2016 IEEE 55th Conference on Decision and Control (CDC)
ARIA Resort & Casino
December 12-14, 2016, Las Vegas, USA

On the Construction of Safe Controllable Regions for Affine Systems with Applications to Robotics

Mohamed K. Helwa and Angela P. Schoellig

Abstract— **This paper studies the problem of constructing in-block controllable (IBC) regions for affine systems. That is, we are concerned with constructing regions in the state space of affine systems such that all the states in the interior of the region are mutually accessible through the region's interior by applying uniformly bounded inputs. We first show that existing results for checking in-block controllability on given polytopic regions cannot be easily extended to address the question of constructing IBC regions. We then explore the geometry of the problem to provide a computationally efficient algorithm for constructing IBC regions. We also prove the soundness of the algorithm. Finally, we use the proposed algorithm to construct safe speed profiles for fully-actuated robots and for ground robots modeled as unicycles with acceleration limits.**

I. Introduction

Given the "safety first" engineering concept, there is an urgent need for developing mathematical foundations for control design under safety constraints. This is particularly relevant for safety-critical systems, in which violating the safety constraints may lead to severe consequences. Optimal/predictive controllers have received special interest for decades since they optimize the system's behavior, while respecting given, hard safety constraints [1], [25]. This is typically achieved by solving a constrained optimization problem at each sampling instant. Nevertheless, there is still a critical need for checkable conditions which define when we can fully control a system under given safety constraints. Such conditions ensure that all the optimal accessibility problems within given safety constraints are feasible.

We recently introduced the study of in-block controllability (IBC) to formalize the study of controllability under safety constraints [12], [17]. The notion of IBC was first introduced for finite-state machines in [4], and was then extended to nonlinear systems on closed sets [5] and to automata [19]. In these three papers, the notion of IBC was used to build hierarchical system structures. However, these papers do not provide conditions for when the IBC property holds. In [12], [17], three easily checkable necessary and sufficient conditions are provided for IBC of affine systems on given polytopes. The conditions require solving linear programming (LP) problems at the vertices of the polytope. The notion of IBC was relaxed in [13] to the case where one can distinguish between soft and hard safety constraints, and the IBC conditions were extended in [9] to controlled switched linear systems. The notion of IBC is also used to build special partitions/covers of the state space of piecewise affine (PWA) hybrid systems/nonlinear systems, respectively, in which each region satisfies the IBC property. These partitions/covers are used to build hierarchical structures and systematically study mutual accessibility problems of these systems (see [14] and [15], respectively). A similar controllability study to IBC can be found in [8], where controllability of continuous-time linear systems under state and/or input constraints was studied under the assumption that the system transfer matrix is right invertible. Compared to the controlled invariance problem [2], IBC has the additional requirement of achieving mutual accessibility. This basic, additional property enables us to use IBC as a basis for building hierarchical control structures [14], [15].

However, in many practical cases, the given affine system is not IBC with respect to (w.r.t.) the given polytope. Hence, it would be important, from practical point of view, to construct the largest possible IBC region within the given polytope, which intuitively represents a large, safe region within which we can fully control the system. Constructing IBC regions is also useful for building the IBC partitions/covers in [14] and [15], respectively, as discussed above.

In this paper, we first show that while checking IBC on given polytopes is easy, building polytopes that satisfy the IBC property requires solving bilinear matrix inequalities (BMIs), which is NP-hard, in general (see [26]). Secondly, we explore the geometry of the problem, and provide a computationally efficient algorithm for constructing IBC regions, which avoids solving BMIs. This geometric approach was introduced in [16] for affine hypersurface systems, a special class of affine systems in which $m = n - 1$, where m is the number of inputs and n is the system dimension. In this paper, we extend the geometric study of [16] to a more general geometric case that can be satisfied for affine systems having $m \geq \frac{n}{2}$. We also prove the soundness of the algorithm. In our geometric study of IBC, we exploit some geometric tools used for the reach control problems (RCP), found in [3], [7], [10], [11]. Thirdly, we show how our proposed algorithm can be useful for constructing safe speed profiles for several classes of robotic systems, including fully-actuated robots and ground robots modeled as unicycles with acceleration limits (that is, constructing a safe speed range at each position of the robot). The proposed safe speed profiles are useful for robot speed scheduling algorithms [21]–[24]. In particular, if the speed scheduling algorithms limit the speeds to the proposed safe speed profiles, then safety of the robots can always be achieved by a feasible

The authors are with the Dynamic Systems Lab (www.dynsyslab.org), Institute for Aerospace Studies, University of Toronto, Canada. M. K. Helwa is also with the Electrical Power and Machines Department, Cairo University. (e-mails: mohamed.helwa@robotics.utias.utoronto.ca, schoellig@utias.utoronto.ca). This research was supported by NSERC grant RGPIN-2014-04634 and the Connaught New Researcher Award.

input and, for example, guarantee obstacle avoidance. One advantage of the proposed safe speed profiles is that they guarantee full controllability of the robots on the constructed position-speed regions. Moreover, the proposed algorithm ensures that any state in the constructed safe position-speed region is reachable from all other states in the safe region while always staying in the safe region. Hence, it would be useful to select the states of the reference trajectories from the proposed safe position-speed regions to ensure that they are reachable within the safety constraints.

