairs of references r_i^∞ and r_j^∞ that satisfy the connection rule (15) the transition time $\tau_{ij} < \infty$ is finite. However, if the strict inequality in the connection rule (15) is replaced with a non-strict inequality, then the transition time τ_{ij} will be infinite $\tau_{ij} = \infty$ when the equality holds $\|r_i^\infty - r_j^\infty\| = \rho_j - 1$.

Next, we describe the construction of the search graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ for moving obstacles $\mathcal{B}_j(t)$. We adopt the approach from [1], [4],

which is split into offline and online phases. In the offline phase, we construct a graph \mathcal{R}_N that describes the N -step reachability properties of the system (7). The nodes of this graph \mathcal{R}_N are pairs $(i, k) \in \mathcal{I} \times \mathbb{N}$ of reference indices $i \in \mathcal{I}$ and planning instances $k = 0, \dots, N$. The edges of the graph \mathcal{R}_N are pairs $((i, k), (j, k+1))$ where $\tau_{ij} < \Delta\tau$, indicating that any state $\dot{x}(t) \in \mathcal{O}_i$ inside the invariant-set \mathcal{O}_i can move into the invariant-set \mathcal{O}_j during a single planning interval $[t_k, t_{k+1})$ where $t_k = k\Delta\tau$. Note that $((i, k), (i, k+1)) \in \mathcal{E}$ is always an edge in the reachability graph \mathcal{R}_N since $\tau_{ii} = 0 < \Delta\tau$ according to Corollary 3. The N -step reachability graph \mathcal{R}_N is a causal $(N+1)$ -partite graph. This property can be exploited to accelerate the graph search in Algorithm 1 [27].

The computational complexity $O(|\mathcal{I}|^2 n_y^2)$ of constructing the edge list \mathcal{E} of the reachability graph $\mathcal{R}_N = (\mathcal{I}, \mathcal{E})$ is independent of the state-dimension and therefore can be efficiently computed for systems with high-dimensional dynamics $n_x \gg n_y$. However, the computational complexity $O(|\mathcal{I}|^2 n_y^2)$ is also quadratic in the number $|\mathcal{I}|$ of reference samples r_i^∞ which can be extremely large $|\mathcal{I}| \gg 1$. Therefore, constructing the reachability graph $\mathcal{R}_N = (\mathcal{I}, \mathcal{E})$ can be onerous. However, the reachability graph \mathcal{R}_N can be constructed offline since it describes the reachability properties of the closed-loop dynamics (7), which are independent of the time-varying environment (1).

Algorithm 2 Search Graph Construction: Moving Obstacles

- 1: initialize graph $\mathcal{G} = \mathcal{R}_N$
 - 2: **for** each $i \in \mathcal{I}$ and $k = 1, \dots, N$ **do**
 - 3: **if** $r_i^\infty \in \bar{\mathcal{B}}_j(t_k)$ **then**
 - 4: remove nodes (i, k) and $(i, k-1)$ from \mathcal{G}
 - 5: **end if**
 - 6: **end for**
-

During the online phase, we use Algorithm 2 to remove nodes (i, k) from the reachability graph \mathcal{R}_N that collide with the moving obstacles. Algorithm 2 is initialized $\mathcal{G} = \mathcal{R}_N$ with the N -step reachability graph \mathcal{R}_N . For each reference index $i \in \mathcal{I}$ and planning instance $k = 1, \dots, N$, Algorithm 2 performs a collision check $[0, I_{n_y}] \mathcal{O}_i \cap \mathcal{B}_j(t_k) \neq \emptyset$ for each obstacle $\{\mathcal{B}_j(t_k)\}_{j \in \mathcal{J}}$ at each planning instance $k = 0, \dots, N$. Collision checks are efficiently performed by testing the inclusion $r_i^\infty \in \bar{\mathcal{B}}_j(t_k)$ where (11) bounds the Minkowski sum $\mathcal{B}_j(t_k) \oplus \rho_i \mathcal{S}$ according to Corollary 1. If the inclusion $r_i^\infty \in \bar{\mathcal{B}}_j(t_k)$ holds then the invariant set \mathcal{O}_i is unsafe during the time interval $[t_k, t_{k+1})$ and therefore the node (i, k) is removed from the search graph \mathcal{G} . In addition, the node $(i, k-1)$ is also removed since the state $\dot{x}(t)$ could prematurely transition $j \rightarrow i$ into the set \mathcal{O}_i during the time interval $[t_{k-1}, t_k]$. The resulting search graph \mathcal{G} is then used by the motion-planner (Algorithm 1) to find a sequence of references $r(t) = r_{\sigma(t)}^\infty$ that moves the output $y(t)$ of the system (2) to the target output y_T while avoiding moving obstacles $y(t) \in \mathcal{Y}(t)$. Note that if collision avoidance is not possible then the graph \mathcal{G} will not be connected.

