Safe Online Planning in Unknown Nonconvex Environments with Implicit Controlled Invariant Sets [⋆]

Zexiang Liu [*] **Necmiye Ozay** [*]

[*] *Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA (e-mails: {zexiang, necmiye}@umich.edu).*

Abstract: This work extends recent results on lifted controlled invariant sets for discrete-time controllable linear systems to non-convex safe sets, consisting of a union of polytopes. First, we show how to construct a closed-form expression of an invariant set in an extended state and input space using additional binary variables. A more efficient encoding is also provided albeit with additional conservativeness. By avoiding the projection of this set onto the original state space, we use this implicit representation in the lifted space as a supervisor in an online planning context where the safe set grows as new information about the environment is sensed. An example with a double-integrator model with a LiDAR sensor is used to demonstrate the overall approach.

Keywords: invariant sets, safe planning

1. INTRODUCTION

For systems, such as self-driving vehicles, mobile robots and airplanes, it is crucial to synthesize controllers to guarantee safety. For instance, for a self-driving vehicle, the vehicle needs to stay within the lane all the time unless it is overtaking other vehicles; or for a wheeled robot navigating a building, the robot needs to avoid any obstacles and other moving objects in the environment and reach its goal area. For these two control tasks, we identify two common features of the safety constraints in real-world applications: (a) The safety constraints can vary over time; (b) The safety constraints can be nonconvex.

Controlled invariant sets are the standard means to deal with the safety constraints in controller synthesis. However, for discrete-time linear systems, the standard iterative method of computing the maximal controlled invariant sets (see Bertsekas (1972); Vidal et al. (2000)) is (a) time consuming, making its online computation infeasible (b) only computationally tractable for convex safety constraints. Therefore, there is a gap between the safety constraints encountered in the real world and the safety constraints standard invariant set methods can handle.

Recent work addressing computational efficiency includes Anevlavis and Tabuada (2019, 2020); Anevlavis et al. (2021)), where a novel method is proposed to compute a controlled invariant set in two steps: First, an implicit controlled invariant set in a lifted space is constructed. Second, the implicit controlled invariant set is projected onto a subspace, where the projection is controlled invariant. Building on this result, we propose two methods in this paper to compute implicit controlled invariant sets for nonconvex safety constraints. More specifically, the main contributions of this work include:

i. We propose a novel method to compute implicit controlled invariant sets within nonconvex safe sets for discrete-time linear systems subject to disturbances (Algorithm 1 in Section 3). The projection of the implicit controlled invariant sets onto a subspace is proven to be controlled invariant.
ii. We extend the L-invariance formulation in our previous work Anevlavis et al. (2021) for nonconvex safe sets (Algorithm 2 in Section 3). We show that Algorithm 2 is more conservative than Algorithm 1, but Algorithm 2 is more scalable.
iii. Inspired by the safe navigation problem in Bajcsy et al. (2019), we demonstrate the use of our implicit controlled invariant sets in the task of navigating a robot through an unknown environment. The safety constraints are generated and updated online based on the LiDAR data. The effectiveness of our method is illustrated by simulations (Section 4).

For the related works dealing with nonconvex constraints: Bajcsy et al. (2019) propose a HJ-reachability-based method that enables online computation of the backward reachable set of the unsafe regions, whose complement is a controlled invariant set. This framework can work with nonconvex safe sets, but the HJ-reachability-based method works for continuous-time systems and the backward reachable set can only be calculated approximately. Wang et al. (2019) compute the maximal invariant sets for discrete-time linear systems with nonconvex smooth

[⋆] This work was supported by NSF Award CNS-1931982 and by the Automotive Research Center, a US Army Center of Excellence for modeling and simulation of ground vehicles, led by the University of Michigan under agreement W56HZV-19-2-0001. DISTRIBUTION A. Approved for public release; distribution unlimited. OPSEC #: 5192 constraints, but their method is only applicable to autonomous systems without control and is potentially time-consuming as it solves a series of semi-definite programs. Li et al. (2020) compute controlled invariant sets for the class of nonconvex safe sets whose complement is a polytope. The main idea is similar to the standard iterative method in Bertsekas (1972) but it customizes the algorithm for the specific class of nonconvex safe sets. Compared with our method, this method cannot handle more general nonconvex safe sets and it is potentially as time-consuming as the standard method. A broader class of works related to nonconvex constraints in control can be found in the literature for online path planning, where instead of computing a controlled invariant set, the efforts are on generating safe trajectories that do not collide with the unsafe regions. For instance, Liu et al. (2017) treat the moving pedestrians as the unsafe regions, which leads to a nonconvex time-varying safe set, and verifies the safe motions online based on reachability analysis.