The paper is organized as follows. Section II reviews IBC. In Section III, we study the problem of constructing IBC regions, and provide a computationally efficient algorithm. In Section IV, we provide applications of the proposed algorithm to robotic systems. Section V concludes the paper. For the geometric background, refer to Section II of [12].

Notation: Let $K \subset \mathbb{R}^n$ be a set. The closure of K is denoted by $\overline{K}$, the interior by K°, and the boundary by ∂K. For vectors $x,\ y \in \mathbb{R}^n$, $x \cdot y$ denotes the inner product of the two vectors. The notation $\|x\|$ denotes the Euclidean norm of x. The notation $\text{co}\{v_1, v_2, \dots\}$ denotes the convex hull of a set of points $v_i \in \mathbb{R}^n$. For subspaces $\mathcal{A},\ \mathcal{B},\ \mathcal{A}+\mathcal{B} := \{a+b\ :\ a \in \mathcal{A},\ b \in \mathcal{B}\}$.

II. In-Block Controllability

In this section we review in-block controllability (IBC). Consider the affine control system:

$$\dot{x}(t) = Ax(t) + Bu(t) + a\,, \qquad x(t) \in \mathbb{R}^n\,, \qquad (1)$$

where $A \in \mathbb{R}^{n\times n}$, $a \in \mathbb{R}^n$, $B \in \mathbb{R}^{n\times m}$, and $\text{rank}(B) = m$. Throughout the paper, we assume that the input $u : [0,\infty) \to \mathbb{R}^m$ is measurable and bounded on any compact time interval to ensure the existence and uniqueness of the solutions of (1) [6]. Let $\phi(x_0, t, u)$ be the trajectory of (1) under a control law u, with initial condition x_0 and evaluated at time t. Next, let $X := \text{co}\{v_1, \cdots, v_p\}$ denote an n-dimensional polytope, with vertex set $\{v_1, \cdots, v_p\}$, and *facets*, i.e., $(n-1)$-dimensional faces, $F_1, \cdots, F_r$. Let h_i denote the unit normal vector to F_i pointing outside X. An *n-dimensional simplex* is a special case of X with $p = n+1$.

Definition 2.1 (In-Block Controllability (IBC) (after [5])): Consider the affine control system (1) defined on an n-dimensional polytope X. We say that (1) is *in-block controllable (IBC)* w.r.t. X if there exists an $M > 0$ such that for all $x,\ y \in X^\circ$, there exist $T \geq 0$ and a control input u defined on $[0,T]$ such that (i) $\|u(t)\| \leq M$ and $\phi(x,t,u) \in X^\circ$ for all $t \in [0,T]$, and (ii) $\phi(x,T,u) = y$.

That is, the system is IBC w.r.t. X if all the states in the interior of X are mutually accessible through its interior using uniformly bounded inputs. We review below the main result on IBC. In [12], it was shown that for studying IBC, we can always apply a coordinate shift, and assume without loss of generality (w.l.o.g.) that we study a linear system

$$\dot{\tilde{x}}(t) = A\tilde{x}(t) + B\tilde{u}(t) \qquad (2)$$

on a new polytope $\tilde{X}$ with $0 \in \tilde{X}^\circ$. For notational convenience and w.l.o.g., we would call $\tilde{X}$, $\tilde{x}$, and $\tilde{u}$ just X, x, and u, respectively, in the rest of the paper. Let $J := \{1, \cdots, r\}$ be the set of indices of the facets of X, and $J(x) := \{j \in J\ :\ x \in F_j\}$ be the set of indices of the facets of X in which x is a point. We define the closed, convex *tangent cone* to X at x as $C(x) := \{y \in \mathbb{R}^n\ :\ h_j \cdot y \leq 0,\ j \in J(x)\}$. Recall that a polytope is *simplicial* if all its facets are simplices.

Theorem 2.1 ([12]): Consider the system (2) defined on an n-dimensional simplicial polytope X satisfying $0 \in X^\circ$. Then, the system (2) is IBC w.r.t. X if and only if

(i) (A, B) is controllable;

(ii) the so-called invariance conditions of X are solvable (that is, for each vertex $v \in X$, there exists $u \in \mathbb{R}^m$ such that $Av + Bu \in C(v)$);

(iii) the so-called backward invariance conditions of X are solvable (that is, for each vertex $v \in X$, there exists $u \in \mathbb{R}^m$ such that $-Av - Bu \in C(v)$). ■

In [12], it was shown that conditions (i)-(iii) of Theorem 2.1 are also necessary for IBC on non-simplicial polytopes. For given polytopes, both the invariance and the backward invariance conditions can be easily checked by solving an LP problem for each vertex of the polytope[1].

Remark 2.1: The definition of IBC can be easily tailored to the case when we have both state and input constraints. Suppose $u \in U \subset \mathbb{R}^m$, where U is a polytope having $0 \in U^\circ$. For this case, the system is IBC if every $x,\ y \in X^\circ$ are mutually accessible through X° using control inputs $u \in U$. Similarly, the definitions of invariance conditions and backward invariance conditions are adapted to restrict u to lie in U. It can be shown that for these tailored definitions, conditions (i)-(iii) of Theorem 2.1 remain necessary for IBC. Also, the proof of the sufficiency of conditions (i)-(iii) in this case is similar to the one in Section V of [12] under the mild assumption on U that for any $\bar{x} \in X$ satisfying $A\bar{x} \in \text{Im}\ (B)$, the image of B, there exists a $\bar{u} \in U^\circ$ such that $A\bar{x} + B\bar{u} = 0$. Details are omitted for brevity.