IV. CASE STUDY: HIGHWAY DRIVING

In this section, we demonstrate the robust invariant-set motion-planner in a simplified automated highway driving case study.

The closed-loop motions of the vehicle in the longitudinal and lateral directions are modeled by linear systems (2). Although the longitudinal and linearized lateral dynamics are independent, they are combined into a single linear system since avoiding obstacles will require coordinated movement in both directions. The state $x_1(t) \in \mathbb{R}^3$ of the closed-loop longitudinal dynamics includes the

longitudinal position $y_1(t)$, longitudinal velocity $\dot{y}_1(t)$, and the integrated longitudinal tracking error $\int_0^t y_1(\tau) - r_1(\tau) d\tau$. The input and output are the throttle $u_1(t)$ and longitudinal position $y_1(t)$, respectively. The state $x_2(t) \in \mathbb{R}^5$ of the closed-loop lateral dynamics includes the lateral position $y_2(t)$ and velocity $\dot{y}_2(t)$, the yaw $\psi(t)$ and yaw-rate $\dot{\psi}(t)$, and the integrated lateral tracking error $\int_0^t y_2(\tau) - r_2(\tau) d\tau$. The input and output are the steering angle $u_2(t)$ and lateral position $y_2(t)$ respectively. The dependence of the lateral dynamics on the longitudinal velocity is modeled as an additive disturbance $w_2(t)$ [15]. See [28, eq. (4.1) and (2.45)] for details on the longitudinal and lateral dynamics.

Since the longitudinal and lateral dynamics are independent, both the robustness analysis (4) and RPI set design (8) can be performed independently. The RPI sets for the combined dynamics are cartesian-products $\mathcal{O}_i = \mathcal{O}_{i,1} \times \mathcal{O}_{i,2}$ of the longitudinal $\mathcal{O}_{i,1}$ and lateral $\mathcal{O}_{i,2}$ RPI sets. A common radius $\rho_1 > 1$ was chosen for all the longitudinal RPI sets $x_{i,1}^\infty + \rho_1 \mathcal{O}_{i,1} \subset \mathbb{R}^3$ such that the projection $r_{i,1}^\infty + \rho_1 \mathcal{S}_1 \in \mathbb{R}^1$ has a radius of 2.5 m. Similarly, a common radius $\rho_2 > 1$ was chosen for the lateral RPI sets $x_{i,2}^\infty + \rho_2 \mathcal{O}_{i,2}$ such that the projection $r_{i,2}^\infty + \rho_2 \mathcal{S}_2 \in \mathbb{R}^1$ has a radius of 0.5 m. According to Theorem 1, this means we need a 2.5×0.5 m safety envelop around the ego-vehicle to account for imperfect reference tracking $y(t) \neq r(t)$.