Notation: For a vector x, x_i is the i th entry of x and $|x|$ is the absolute value of x defined element-wise. With a slight abuse of notation, the cardinality of a set U is denoted by $|U|$. The concatenation of vectors $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$ is denoted by $(x, y) \in \mathbb{R}^{n+m}$. The summation of a point x and a set Y is defined by $x + Y = \{x + y \mid y \in Y\}$; the summation and difference of two sets X and Y are defined by $X + Y = \{x + y \mid x \in X, y \in Y\}$ and $X - Y = \{x \mid x + Y \subseteq X\}$. The linear transformation of a set X under matrix T is denoted by $TX = \{Tx \mid x \in X\}$. The projection of a set X in $\mathbb{R}^n$ onto the first m coordinates, $m < n$, is denoted by $Proj_{1:m}(X) = \{(x_1, \cdots, x_m) \mid (x_1, \cdots, x_m, \cdots, x_n) \in X\}$. The Cartesian product of sets X and Y is denoted by $X \times Y$, or X^2 when $Y = X$.

2. PRELIMINARIES

We consider a discrete-time linear system

$$\Sigma : x(t + 1) = Ax(t) + Bu(t) + d(t), \tag{1}$$

with state $x \in \mathbb{R}^n$, input $u \in \mathbb{R}^m$ and disturbance $d \in D \subseteq \mathbb{R}^n$. The set D encodes the constraints on the disturbance d, represented by a polytope in $\mathbb{R}^n$.

Assumption 1. The system matrix A is nilpotent. That is, there exists a non-negative integer h, $h \leq n$, such that $A^h = 0$.

Remark 1. Any controllable system with state and input constraints can be equivalently converted to a linear system as in (1) with the matrix A nilpotent and the input u unconstrained. See Section II of Anevlavis and Tabuada (2019) and Remark 1 of Anevlavis et al. (2021) for the details.

The safety constraints imposed on the state x is represented by a union $S = \cup_{i=1}^{N} S_i$ of polytopes $S_i \subseteq \mathbb{R}^n$ for i from 1 to N. We call S the safe set of the system Σ.

A set $C \subseteq S$ is a *controlled invariant set* of Σ within the safe set S if for all $x \in C$, there exists an input $u \in \mathbb{R}^m$ such that $Ax + Bu + D \subseteq S$.

Problem statement: Given the system Σ, compute a controlled invariant set of Σ within the safe set S.

Given the system Σ, a control signal $u : \mathbb{N} \to \mathbb{R}^m$ and the initial state $x(0)$, the *k-step reachable set* $R_u^k(x(0))$ is the set of all possible states $x(k)$ at time k under the input sequence $(u(t))_{t=0}^{k-1}$ and an arbitrary disturbance sequence $(d(t))_{t=0}^{k-1}$ in D^k, which can be explicitly expressed as follows:

$$R_u^k(x(0)) = A^k x(0) + \sum_{t=0}^{k-1} A^{k-t-1}\left[Bu(t) + D\right]. \tag{2}$$

We define $\mathbf{x}_k(x, u) = A^k x + \sum_{t=0}^{k-1} A^{k-t-1} Bu(t)$ and $D_k = \sum_{i=0}^{k-1} A^i D$. Since $A^h = 0$, $D_k = D_h$ for all $k \geq h$. The expression of $R_u^k(x(0))$ can be rewritten as

$$R_u^k(x(0)) = \mathbf{x}_k(x(0), u) + D_k. \tag{3}$$

Note that the first term $\mathbf{x}_k(x(0), u)$ above is the nominal state of system at time k under control inputs $(u(t))_{t=0}^{k-1}$ when the disturbance term is equal to 0; the second term D_k is the set of derivations from the nominal state $\mathbf{x}_k(x_0, u)$ at time k due to the disturbance in D. When $k = 0$, we adopt the convention that $D_0 = \{0\}$ and $R_u^0(x(0)) = \mathbf{x}_0(x(0), u) = x(0)$.

3. CONTROLLED INVARIANCE IN NONCONVEX SETS

In this section, we first develop a novel method that represents an implicit controlled invariant set by a set of mixed-integer linear inequalities in a lifted space. The projection of the implicit controlled invariant sets onto a lower dimensional subspace is controlled invariant. Different from our previous work Anevlavis et al. (2021), our first method does not specify any structure in the controller and works with nonconvex constraints. In the second part of this section, we provide a direct extension of Method 2 in Anevlavis et al. (2021) to nonconvex constraints. The second method is more conservative than the first method, as a periodic structure is imposed as one constraint of the controller. But we show that the second method is more scalable than the first method, due to the lower dimensionality of the lifted space.

The key idea of both methods is to restrict the input u within a finite subset $\widehat{U} = \{u_1, u_2, \cdots, u_L\}$ of $\mathbb{R}^m$. The cardinality L of $\widehat{U}$ is a design parameter selected by users. Due to the finiteness of the input set $\widehat{U}$ and the nilpotency of the matrix A, given any initial state $x_0 \in \mathbb{R}^n$, the total number of the distinct k-step reachable sets $R_u^k(x_0)$ is finite, revealed by the following theorem.

Theorem 1. Given any control signal $u : \mathbb{N} \to \widehat{U}$, the cardinality of the set $\{R_u^k(x_0)\}_{k=0}^{\infty}$ of k-step reachable sets for all $k \geq 0$ is less than or equal to $h + L^h$.