III. Construction of IBC Regions

In this section, we study the problem of constructing IBC regions for affine systems. Following [12], we know that w.l.o.g. the problem of studying IBC of an affine system can be transformed to studying a linear system on a new polytope X with $0 \in X^\circ$. Thus, we consider a linear system (2). Given the necessity of condition (i) of Theorem 2.1 for IBC, in our study of constructing IBC polytopes, we assume w.l.o.g. that (2) is controllable. We then construct around the origin an IBC polytope for (2).

Problem 3.1 (Construction of IBC Polytopic Regions): Given a controllable linear system (2), construct a polytope X such that $0 \in X^\circ$ and (2) is IBC w.r.t. X.

It can be easily shown that if (2) is IBC w.r.t. the polytope X using uniformly bounded inputs satisfying $\|u\| \leq M$, then it is also IBC w.r.t. $\lambda X := \{x \in \mathbb{R}^n\ :\ x = \lambda y,\ y \in X\}$, a

[1] The invariance conditions and the backward invariance conditions are only checked at the vertices of X since solvability of these conditions at the vertices implies by a simple convexity argument that they are solvable at all boundary points of X [7].

λ-scaled version of X, for every $\lambda > 0$ and using uniformly bounded inputs satisfying $\|u\| \leq \lambda M$.

While checking IBC on given polytopes is easy and incorporates solving LP problems, building IBC polytopic regions is considerably more difficult. Theorem 2.1 suggests that we build around the origin simplicial polytopes satisfying both the invariance and the backward invariance conditions. Two difficulties are faced here. First, to build a polytope X satisfying the invariance conditions (similar argument holds for the backward invariance conditions), we would need to select the vertices of X, v_i, the unit normal vectors to the facets of X, h_j, and the control inputs at the vertices, u_i, such that $h_j \cdot (Av_i + Bu_i) \leq 0$, for all $j \in J(v_i)$. Since h_j, v_i, and u_i are all unknowns in this case, we have a set of bilinear matrix inequalities (BMIs), the solving of which is in general NP-hard [26]. Second, even if one constructs a polytope X satisfying both the invariance conditions and the backward invariance conditions, one still needs to verify that X is simplicial since the proof of the sufficiency of Theorem 2.1 only holds for simplicial polytopes.

To face these difficulties, one can exploit available software packages for solving BMIs offline such as PENBMI [18]. Another possible approach is to use trial-and-error with the aid of Theorem 2.1. It is clear that these two approaches are computationally expensive, and for the second approach, there is no guarantee that one will eventually find the IBC polytope. Instead, in this paper, we explore the geometry of the problem, and try to provide a computationally efficient algorithm for building IBC polytopes that avoids solving BMIs. We initiated this geometric study in [16] for hypersurface systems with $m = n - 1$, and here we extend the study of [16] to a more general geometric case. To that end, let $\mathcal{B} := \text{Im}\ (B)$ be the image of B, and define the set of possible equilibria of (2):

$$\mathcal{O} := \{\ x \in \mathbb{R}^n\ :\ Ax \in \mathcal{B}\ \}. \tag{3}$$

At any point in $\mathcal{O}$, the vector field of (2) can vanish by proper selection of the input u. Also, if $x_0 \in \mathbb{R}^n$ is an equilibrium point of (2) under some input, then $x_0 \in \mathcal{O}$ [3]. It can be verified that $\mathcal{O}$ is closed, affine, and its dimension is $m \leq \kappa \leq n$ [10]. Notice that both $\mathcal{B}$ and $\mathcal{O}$ are properties of the system (2), and, as such, they can be calculated before constructing the polytope X.

For the geometric case $\mathcal{O} + \mathcal{B} = \mathbb{R}^n$, we provide a computationally efficient algorithm for constructing IBC polytopes. We now show that this geometric condition is more general than the condition $m = n - 1$ considered in [16]. If $m = n-1$, then the dimension of $\mathcal{O}$ is $n-1 \leq \kappa \leq n$ [10]. If $\kappa = n$, then $\mathcal{O} + \mathcal{B} = \mathbb{R}^n$ clearly holds. We then show that $\mathcal{O} + \mathcal{B} = \mathbb{R}^n$ holds for the case when $\kappa = n - 1$. We claim that $\mathcal{B}$ is not subset of $\mathcal{O}$. Otherwise, we have $Ax + Bu \in \mathcal{B} \subset \mathcal{O}$ for all $x \in \mathcal{O}$, and so $\mathcal{O}$ is an invariant set under any selection of the control input u, which contradicts controllability of (2). If $\mathcal{B}$ is not subset of $\mathcal{O}$, then we can identify a non-zero vector $b \in \mathcal{B}$ such that $b \notin \mathcal{O}$. Since $\kappa = n - 1$, then clearly $\mathcal{O} + \mathcal{B} = \mathbb{R}^n$. On the other hand, for the following linear system, $\mathcal{O} + \mathcal{B} = \mathbb{R}^n$ holds, while $m < n - 1$:

$$\dot{x}(t) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} u(t). \tag{4}$$

Indeed, since the dimension of $\mathcal{B}$ is m and the dimension of $\mathcal{O}$ is $m \leq \kappa \leq n$ [10], $\mathcal{O} + \mathcal{B} = \mathbb{R}^n$ may be achieved for systems having $m \geq \frac{n}{2}$ as in (4), which is a significant relaxation of the condition of [16]. We also consider our study as a milestone in studying the general case in the future. We start by reviewing two geometric results of [16].