Reachability graphs $\mathcal{R}_{N,1}$ and $\mathcal{R}_{N,2}$ for the longitudinal and lateral dynamics were independently constructed, with a common planning rate of $\Delta\tau = 1$ s and planning horizon of $N\Delta\tau = 20$ s. The nodes \mathcal{I}_1 of the longitudinal reachability graph $\mathcal{R}_{N,1} = (\mathcal{I}_1, \mathcal{E}_1)$ were sampled by gridding the longitudinal position with a resolution of 1 m for ± 20 m around the nominal trajectory $y_1(t) = v_{ego}t$ where $v_{ego} = 24$ m/s is the nominal velocity of the ego-vehicle. Nodes $i, j \in \mathcal{I}_1$ are connected by an edge $(i, j) \in \mathcal{E}_1$ if it is possible to reach safe-set $\mathcal{O}_{j,1}$ from $\mathcal{O}_{i,1}$ within one planning interval, i.e., $\tau_{ij} \leq \Delta\tau$ where the planning rate is $\Delta\tau = 1$ second. Due to the meek closed-loop dynamics, only adjacent nodes were connected, i.e., $(i, j) \in \mathcal{E}_1$ if and only if $r_i^\infty = r_j^\infty \pm 1$. Likewise, the nodes \mathcal{I}_2 of the lateral reachability graph $\mathcal{R}_{N,2} = (\mathcal{I}_2, \mathcal{E}_2)$ were sampled by gridding the lateral position with a resolution of 0.25 m for ± 2 m. Nodes $i, j \in \mathcal{I}_2$ were connected by an edge $(i, j) \in \mathcal{E}_2$ if it is possible to reach safe-set $\mathcal{O}_{j,2}$ from $\mathcal{O}_{i,2}$ within one planning interval of $\Delta\tau = 1$ second. Again, due to the meek closed-loop dynamics, only adjacent nodes were connected, i.e., $(i, j) \in \mathcal{E}_1$ if and only if $r_i^\infty = r_j^\infty \pm 0.25$. The reachability graph $\mathcal{R}_N = (\mathcal{I}, \mathcal{E})$ for the combined longitudinal and lateral dynamics was obtained by computing the tensor product $\mathcal{R} = \mathcal{R}_{N,1} \times \mathcal{R}_{N,2}$ of the longitudinal $\mathcal{R}_{N,1}$ and lateral $\mathcal{R}_{N,2}$ reachability graphs, producing a graph with $|\mathcal{I}| = 28,414$ nodes and $|\mathcal{E}| \approx 30 \times 10^6$ edges. Recall that the reachability graph \mathcal{R} is constructed offline since it describes reachability properties of the system dynamics (3) which are independent of the time-varying environment (1).

In this case study, the ego-vehicle shares the road with $|\mathcal{J}| = 2$ obstacle-vehicles $\{\mathcal{B}_j(t)\}_{j=1,2}$, as shown in Figure 2. The first obstacle $\mathcal{B}_1(t)$ is initially 27 m ahead of the ego-vehicle, in the same lane, traveling 2 m/s slower than the ego-vehicle. This means that the ego-vehicle must either slow down or change lanes to avoid this obstacle-vehicle $\mathcal{B}_1(t)$. The second obstacle-vehicle $\mathcal{B}_2(t)$ is in the opposite lane, initially 10 m behind the ego-vehicle and traveling 1 m/s faster than the ego-vehicle. This means that if the ego-vehicle decides to change lanes, then it will either need to speed-up and get ahead of the obstacle-vehicle $\mathcal{B}_2(t)$ or slow-down and get behind it. This case study illustrates that the invariant-set motion-planner uses knowledge of the closed-loop vehicle dynamics to both plan and execute driving maneuvers.

Algorithm 2 was used to remove unsafe nodes from the reachability

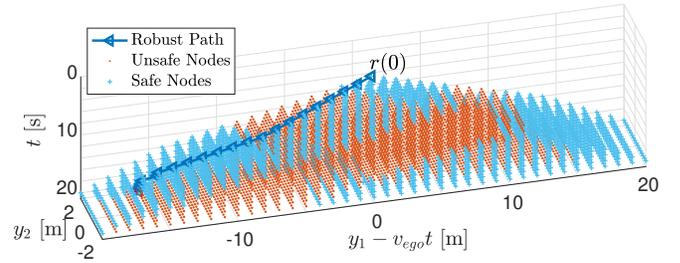


Fig. 1. Nodes \mathcal{I} of reachability graph $\mathcal{R}_N = (\mathcal{I}, \mathcal{E})$. Only nodes $i \in \mathcal{I}$ reachable from initial condition $(y_1, y_2) = (0, -1.5)$ and inside road boundaries are shown.

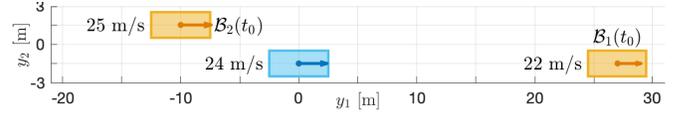


Fig. 2. Driving scenario involving ego-vehicle (blue) and two obstacle-vehicles (yellow) traveling at different speeds.

graph \mathcal{R}_N . The nodes $i \in \mathcal{I}$ that were determined to be unsafe are marked by red dots in Figure 1. Executing the algorithm in MATLAB on a 2014 MacBook Pro with a 2.5 GHz i7 processor and 16 GB of RAM, it required ≈ 10 ms to perform the collision checks. The remaining safe nodes (marked by blue plus-signs in Figure 1) comprise the search graph \mathcal{G} used by Algorithm 1. For this scenario, searching the graph \mathcal{G} required ≈ 30 ms to find the sequence of references $r_{\sigma(0)}^\infty, \dots, r_{\sigma(N)}^\infty$ shown in Figure 1, where the initial state $x(t_0) \in \mathcal{O}_{\sigma(0)}$ is contained in the initial safe-set $\mathcal{O}_{\sigma(0)}$. Thus, the entire algorithm required ≈ 40 ms which is $\approx 500 \times$ faster than the planning horizon $N\Delta\tau = 20$ s.