Proof. By (3), the reachable set $R_u^k(x_0)$ is the sum of $\mathbf{x}_k(x(0), u)$ and D_k. Since $u(t) \in \widehat{U}$ for all t and $A^k = 0$ for all $k \geq h$, we can verify that $\{\mathbf{x}_k(x(0), u)\}_{k=0}^{\infty}$ and $\{D_k\}_{k=0}^{\infty}$ are both finite sets. Thus, the set of all the reachable sets are finite. The cardinality of $\{R_u^k(x_0)\}_{k=0}^{\infty}$ can be easily obtained by solving a combinatorial problem. ■

Since the set of all reachable sets is finite, given the initial state and any control signal with codomain restricted to $\widehat{U}$, we can check if the closed-loop trajectory stays in the safe set indefinitely by checking finitely many set containments. That enables our closed-form construction of controlled invariant sets in two moves: We first construct a set C_{lift} in a high-dimensional space. C_{lift} contains all pairs of initial state x_0 and input set $\widehat{U}$ for which there exists a control signal $u : \mathbb{N} \to \widehat{U}$ such that the closed-loop trajectory stays within the safe set indefinitely. Then, we show that the projection of C_{lift} onto its first n coordinates is a controlled invariant set of Σ within the safe set S.

Let us define the lifted set formally: C_{lift} is the set of points $(x_0, u_1, \cdots, u_L)$ satisfying that there exists a control signal $u : \mathbb{N} \to \{u_i\}_{i=1}^L$ such that the reachable states $R_u^k(x_0)$ at time k is contained by the safe set S for all $k \geq 0$. That is,

$$\begin{aligned} C_{lift} =& \{(x_0, u_1, \cdots, u_L) \mid \widehat{U} = \{u_i\}_{i=1}^L, \\ & \exists u : \mathbb{N} \to \widehat{U}, R_u^k(x_0) \subseteq S, \forall k \geq 0\}. \end{aligned} \tag{4}$$

Suppose that we can write C_{lift} in closed form. If we project C_{lift} onto the first n coordinates, the projection consists of the states where there exist a finite set $\widehat{U} \subseteq U$ with $|\widehat{U}| = L$ and a control signal $u : \mathbb{N} \to \widehat{U}$ ensuring $x(k) \in S$ for all $k \geq 0$. That is, the projection belongs to a controlled invariant set within S. It is shown by the following theorem that the projection is actually controlled invariant by itself.

Theorem 2. The projection $Proj_{1:n}(C_{lift})$ of C_{lift} onto the first n coordinates is a controlled invariant set of Σ within S.

Proof. By the construction of C_{lift}, a state $x \in Proj_{1:n}(C_{lift})$ if and only if there exists a controller $u_x : \mathbb{N} \to \widehat{U}$ such that all future states reachable from x under the control of u_x are within S. Let x' be a future state reached from x under u_x at time t. It is clear that all the future states reached by x' under the control of u_x after time t stay within S. Thus, $x' \in Proj_{1:n}(C_{lift})$ and for any $x \in Proj_{1:n}(C_{lift})$, there exists a u such that for all $x' \in Ax + Bu + ED$, $x' \in Proj_{1:n}(C_{lift})$. That is, $Proj_{1:n}(C_{lift})$ is a controlled invariant set within S. ■

So far, we know that the projection of C_{lift} is a controlled invariant set, but we do not know how to construct C_{lift} in a tractable way. The following theorem provides a useful observation for constructing C_{lift}.

Theorem 3. The set C_{lift} defined in (4) is equal to the set of points $(x_0, u_1, \cdots, u_L)$ for each of which there exists a control signal $u : \mathbb{N} \to \widehat{U}$ such that $R_u^k(x_0) \subseteq S$ for all k from 0 to $h + L^h$. That is,

$$\begin{aligned} C_{lift} =& \{(x_0, u_1, \cdots, u_L) \mid \widehat{U} = \{u_i\}_{i=1}^L, \exists u : \mathbb{N} \to \widehat{U}, \\ & R_u^k(x_0) \subseteq S, \forall 0 \leq k \leq h + L^h\}. \end{aligned} \tag{5}$$

Proof. We denote the right hand set of (5) by C'_{lift}. By definition of C_{lift} in (4), $C'_{lift} \supseteq C_{lift}$. Thus, it is left to show that $C'_{lift} \subseteq C_{lift}$.

Let $(x_0, u_1, \cdots, u_L) \in C'_{lift}$. Then there exists an $u : \mathbb{N} \to \widehat{U}$ guaranteeing $R_u^k \subseteq S$ for k from 0 to $h + L^h$. By Theorem 1, the number of all the possible reachable sets are less than or equal to $h + L^h$. Thus, we know that the sequence of reachable sets $(R_u^k(x_0))_{k=0}^{h+L^h}$ must repeat at some time between 0 and $h+L^h$. That is, there exists k_1, $k_2 \in [0, h + L^h]$ such that $k_1 < k_2$ and $R_u^{k_1}(x_0) = R_u^{k_2}(x_0)$. Then, we can construct a new controller $v(t) : \mathbb{N} \to \widehat{U}$ satisfying that $v(t) = u(t)$ for t from 0 to $k_1 - 1$ and $v(t) = u(k_1 + \delta(t))$ for $t \geq k_1$, where $\delta(t)$ is equal to the modulo $(t - k_1) \bmod (k_2 - k_1)$. It is easy to check that $v(t)$ guarantees the k-step reachable set of x_0 for all $k \geq 0$ belong to the set $\{R_u^k(x_0)\}_{k=0}^{k_2}$. Thus, by definition of C_{lift} in (4), $(x_0, u_1, \cdots, u_L\} \in C_{lift}$. ■

According to (5) in Theorem 3, C_{lift} can be encoded by a set of mixed integer linear inequality constraints. We first introduce the mixed integer formulation of each sub-constraint in (5) and then glue all together.