Lemma 3.1 ([16]): Consider the linear system (2). For any polytope X, if $v \in \mathcal{O}$ is a vertex of X, then the invariance conditions and the backward invariance conditions of X are solvable at v. ■

Lemma 3.2 ([16]): Consider the linear system (2). For any polytope X, if $\mathcal{B} \cap C^\circ(v) \neq \emptyset$ at a vertex v of X, where $C^\circ(v)$ denotes the interior of $C(v)$, then the invariance conditions and the backward invariance conditions of X are solvable at v. ■

Since $\mathcal{B}$ and $\mathcal{O}$ are properties of the linear system and can be calculated before constructing the polytope X, Lemmas 3.1 and 3.2 suggest that we construct the polytope X such that the vertices of X lie on $\mathcal{O}$, or the subspace $\mathcal{B}$ dips into the interior of the tangent cones to the constructed polytope X at the vertices. This ensures that both the invariance conditions and the backward invariance conditions are solvable at the vertices of the constructed polytope X. However, as mentioned before, there is still the difficulty that the proof of the sufficiency of Theorem 2.1 was carried out in [12] only for simplicial polytopes, and so Theorem 2.1 may not apply. An extension of Theorem 2.1 is needed.

Theorem 3.3: Consider a controllable linear system (2) defined on an n-dimensional polytope X satisfying $0 \in X^\circ$. If for each vertex v of X, either $v \in \mathcal{O}$ or $\mathcal{B} \cap C^\circ(v) \neq \emptyset$, then the system (2) is IBC w.r.t. X.

Proof: By assumption and from Lemmas 3.1, 3.2, both the invariance and the backward invariance conditions are solvable at the vertices of X. Although the three conditions of Theorem 2.1 hold, X in our case is not necessarily simplicial, and so we cannot exactly follow the same sufficiency proof in [12] of Theorem 2.1. Indeed, the proof of Theorem 2.1 is divided into three parts. In the first part, the invariance conditions are used to construct a continuous piecewise linear (PWL) feedback, and under the assumption that X is simplicial, it is proved that all the trajectories initiated in X° eventually tend to $\mathcal{O}$ through X°, and so they reach close to $\mathcal{O}$ in finite time. Then, in the second part, controllability of (A, B) is used to construct a piecewise continuous input that makes the trajectories initiated nearby $\mathcal{O}$ slide along $\mathcal{O}$ inside X° towards $0 \in X^\circ$ in finite time. Third, using the backward invariance conditions and a similar argument to the first two parts, it is shown that one can steer the backward dynamical system $\dot{x} = -Ax - Bu$ from any state in X° to 0 in finite time through X° using uniformly bounded input. Equivalently, one can steer the system (2) from 0 to any final state in X° in finite time through X° using uniformly bounded input. One can see that the assumption that X is simplicial is used in [12] only in the first part of the proof to show that all trajectories initiated in X° tend to $\mathcal{O}$, and so our task is reduced to prove this part in our case for any polytope, not necessarily simplicial. The details of the proof are omitted from this conference paper for brevity. ■

We now provide under the geometric condition $\mathcal{O}+\mathcal{B}=\mathbb{R}^n$ a computationally efficient algorithm for constructing a polytope X such that $0 \in X^\circ$ and the vertices of X satisfy $v \in \mathcal{O}$ or $\mathcal{B} \cap C^\circ(v) \neq \emptyset$, which implies from Theorem 3.3 that the given system is IBC w.r.t. X. We then prove the soundness of the algorithm.

Algorithm 3.1:

Given: A controllable linear system (2) satisfying $\mathcal{O}+\mathcal{B}=\mathbb{R}^n$; Suppose $\mathcal{B}=\text{span}\{b_1,\cdots,b_m\}$, and $\{o_{m+1},\cdots,o_n\}$ are such that $o_k \in \mathcal{O}$ for all $k=m+1,\cdots,n$ and $\mathbb{R}^n=\text{span}\{b_1,\cdots,b_m,o_{m+1},\cdots,o_n\}$.

Objective: Construct an n-dimensional polytope X such that $0 \in X^\circ$ and the system (2) is IBC w.r.t. X.