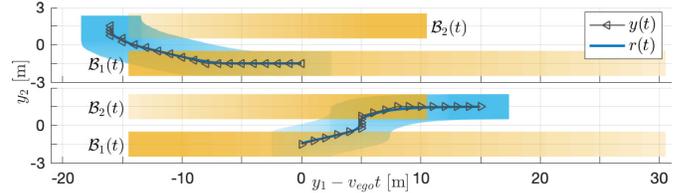


Fig. 3. Vehicle path relative to nominal trajectory $v_{ego}t$ of the ego-vehicle without disturbances where $v_{ego} = 24$ m/s; robust invariant-set motion-planner on top and nominal planner on bottom. Yellow regions indicate spatial extent of the obstacle vehicles $\mathcal{B}_j(t)$ and blue region indicates spatial extent of the ego-vehicle over time where darkening of sets indicates passage of time.

The motion $y(t)$ of the vehicle relative to its nominal constant-velocity trajectory $y_1(t) = v_{ego}t$ while tracking the reference $r(t) = r_{\sigma(t)}^\infty$ is shown in Figure 3. For comparison, Figure 3 also shows the motion of the vehicle tracking a path produced by a non-robust invariant-set motion-planner. The nominal and robust invariant-set motion-planners produce qualitatively different behaviors; The nominal motion-planner increases the speed of the ego-vehicle and changes lanes in front of the second obstacle-vehicle $\mathcal{B}_2(t)$. Whereas, the robust motion-planner slows the ego-vehicle and changes lanes behind the second obstacle-vehicle $\mathcal{B}_2(t)$. Slowing the vehicle appears as backward motion in Figure 3 which shows the longitudinal position $y_1(t)$ relative $y_1(t) - v_{ego}t$ to a constant velocity trajectory $v_{ego}t$.

The advantage of the path chosen by the robust motion-planner is apparent when there are disturbances $w(t) \in \mathcal{W}$ acting on the ego-vehicle. Figure 4 shows the same scenario as Figure 3, but with a random rate-bounded disturbance $w(t) \in \mathcal{W}$ perturbing the vehicle. The disturbances degrade the tracking performance as shown in Figure 5, which shows the lateral tracking performance for the nominal motion-planner. During 4 time-intervals [7, 9], [8, 9], [9, 10], and [11, 12], the disturbance $w(t) \in \mathcal{W}$ forced the system state $x(t) \notin \mathcal{O}_{\sigma(t)}$ to leave the non-robust positive invariant sets. During one of these intervals [9, 10], the obstacle vehicle $\mathcal{B}_1(t)$ was nearby causing a collision at time $t = 9.7 \in [9, 10]$, as shown in Figure 4 (bottom).

In contrast, since the robust invariant-set motion-planner uses RPI sets, the disturbances $w(t) \in \mathcal{W}$ cannot force the system state out of the sets. Thus, $x(t) \in \mathcal{O}_{\sigma(t_k)}$ for all $t \in [t_k, t_{k+1}]$. Therefore, since the system state $x(t)$ are always inside the specified neighborhood $\mathcal{O}_{\sigma(t_k)}$ during the specified time interval $[t_k, t_{k+1}]$, we can guarantee that the ego-vehicle will not collide with an obstacle, despite the presence of disturbances $w(t) \in \mathcal{W}$.

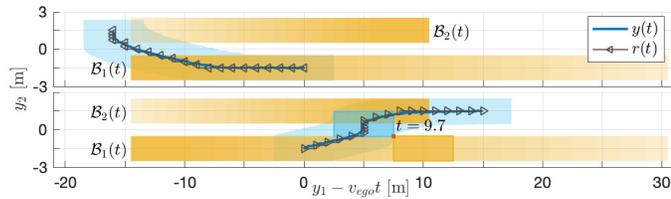


Fig. 4. Vehicle paths with disturbances; robust invariant-set motion-planner on top and nominal planner on bottom. Nominal planner (bottom) does not account for additional track error $e(t) = y(t) - r(t)$ due to disturbances $w(t)$ which causes a collision at $t = 9.7$ seconds.