Define the constant $M = +\infty$ (or a large enough number), and denote the vector with all entries equal to 1 by $\mathbf{1}$. To encode the constraints $u(k) \in \widehat{U}$ for k from 0 to $h+L^h-1$, we introduce binary variables $a_{k,i} \in \{0, 1\}$ for i from 1 to L. Then, $u(k) \in \widehat{U}$ if and only if there exists $a_{k,i} \in \{0, 1\}$ for i from 1 to L such that

$$\begin{aligned} & |u(k) - u_i| \leq (1 - a_{k,i})M\mathbf{1}, \\ & \sum_{i=1}^{L} a_{k,i} \geq 1 \text{ (or } \sum_{i=1}^{L} a_{k,i} = 1), \end{aligned} \tag{6}$$

Next, note that $R_u^k(x_0) \subseteq S$ is equivalent to $\mathbf{x}_k(x_0, u) \in S - D_k$, where $\mathbf{x}_k(x_0, u)$ is a linear function of x_0, $u(0)$, ..., $u(k-1)$ and $S - D_k$ is a union of polytopes in $\mathbb{R}^n$. Suppose $S - D_k = \cup_{j=1}^{N_k} S_{j,k}$ where $S_{j,k} = \{x \mid H_{j,k}x \leq h_{j,k}\}$ is a polytope in $\mathbb{R}^n$ for each j from 1 to N_k. Then, $R_u^k(x_0) \subseteq S$ if and only if there exists $s_{k,j} \in \{0, 1\}$ for j from 1 to N_k such that

$$\begin{aligned} & H_{j,k}\mathbf{x}_k(x_0, u) \leq h_{j,k} + (1 - s_{k,j})M\mathbf{1}, \\ & \sum_{j=1}^{N_k} s_{k,j} \geq 1 \text{ (or } \sum_{j=1}^{N_k} s_{k,j} = 1). \end{aligned} \tag{7}$$

Denote the vector consisting of variables $a_{k,i}$ for k from 0 to $h+L^h-1$ and i from 1 to L by α, and denote the vector consisting of variables $s_{k,j}$ for k from 0 to $h + L^h$ and j from 1 to N_k by ξ. Also, denote $\mu = (u(0), u(1), \cdots, u(h + L^h - 1))$. Then, by (6) and (7), we can construct the set C_{mix} of points $(x_0, u_1, \cdots, u_L, \mu, \alpha, \xi)$ satisfying the mixed integer constraints:

$$\begin{aligned} & \forall k, 0 \leq k \leq h + L^h - 1, \forall i, 1 \leq i \leq L : \\ & |u(k) - u_i| \leq (1 - a_{k,i})M\mathbf{1}, \ \sum_{i=1}^{L} a_{k,i} \geq 1; \\ & \forall k, 0 \leq k \leq h + L^h, \forall j, 1 \leq j \leq N_k : \\ & H_{j,k}\mathbf{x}_k(x_0, u) \leq h_{j,k} + (1 - s_{k,j})M\mathbf{1}, \ \sum_{j=1}^{N_k} s_{k,j} \geq 1. \end{aligned} \tag{8}$$

By the construction of C_{mix}, the closed-form expression of C_{lift} can be obtained by projecting of C_{mix} onto the first $n + mL$ coordinates, that is $C_{lift} = Proj_{1:(n+mL)}(C_{mix})$. Or if we project C_{mix} onto the first n coordinates, we obtain the controlled invariant set $Proj_{1:n}(C_{lift})$ directly.

Remark 2. If it is hard to find the polytopes $S_{j,k}$ such that $\cup_{j=1}^{N_k} S_{j,k} = S - D_k$, we can instead find $S_{j,k}$ such that the union of $S_{j,k}$ inner approximates $S - D_k$, that is $\cup_{j=1}^{N_k} S_{j,k} \subseteq S - D_k$. In this case, $Proj_{1:(n+mL)}(C_{mix})$ is an inner approximation of C_{lift}. Most importantly, it can be proven that the projection $Proj_{1:n}(C_{mix}) \subseteq Proj_{1:n}(C_{lift})$ is still a controlled invariant set within S. ■

By Remark 2, if $S = \cup_{j=1}^{N} S_j$ is the union of polytopes S_j, one feasible $S_{j,k}$ can be as simple as $S_j - D_k$ for j from 1 to N. The method of computing controlled invariant sets by projecting C_{mix} is summarized as Algorithm 1. Note