Steps:

1) Construct an initial n-dimensional polytope P such that $0 \in P^\circ$, and let $\{v_1,\cdots,v_p\}$ denote the vertices of P.
2) Let $T := [b_1 \ \cdots \ b_m \ o_{m+1} \ \cdots \ o_n]$ and $T_\mathcal{O} := [0 \ \cdots 0 \ o_{m+1} \ \cdots \ o_n]$. For v_i, $i=1,\cdots,p$, calculate $\bar{o}_i = T_\mathcal{O} T^{-1} v_i$.
3) Select $\alpha > 1$, and define $\tilde{o}_i := \alpha \bar{o}_i$ for $i=1,\cdots,p$.
4) Define $X := \text{co}\{v_1,\cdots,v_p,\tilde{o}_1,\cdots,\tilde{o}_p\}$. ◁

Theorem 3.4: Consider a controllable linear system (2) satisfying $\mathcal{O}+\mathcal{B}=\mathbb{R}^n$. Then, Algorithm 3.1 always terminates successfully, and (2) is IBC w.r.t. X.

Proof: Since $\mathcal{O}+\mathcal{B}=\mathbb{R}^n$, one can always identify $o_{m+1},\cdots,o_n$ such that $o_k \in \mathcal{O}$ for all $k=m+1,\cdots,n$, and $\mathbb{R}^n=\text{span}\{b_1,\cdots,b_m,o_{m+1},\cdots,o_n\}$. Since T has linearly independent columns, it is invertible. Hence, one can always calculate $\bar{o}_i, \tilde{o}_i$, and then construct X. By construction, $0 \in P^\circ \subset X^\circ$. We now prove that the vertices of X satisfy $v \in \mathcal{O}$ or $\mathcal{B} \cap C^\circ(v) \neq \emptyset$. Let $c_i=(c_{i1},c_{i2},\cdots,c_{in}) := T^{-1}v_i$. It is straightforward to show $v_i=\sum_{j=1}^m c_{ij}b_j+\sum_{j=m+1}^n c_{ij}o_j$, $\sum_{j=1}^m c_{ij}b_j =: b_{v_i} \in \mathcal{B}$, and $\sum_{j=m+1}^n c_{ij}o_j \in \mathcal{O}$. From step 2, $\bar{o}_i = T_\mathcal{O} c_i = \sum_{j=m+1}^n c_{ij}o_j \in \mathcal{O}$. Thus, $v_i = b_{v_i}+\bar{o}_i$. Since $\mathcal{O}$ is affine and $0 \in \mathcal{O}$, $\tilde{o}_i := \alpha\bar{o}_i \in \mathcal{O}$. Notice that $\bar{o}_i \in \text{co}\{\tilde{o}_i, 0\}$, and if $\bar{o}_i \neq 0$, then $\tilde{o}_i \neq \bar{o}_i$. Since $\tilde{o}_i \in X$ and $0 \in X^\circ$, $\bar{o}_i \in X^\circ$. Now if v_i, $i \in \{1,\cdots,p\}$, is a vertex of X, then $v_i - b_{v_i} = \bar{o}_i \in X^\circ$ implies that $-b_{v_i} \in \mathcal{B}$ dips into the interior of the tangent cone to X at v_i, i.e. $-b_{v_i} \in \mathcal{B} \cap C^\circ(v_i) \neq \emptyset$. From Theorem 3.3, (2) is IBC w.r.t. X. ■

As discussed before, for any $\lambda > 0$, (2) is also IBC w.r.t. λX using λ-scaled inputs of the ones used to solve mutual accessibility problems on X°. This may be useful in two ways. First, if it is required to keep the system within given, hard safety constraints that form a region X_c around the origin, then one can first use Algorithm 3.1 to construct an IBC polytopic region X satisfying $0 \in X^\circ$, and then one can simply scale X such that $\lambda X \subset X_c$. Here, λX represents a safe region, within which we can fully control our system. Second, for the case of input constraints ($u \in U \subset \mathbb{R}^m$, where $0 \in U^\circ$), we can similarly scale X such that on λX, $\lambda < 1$, the IBC property is achieved using $u \in U$.

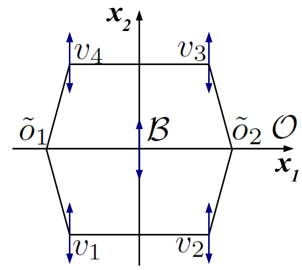

Fig. 1. The constructed IBC polytope X in Example 3.1.

Example 3.1: Consider the double integrator $\dot{x}_1 = x_2$, $\dot{x}_2 = u$. The system is evidently controllable. We have $\mathcal{O}=\{x \in \mathbb{R}^2 \ : \ x_2 = 0\}$, the x_1-axis, and $\mathcal{B}=\text{span}\{(0,1)\}$, the x_2-axis. Hence, $\mathcal{O}+\mathcal{B}=\mathbb{R}^2$. We follow the steps of Algorithm 3.1: (1) We construct $P=\text{co}\{v_1,\cdots,v_4\}$, where $v_1=(-0.8,-1)$, $v_2=(0.8,-1)$, $v_3=(0.8,1)$, and $v_4=(-0.8,1)$ [2]; (2) we have $b_1=(0,1)$, $o_2=(1,0)$, and we calculate $\bar{o}_1=\bar{o}_4=(-0.8,0)$ and $\bar{o}_2=\bar{o}_3=(0.8,0)$; (3) we select $\alpha=1.25$, and so $\tilde{o}_1=\tilde{o}_4=(-1,0)$ and $\tilde{o}_2=\tilde{o}_3=(1,0)$; (4) the system is IBC w.r.t. $X=\text{co}\{v_1,\cdots,v_4,\tilde{o}_1,\tilde{o}_2\}$ shown in Figure 1. ◁

IV. Applications to Robotics

We show how Algorithm 3.1 can be useful for constructing safe speed profiles for different robotic systems.