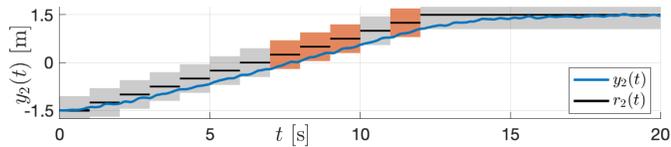


Fig. 5. Lateral tracking performance. Black lines are reference $r(t) = r_{\sigma(t)}^{\infty}$ and gray/red region are (non-robust) PI sets. Red regions indicate when the random rate-bounded disturbances $w(t) \in \mathcal{W}$ force the system trajectory $y(t) \notin \mathcal{O}_i$ to leave the (non-robust) PI sets \mathcal{O}_i .

V. CONCLUSIONS

This paper adapted the invariant-set motion-planner for systems with persistently varying disturbances and parametric model uncertainty. This required reformulating the dynamics in velocity-form so that the system output appeared directly in the velocity-state. The velocity-form was used to find an invariant subset of the state-space where it is safe to track a particular reference. These safe-sets were connected using a more conservative connection-rule that accounts for additional reference tracking error due to the presence of disturbances. For scenarios with moving obstacles, we were able to bound the time required to transition from safely tracking one reference to another. The robust motion-planner was demonstrated for a highway driving scenario where the robust invariant-set motion-planner was able to prevent collisions by incorporating knowledge of the closed-loop longitudinal and lateral vehicle dynamics into the search graph.

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 *Conf. 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] 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.
- [6] J. Kuffner, S. Kagami, K. Nishiwaki, M. Inaba, and H. Inoue, "Dynamically-stable motion planning for humanoid robots," *Autonomous Robots*, 2002.
- [7] A. Perez, S. Karaman, A. Shkolnik, E. Frazzoli, S. Teller, and M. Walter, "Asymptotically-optimal path planning for manipulation using incremental sampling-based algorithms," in *Intelligent Robots and Systems*, 2011.
- [8] J. H. Reif, "Complexity of the mover's problem and generalizations," in *Conference on Foundations of Computer Science*, 1979.
- [9] S. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.
- [10] S. LaValle and J. Kuffner, "Randomized kinodynamic planning," *Int. J. of Robotics Research*, 2001.
- [11] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, 2011.
- [12] B. D. Luders, S. Karaman, and J. P. How, "Robust sampling-based motion planning with asymptotic optimality guarantees," in *Guidance, Navigation, and Control Conference*, 2013.
- [13] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *Int. Conf. on Robotics and Automation*, 2011.
- [14] N. A. Melchior and R. Simmons, "Particle rrt for path planning with uncertainty," in *Int. Conf. on Robotics and Automation*, 2007.
- [15] M. Althoff, D. Althoff, D. Wollherr, and M. Buss, "Safety verification of autonomous vehicles for coordinated evasive maneuvers," in *Intelligent Vehicles Symposium*, 2010.
- [16] M. Althoff and J. Dolan, "Online verification of automated road vehicles using reachability analysis," *Transactions on Robotics*, 2014.
- [17] M. Koschi and M. Althoff, "SPOT: A tool for set-based prediction of traffic participants," in *Intelligent Vehicles Symposium*, 2017.
- [18] F. Blanchini, F. Pellegrino, and L. Visentini, "Control of manipulators in a constrained workspace by means of linked invariant sets," *Int. J. of Robust and Nonlinear Control*, 2004.
- [19] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts, "LQR-trees: Feedback motion planning via sums-of-squares verification," *Int. J. of Robotics Research*, 2010.
- [20] 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.
- [21] 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.
- [22] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, 2005.
- [23] F. Blanchini and S. Miani, *Set-Theoretic Methods in Control*. Birkhäuser, 2007.
- [24] S. Boyd, L. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Society for Industrial and Applied Mathematics, 1994.
- [25] H. Khalil, *Nonlinear Systems*. Prentice Hall, 2002.
- [26] G. Betti, M. Farina, and R. Scattolini, "A robust MPC algorithm for offset-free tracking of constant reference signals," *Trans. on Automatic Control*, 2013.
- [27] D. P. Bertsekas, *Network Optimization: Continuous and Discrete Models*. Athena Scientific, 1998.
- [28] R. Rajamani, *Vehicle Dynamics and Control*. Springer, 2010.