Algorithm 1 Full Input Encoding

```
for all $k$ from 0 to $h$ do
    Construct $S_{j,k}$ such that $\cup_{j=1}^{N_k} S_{j,k} = S - D_k$ (or $\cup_{j=1}^{N_k} S_{j,k} \subseteq S - D_k$)
end for
Construct $C_{mix}$ using constraints in (8).
return $Proj_{1:n}(C_{mix})$.
```

that C_{mix} is a polytope (since all constraints are given by linear inequalities) in space $\mathbb{R}^{n+m(L+h+L^h)} \times \{0,1\}^b$, where $b = (n+L^h)L + \sum_{k=1}^{h-1} N_k + (h+L^h-h+2)N_h$. The number of auxiliary variables introduced in C_{mix} is $m(h+L^h) + \sum_{k=1}^{h-1} N_k + (h+L^h-h+2)N_h$. Thus, the projection of C_{mix} does not scale well when L is large. To obtain a more tractable method, we introduce a simplification of C_{mix} inspired by the L-invariance formulation in Anevlavis et al. (2021).

The idea is very intuitive: When constructing C_{mix}, we introduce the binary variables $a_{k,i}$ and continuous variables $u(k)$ to encode all possible control signals u that map $\mathbb{N}$ to $\widehat{U}$. If we fix the structure of the control signal u, then we can get rid of $a_{k,i}$ and $u(k)$ and lower the dimension of the lifted set. One reasonable structure of u used in Anevlavis et al. (2021) is to circulate from u_1 to u_L recursively, that is, for $t \geq 0$,

$$u(t) = u_{cir}(t) := u_{(t \bmod L)+1}, \tag{9}$$

where (t mod L) is the remainder when t is divided by L. We denote this control signal by u_{cir}.

We define another lifted set C_L, similar to C_{lift}, but with this fixed control signal, that is,

$$C_L = \{(x_0, u_1, \cdots, u_L) \mid R^k_{u_{cir}}(x_0) \subseteq S, \forall k \geq 0\}. \tag{10}$$

Anevlavis et al. (2021) shows that the projection $Proj_{1:n}(C_L)$ of C_L onto the first n coordinates is controlled invariant [1].

Under this specific control signal u_{cir}, we can easily verify that

$$\{R^k_u(x_0)\}_{k=0}^{\infty} = \{R^k_u(x_0)\}_{k=0}^{h+L-1}. \tag{11}$$

Thus, C_L can be encoded by the following mixed integer linear inequalities: Let $s_{k,j}$ be binary variables for k from 0 to $h+L-1$ and j from 1 to N_k. Denote the vector consisting of all $s_{k,j}$ by ξ. Let $C_{mix,L}$ be the set of points $(x_0, u_1, \cdots, u_L, \xi)$ satisfying the mixed integer constraints:

$$\begin{aligned} &\forall k, 0 \leq k \leq h+L-1, \forall j, 1 \leq j \leq N_k : \\ &H_{j,k}\mathbf{x}_k(x_0, u_{cir}) \leq h_{j,k} + (1 - s_{k,j})M\mathbf{1}, \\ &\sum_{j=1}^{N_k} s_{k,j} \geq 1, \end{aligned} \tag{12}$$

where recall that $\mathbf{x}_k(x_0, u_{cir})$ is a linear function of x_0, u_1, ..., u_L. Note that there are just $\sum_{k=1}^{h-1} N_k + LN_h$ auxiliary variables, which is much less than the auxiliary variables introduced for constructing C_{mix}.

[1] Anevlavis et al. (2021) considers the case where S is convex. But the same argument easily extends to the case that S is nonconvex.

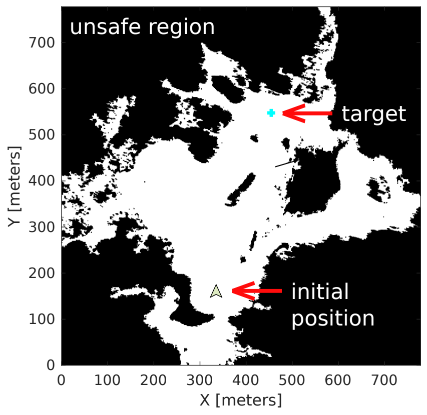


Fig. 1. The workspace of the robot: The initial position, the target position and the unsafe region are respectively indicated by the yellow arrowhead, the cyan point and the dark area.

Note that similar to Remark 2, if we compute $S_{j,k}$ whose union inner approximates $S - D_k$, $Proj_{1:n}(C_L)$ is still a controlled invariant set within S. We summarize this simplified method to compute controlled invariant set in Algorithm 2.

Algorithm 2 L-Invariance Input Encoding

```
for all $k$ from 0 to $h$ do
    Construct $S_{j,k}$ such that $\cup_{j=1}^{N_k} S_{j,k} = S - D_k$ (or $\cup_{j=1}^{N_k} S_{j,k} \subseteq S - D_k$)
end for
Construct $C_{mix,L}$ using constraints in (12).
return $Proj_{1:n}(C_{mix,L})$.
```

Note that Algorithm 2 can be extended for control signals other than u_{cir}, or even consider multiple different control signals by introducing more auxiliary binary variables as in Algorithm 1. The trade-off here is between the scalability and the size of the resulting invariant sets.