A. *Fully-Actuated Robots*

Consider a fully-actuated robot with N links, modeled by:

$$D(q)\ddot{q}+C(q,\dot{q})\dot{q}+g(q)=B(q)\tau, \tag{5}$$

where $q=(q_1,\cdots,q_N)$ is the vector of generalized coordinates[3], $\dot{q}=(\dot{q}_1,\cdots,\dot{q}_N)$ is the vector of velocities, τ is the vector of generalized applied forces[4], and D is a positive definite matrix. For fully-actuated robots, it is well-known that $B \in \mathbb{R}^{N\times N}$ is full-rank, and so one can use the feedback law

$$\tau = B^{-1}(q)(C(q,\dot{q})\dot{q}+g(q)+D(q)u) \tag{6}$$

to convert (5) into the equivalent controllable linear system

$$\ddot{q}=u, \tag{7}$$

which is a set of decoupled double integrators, $\ddot{q}_i=u_i$, in the different generalized coordinates. Suppose that we have the position constraints $q_i \in [q_{i,min}, q_{i,max}]$, the velocity limits of the robot $\dot{q}_i \in [\dot{q}_{i,min}, \dot{q}_{i,max}]$, where $0 \in (\dot{q}_{i,min}, \dot{q}_{i,max})$, and the actuator limits $\tau_i \in [\tau_{i,min}, \tau_{i,max}]$, where $0 \in (\tau_{i,min}, \tau_{i,max})$. Assume that the position space is free of kinematic singularities, and that w.l.o.g. $0 \in (q_{i,min}, q_{i,max})$ for each i. Operating the robot within the maximum velocity

[2]One can verify using Theorem 2.1 that the system is not IBC w.r.t. P.

[3]The element q_i represents the angle of link i if joint i is revolute (we assume $q_i \in (-\pi,\pi]$), or it is the displacement if joint i is prismatic.

[4]That is forces and/or torques.

limits does not ensure that the robot remains within the required position limits, and consequently it does not ensure safety of operation such as collision avoidance. Instead, it is required to define a safe speed profile for the robot. That is, for each value of q_i within the position limits, we define a corresponding range of safe velocities, resulting in an overall safe region in the position-velocity state space.

We assume that for the given position-speed limits, the feedback linearization (6) can be carried out within the actuator limits of the robot, provided that for each i, u_i is selected within $[u_{i,min}, u_{i,max}]$, where $0 \in (u_{i,min}, u_{i,max})$. Hence, our task is reduced to finding for (7) a safe controllable region, within the given position-speed ranges, while taking into account the limits on u_i. It is straightforward to verify that for the controllable system (7), $\mathcal{O} + \mathcal{B} = \mathbb{R}^{2N}$, and so Algorithm 3.1 can be used. Since (7) is a set of decoupled double integrators, one can apply Algorithm 3.1 for each subsystem $\ddot{q}_i = u_i$ to find a safe speed profile for each generalized coordinate q_i (similar problem to Example 3.1).

As discussed after Theorem 3.4, although Algorithm 3.1 does not directly take the actuator limits into consideration in calculating the IBC polytope X, one can always scale X to find another IBC polytope λX, in which all the mutual accessibility problems are achieved using control inputs within the actuator limits. For the double integrator example ($\dot{x}_1 = x_2$, $\dot{x}_2 = u$, $u \in [u_{min}, u_{max}]$, where $0 \in (u_{min}, u_{max})$), this can be simply done as follows. One should first verify after constructing X using Algorithm 3.1 that at each vertex of X not in $\mathcal{O} = \{x \in \mathbb{R}^2 \ : \ x_2 = 0\}$, both the strict invariance and the strict backward invariance conditions are achieved using inputs $u \in [u_{min}, u_{max}]$ [5]. If the verification result is positive, then in spite of the actuator limits, it can be shown the system is IBC w.r.t. X using inputs u satisfying $u \in [u_{min}, u_{max}]$. Instead, with the aid of the fact that here $\mathcal{B} = \text{span}\{(0, 1)\}$, it can be shown that one can always scale the x_2-component of the vertices of X (scale the velocity profile) to end up with a new IBC polytope X' for which the mutual accessibility problems are achieved using inputs u within $[u_{min}, u_{max}]$.

To make our discussion more concrete, consider a double integrator with position limits $-1 \leq x_1 \leq 1$, velocity limits $-1 \leq x_2 \leq 1$, and actuator limits $-9 \leq u \leq 9$. It is required to find a safe controllable position-speed region within the given limits. Using Algorithm 3.1, we constructed in Example 3.1 the IBC polytope X shown in Figure 1. One can easily verify that under $-9 \leq u \leq 9$, both the strict invariance and the strict backward invariance conditions are solvable at the vertices outside $\mathcal{O}$, and so the system is IBC w.r.t. X using control inputs $-9 \leq u \leq 9$. Now suppose that we have tighter actuator limits $-4.5 \leq u \leq 4.5$. For this case, it can be easily verified that the invariance conditions are not solvable at the vertex $v_3 = (0.8, 1) \notin \mathcal{O}$, and so we need to scale the set X, or as discussed above, scale the velocity-component of the vertices not in $\mathcal{O}$. For a scaling

[5]Since the polytope X is known from Algorithm 3.1, this verification can be carried out by solving LP problems.