As a final note, we want to highlight that for certain applications, we do not have to project C_{mix} or $C_{mix,L}$ and thus avoid the computational cost of the projection operation. For example, if the goal is to find a safe input for some given state, it can be achieved by solving a mixed integer linear program with constraints in C_{mix} or $C_{mix,L}$. We apply this idea in the following section.

4. CASE STUDY

In this section, we apply the proposed method to a robot navigation problem: We want to navigate a robot from the initial position to the target region in the map shown by Figure 1. We assume that the map is unknown in the beginning and is built online based on LiDAR measurements. During the navigation process, the robot needs to avoid colliding with the obstacles, indicated by the dark area in Figure 1. This case study is inspired by the robot navigation problem in Bajcsy et al. (2019).

4.1 Robot dynamics

Many nonlinear robot systems can be transformed into multiple-integrator dynamics by feedback linearization, such as the unicycle model (see De Luca et al. (2001)). For this reason, we model a ground robot in the X-Y plane by a double-integrator model:

$$\Sigma_c : \begin{cases} \ddot{x} = u_1 \\ \ddot{y} = u_2, \end{cases} \tag{13}$$

where (x, y) is the position of the robot and u_1, u_2 are the control inputs. Denote $v_x = \dot{x}$ and $v_y = \dot{y}$. The states of Σ_c consist of (x, y, v_x, v_y). For simplicity, we assume that the system has no disturbance term. The continuous-time system in (13) is discretized with time step Δt, via forward Euler method. As the discretized system is controllable, our method is applicable according to Remark 1. Also, in order to handle the input constraints, we lift the discretized system to a system with an extended state space $(x, y, v_x, v_y, u_1, u_2)$ (the details can be found in the references listed in Remark 1).

4.2 The online safety constraints

The safe set consists of two parts: First, the absolute values $|u_i|$ of control inputs are bounded by a constant u_{max} for $i = 1, 2$. The absolute values $|v_x|$, $|v_y|$ of velocities are bounded by a constant v_{max}.

Second, we want to constrain (x, y) to stay within the obstacle-free region, shown by the white nonconvex area in Figure 1. To make the problem more challenging, we assume that the map is unknown to the robot, but a LiDAR sensor attached to the robot can sense the environment. The sensing range of the LiDAR is a disk with radius equal to 100 m, centered at the robot position (x, y). Suppose that $M(t) \subseteq R^2$ is the obstacle-free region confirmed by the LiDAR data up to time t. Note that $M(t) \subseteq M(t + 1)$ for all $t \geq 0$. Ideally, $M(t)$ should be the safety constraint on $(x(t), y(t))$ at time t. But, in practice the convex decomposition of $M(t)$ contains more and more pieces over time, which significantly increases the computation time. We denote $M_B(t)$ as the union of the 10 largest rectangles in $M(t)$ that contain $(x(t), y(t))$. To reduce the computation burden, we specify $M_B(t)$, instead of $M(t)$, as the safety constraint on (x, y) at time t.

Combining the above constraints, define

$$\begin{aligned} S(t) =& \{(x, y, v_x, v_y, u_1, u_2) \mid (x, y) \in M_B(t), |v_x| \leq v_{max}, \\ & |v_y| \leq v_{max}, |u_i| \leq u_{max}, i = 1, 2\} \end{aligned}$$

We use the set $S(t)$ as the safe set at time t, generated online based on the LiDAR measurements.

4.3 Control framework

The overall control framework is shown in Figure 2. At the initialization stage, a blank 2-dimensional occupancy grid map is created, and a reference trajectory is generated by the path planner assuming no obstacles.

At each time instant t, the map is updated based on the latest LiDAR measurements. Then, the path planner checks if the current reference trajectory collides with known obstacles in the current map. If a collision is detected, a new reference path that does not collide with any known obstacles is generated. Next, a nominal controller provides a candidate control input $(\overline{u}_1(t), \overline{u}_2(t))$ that tracks the reference path.

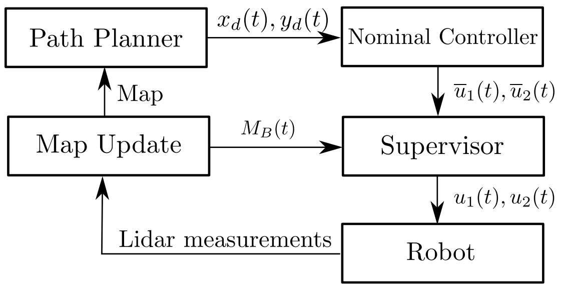


Fig. 2. The overall control framework

To supervise the candidate control input, we construct the obstacle-free region $M(t)$ and the safe set $S(t)$, defined in Section 4.2 based on the current map. Algorithm 2 takes the safe set $S(t)$ as input and returns the lifted set $C_{mix,L}(t)$. Given the robot state $(x(t), y(t), v_x(t), v_y(t))$ at time t and a time index $t' \leq t$, we define the set $U_{safe}(t, t')$ of safe inputs by

$$\begin{aligned} U_{safe}(t, t') = \{&(u_1, u_2) \mid |u_i| \leq u_{max}, i = 1, 2, \\ &(\overline{x}, \overline{y}, \overline{v}_x, \overline{v}_y) \in Proj_{1:4}(C_{mix,L}(t'))\}, \end{aligned} \tag{14}$$

where $\overline{x} = x(t) + v_x(t)\Delta t$, $\overline{y} = y(t) + v_y(t)\Delta t$, $\overline{v}_x = v_x(t) + u_1\Delta t$, and $\overline{v}_y = v_y(t) + u_2\Delta t$.

Assume that $U_{safe}(0, 0)$ is nonempty. The following theorem guarantees that there always exists a $t' \leq t$ such that $U_{safe}(t, t')$ is nonempty for all $t \geq 0$.

Theorem 4. If $U_{safe}(t, t')$ is nonempty and $(u_1(t), u_2(t)) \in U_{safe}(t, t')$, then $U_{safe}(t + 1, t')$ is nonempty.

Proof. (Sketch) Since $(u_1(t), u_2(t)) \in U_{safe}(t, t')$, by construction of $U_{safe}(t)$, $(x(t+1), y(t+1), v_x(t+1), v_y(t+1)) \in Proj_{1:4}(C_{mix,L}(t'))$. Note that $Proj_{1:6}(C_{mix,L}(t'))$ is a controlled invariant set within $S(t')$. Thus, it is easy to show that there exists $u_i(t + 1)$ with $|u_i(t + 1)| \leq u_{max}$ such that $(x(t + 2), y(t + 2), v_x(t + 2), v_y(t + 2)) \in Proj_{1:4}(C_{mix,L}(t'))$, and thus $U_{safe}(t + 1, t') \neq \emptyset$. ■

Finally, let $t^* \leq t$ be the latest time index such that $U_{safe}(t, t^*)$ is nonempty. The supervisor projects the candidate input $(\overline{u}_1(t), \overline{u}_2(t))$ onto the set $U_{safe}(t, t^*)$, resulting in the safe input $(u_1(t), u_2(t))$. The safe input $(u_1(t), u_2(t))$ is the actual control commands sent to the robot. As a direct corollary of Theorem 4, the robot satisfies all the safety constraints defined in Section 4.2 indefinitely under the proposed control framework.

Corollary 1. Assume that $U_{safe}(0, 0)$ is nonempty and $(x(0), y(0)) \in M(0)$ and $|v_x(0)|, |v_y(0)| \leq v_{max}$. The proposed control framework guarantees that $(x(t), y(t)) \in M(t)$, $|v_x(t)|, |v_y(t)| \leq v_{max}$ and $|u_i(t)| \leq u_{max}$ with $i = 1, 2$ for all $t \geq 0$.

Proof. According to Theorem 4, $U_{safe}(t, t^*)$ is nonempty for all $t \geq 0$. By definition of $U_{safe}(t, t^*)$, $(u_1(t), u_2(t)) \in U_{safe}(t, t^*)$ implies that (a) $|u_i(t)| \leq u_{max}$ with $i = 1, 2$ and (b) $(x(t + 1), y(t + 1)) \in M(t^*) \subseteq M(t + 1)$ and (c) $|v_x(t + 1)|, |v_y(t + 1)| \leq v_{max}$. (b) and (c) are due to the fact that $Proj_{1:4}(C_{mix,L}(t^*)) \subseteq M(t^*) \times [-v_{max}, v_{max}]^2$. ■

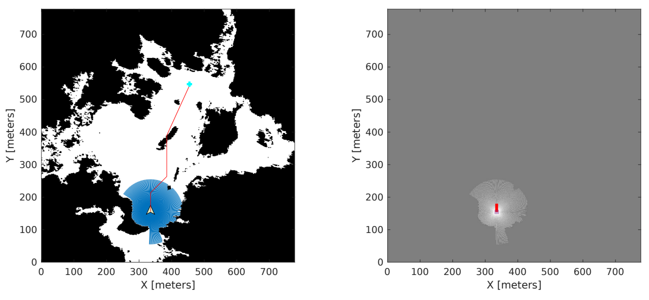
4.4 Simulation setup and results

We use a linear feedback controller as the nominal controller. The supervisor is implemented by a mixed integer programming that computes the projection of $(\overline{u}_1(t), \overline{u}_2(t))$ onto $U_{safe}(t, t^*)$ directly from $C_{mix,L}(t^*)$. The MATLAB Navigation Toolbox is used to simulate the LiDAR sensor, update the occupancy grid map and generate the reference path based on the A* algorithm.