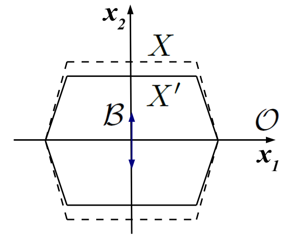

Fig. 2. The constructed IBC polytope X' under $-4.5 \leq u \leq 4.5$.

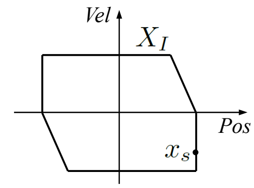

Fig. 3. A safe speed profile obtained by intuition or using the controlled invariance property.

factor $\lambda = 0.9$ of the velocity components, one can verify that for the new polytope X' shown in Figure 2, both the strict invariance and the strict backward invariance conditions are solvable at the vertices of X' not in $\mathcal{O}$. Hence, X' satisfies the IBC property under $-4.5 \leq u \leq 4.5$.

We now show the advantages of the proposed safe speed profiles in Figures 1, 2 compared to the ones obtained by intuition or using the controlled invariance property (see, for instance, the polytope X_I in Figure 3). First, our proposed method provides a systematic procedure for obtaining the vertices of the safe polytope, an advantage compared to the intuitive method, especially for complex systems. Second, our constructed polytopes satisfy the IBC property, and so there is no loss of generality (in terms of controllability) in restricting the robot to operate in the proposed safe regions. On the other hand, the regions found by intuition or through the controlled invariance property are not necessarily IBC. Third, since any state in our proposed safe position-speed regions is reachable from any other state in the region within the region itself, then in planning a reference trajectory for the robot, it would be useful to select the states of the reference trajectory inside the proposed regions, which ensures that they can be reached within the safety constraints and the actuator limits. On the other hand, one can verify that the state $x_s \in X_I$, shown in Figure 3, is not reachable from $0 \in X_I^\circ$ within X_I, i.e., it is not reachable from other states in the safe region within the region itself. One can see from Figures 1, 2 that our proposed algorithm automatically excludes these non-reachable parts of the safe region.

B. Ground Robots

We consider ground robots, modeled by:

$$\begin{aligned} \dot{x}_1 &= x_4 cos(x_3) \\ \dot{x}_2 &= x_4 sin(x_3) \\ \dot{x}_3 &= u_2 \\ \dot{x}_4 &= u_1, \end{aligned} \tag{8}$$

where (x_1, x_2) is the position of the robot in a world frame, x_3 is its orientation w.r.t. the x_1-axis, x_4 is the linear driving velocity, u_1 is the linear driving acceleration input, and u_2 is the steering velocity input. Notice that (8) differs from the kinematic model of unicycles, in which it is assumed that one can directly control the linear driving velocity. While under the kinematic model we can ensure safety of the ground robots since we can decelerate the robot to zero velocity immediately, this is not the case for the more practical model (8). Imagine a scenario in which the robot is initiated at a high linear velocity x_4 in the direction of the edges of a given region. It may happen that with the limits on u_1, we cannot decelerate the robot fast enough to avoid collision. Hence, we study the construction of safe speed profiles for (8). We assume that for low linear velocities, $|x_4| \leq x_{4,min}$, we can safely connect any two states of (8) in the given position-velocity limits, and so the problem would be only in operating the robot at high linear velocities.

The system (8) can be feedback linearized as follows (Chapter 5 of [20]). By defining the outputs $y_1 = x_1$, $y_2 = x_2$, and using the feedback linearization law:

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} cos(x_3) & -x_4 \ sin(x_3) \\ sin(x_3) & x_4 \ cos(x_3) \end{bmatrix}^{-1} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \quad (9)$$

we get $\ddot{y}_1 = \ddot{x}_1 = v_1$ and $\ddot{y}_2 = \ddot{x}_2 = v_2$, which are two decoupled double integrators representing the dynamics in the two Cartesian directions. The matrix in (9) is invertible at any state except those having $x_4 = 0$. Thus, for very low linear velocities, one should not use (9) to avoid the singularity problem. Similar to the previous subsection, given position and velocity limits in the two Cartesian directions as well as limits on the acceleration inputs $v_1, \ v_2$ in these directions, one can exploit Algorithm 3.1 to construct an IBC region for the linearized system. This region represents safe speed profiles for the robot in the two Cartesian directions. Now to connect any two states $x_0, \ x_f$ within the obtained safe region, one can start by finding a connecting trajectory $x(t), \ t \in [0, t_f]$, for the linearized model. Then, one can depend on the equivalence between the linearized model and (8) as long as x_4 does not drop to a low value. For the parts of the trajectory $x(t)$ with low x_4, one should avoid using (9), and directly control the nonlinear model (8) to connect the two states of the trajectory having low linear velocities, which can always be done safely by assumption as stated at the end of first paragraph in this subsection. Details are omitted from this conference paper for brevity.

V. CONCLUSIONS

We studied the problem of constructing IBC regions for affine systems, which are safe regions within which we can fully control the given affine system using uniformly bounded inputs. After formulating the problem, we discussed the difficulties that are faced if one tries to directly exploit the existing results for checking IBC on given polytopes. Instead, we provided a computationally efficient algorithm for constructing IBC regions, and proved its soundness. As sample case studies, we showed how our proposed algorithm can be useful for constructing safe speed profiles for different classes of robotic systems.

REFERENCES

[1] A. Aswani, H. Gonzalez, S. S. Sastry, C. Tomlin. Provably safe and robust learning-based model predictive control. *Automatica*. Vol. 49, pp. 1216-1226, 2013.

[2] F. Blanchini. Set invariance in control. *Automatica*. Vol. 35, pp. 1747-1767, 1999.

[3] M. E. Broucke. Reach control on simplices by continuous state feedback. *SIAM Journal on Control and Optimization*. Vol. 48(5), pp. 3482-3500, 2010.

[4] P. E. Caines, Y. J. Wei. The hierarchical lattices of a finite machine. *Systems and Control Letters*. Vol. 25(4), pp. 257-263, 1995.

[5] P. E. Caines, Y. J. Wei. Hierarchical hybrid control systems: a lattice theoretic formulation. *IEEE Transactions on Automatic Control*. Vol. 43(4), pp. 501-508, 1998.

[6] A. F. Filippov. *Differential Equations with Discontinuous Right Hand Sides.* Kluwer Academic Publishers, 1988.

[7] L. C. G. J. M. Habets, J. H. van Schuppen. A control problem for affine dynamical systems on a full-dimensional polytope. *Automatica*. Vol. 40(1), pp. 21-35, 2004.

[8] W. P. M. H. Heemels, M. K. Camlibel. Controllability of linear systems with input and state constraints. *IEEE Conference on Decision and Control*, New Orleans, 2007, pp. 536-541.

[9] M. K. Helwa. In-block controllability of controlled switched linear systems on polytopes. *IFAC-Papers OnLine, Elsevier*. Vol. 48 (27), pp. 7-12, 2015.

[10] M. K. Helwa, M. E. Broucke. Monotonic reach control on polytopes. *IEEE Transactions on Automatic Control*. Vol. 58(10), pp. 2704-2709, 2013.

[11] M. K. Helwa, M. E. Broucke. Flow functions, control flow functions, and the reach control problem. *Automatica*. Vol. 55, pp. 108-115, 2015.

[12] M. K. Helwa, P. E. Caines. In-block controllability of affine systems on polytopes. *IEEE Conference on Decision and Control,* Los Angeles, 2014, pp. 3936-3942.

[13] M. K. Helwa, P. E. Caines. Relaxed in-block controllability of affine systems on polytopes. *IEEE Conference on Decision and Control*, Los Angeles, 2014, pp. 3943-3949.

[14] M. K. Helwa, P. E. Caines. Hierarchical control of piecewise affine hybrid systems. *IEEE Conference on Decision and Control,* Los Angeles, 2014, pp. 3950-3956.

[15] M. K. Helwa, P. E. Caines. Epsilon controllability of nonlinear systems on polytopes. *IEEE Conference on Decision and Control,* Osaka, Japan, 2015, pp. 252-257.

[16] M. K. Helwa, P. E. Caines. On the construction of in-block controllable covers of nonlinear systems on polytopes. *IEEE Conference on Decision and Control,* Osaka, Japan, 2015, pp. 276-281.

[17] M. K. Helwa, P. E. Caines. In-block controllability of affine systems on polytopes. *IEEE Transactions on Automatic Control*. Accepted, 2016.

[18] D. Henrion, J. Löfberg, M. Kocvara, M. Stingl. Solving polynomial static output feedback problems with PENBMI. *IEEE Conference on Decision and Control, and the European Control Conference,* Seville, 2005, pp. 7581-7586.

[19] P. Hubbard, P. E. Caines. Dynamical consistency in hierarchical supervisory control. *IEEE Transactions on Automatic Control*. Vol. 47(1), pp. 37-52, 2002.

[20] A. Isidori. *Nonlinear Control Systems*. Springer-Verlag, 1995.

[21] K. Kant, S. W. Zucker. Toward efficient trajectory planning: the path-velocity decomposition. *The International Journal of Robotics Research*. Vol. 5(3), pp. 72-89, 1986.

[22] C. J. Ostafew, A. P. Schoellig, T. D. Barfoot, J. Collier. Speed daemon: experience-based mobile robot speed scheduling. *The Canadian Conference on Computer and Robot Vision*, Montreal, 2014, pp. 56-62.

[23] M. Prado, A. Simón, E. Carabias, A. Perez, F. Ezquerro. Optimal velocity planning of wheeled mobile robots on specific paths in static and dynamic environments. *Journal of Robotic Systems*. Vol. 20(12), pp. 737-754, 2003.

[24] O. Purwin, R. D'Andrea. Trajectory generation and control for four wheeled omnidirectional vehicles. *Robotics and Autonomous Systems*. Vol. 54(1), pp. 13-22, 2006.

[25] J. B. Rawlings, D. Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, LLC, 2009.

[26] O. Toker, H. Özbay. On the NP-hardness of solving bilinear matrix inequalities and simultaneous stabilization with static output feedback. *The American Control Conference*, Seattle, 1995, pp. 2525-2526.