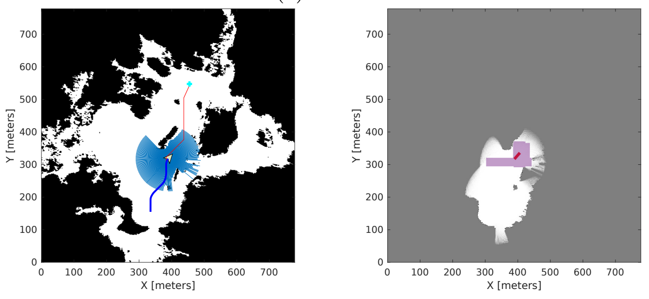
In the simulation, the parameters are $L = 10$, $\Delta t = 0.1s$, $v_{max} = 30m/s$, $u_{max} = 40m/s^2$. We run the simulation in MATLAB R2020b on a laptop with i7-8650 CPU and 16 GB memory. The mixed-integer programming is implemented via YALMIP (Lofberg (2004)) and solved by GUROBI (Gurobi Optimization (2020)). The average computation time for constructing the lifted set $C_{mix,L}(t)$ and solving the mixed-integer programming at each time step is $1.68s$. The average computation time shows the efficiency of our method, considering the safe set is nonconvex and being updated at every time step.

The simulation results are shown in Figure 3: The robot reaches the target region at $t = 20.4s$, and thanks to the supervisor, the robot satisfies the input and velocity constraints and stays within the time-varying safe region (white area in the right column of Figure 3) all the time. As a comparison, when the supervisor is disabled, the velocity constraint is violated at time $t = 2.7s$. The full simulation video can be found at `https://youtu.be/k1OdIr4YB8k` .

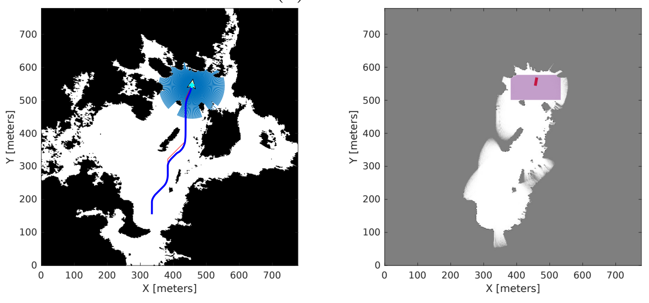
Acknowledgements: We would like to thank Tzanis Anevlavis and Paulo Tabuada of UCLA for many valuable discussions that inspired some of the ideas in this paper.



(a) $t = 0s$



(b) $t = 8s$



(c) $t = 20.4s$

Fig. 3. Simulation screenshots at times $t = 0s$, $8s$ and 20.4s. For figures on the left, the reference path and the robot's actual trajectory are shown by the red and blue curves; the disk of the blue rays is the LiDAR measurements; the position and direction of the arrowhead indicate the position and moving direction of the robot. For figures on the right, the white and grey areas indicate the obstacle-free region $M(t)$ and the unknown region; the purple boxes are the 10 largest boxes in $M(t)$ that contain the current robot position, whose union is $M_B(t)$.

REFERENCES

Anevlavis, T., Liu, Z., Ozay, N., and Tabuada, P. (2021). An enhanced hierarchy for (robust) controlled invariance. Proc. of ACC 2021.

Anevlavis, T. and Tabuada, P. (2019). Computing controlled invariant sets in two moves. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, 6248–6254. IEEE.

Anevlavis, T. and Tabuada, P. (2020). A simple hierarchy for computing controlled invariant sets. In *Proceedings of the 23rd International Conference on Hybrid Systems: Computation and Control*, 1–11.

Bajcsy, A., Bansal, S., Bronstein, E., Tolani, V., and Tomlin, C.J. (2019). An efficient reachability-based framework for provably safe autonomous navigation in unknown environments. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, 1758–1765. IEEE.

Bertsekas, D. (1972). Infinite time reachability of state-space regions by using feedback control. *IEEE Transactions on Automatic Control*, 17(5), 604–613.

De Luca, A., Oriolo, G., and Vendittelli, M. (2001). Control of wheeled mobile robots: An experimental overview. In *Ramsete*, 181–226. Springer.

Gurobi Optimization, L. (2020). Gurobi optimizer reference manual. URL `http://www.gurobi.com`.

Li, N., Han, K., Girard, A., Tseng, H.E., Filev, D., and Kolmanovsky, I. (2020). Action governor for discrete-time linear systems with non-convex constraints. *arXiv preprint arXiv:2005.08358.*

Liu, S.B., Roehm, H., Heinzemann, C., Lütkebohle, I., Oehlerking, J., and Althoff, M. (2017). Provably safe motion of mobile robots in human environments. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1351–1357. IEEE.

Lofberg, J. (2004). Yalmip: A toolbox for modeling and optimization in matlab. In *2004 IEEE international conference on robotics and automation (IEEE Cat. No. 04CH37508)*, 284–289. IEEE.

Vidal, R., Schaffert, S., Lygeros, J., and Sastry, S. (2000). Controlled invariance of discrete time systems. In *International Workshop on Hybrid Systems: Computation and Control*, 437–451. Springer.

Wang, Z., Jungers, R.M., and Ong, C.J. (2019). Computation of the maximal invariant set of discrete-time systems subject to quasi-smooth non-convex constraints. *arXiv preprint arXiv:1912.09727.*