Zero-shot Transferable and Persistently Feasible Safe Control for High Dimensional Systems by Consistent Abstraction

Tianhao Wei[1], Shucheng Kang[2], Ruixuan Liu[1], and Changliu Liu[1]

***Abstract*— Safety is critical in robotic tasks. Energy function based methods have been introduced to address the problem. To ensure safety in the presence of control limits, we need to design an energy function that results in persistently feasible safe control at all system states. However, designing such an energy function for high-dimensional nonlinear systems remains challenging. Considering the fact that there are redundant dynamics in high dimensional systems with respect to the safety specifications, this paper proposes a novel approach called abstract safe control. We propose a system abstraction method that enables the design of energy functions on a low-dimensional model. Then we can synthesize the energy function with respect to the low-dimensional model to ensure persistent feasibility. The resulting safe controller can be directly transferred to other systems with the same abstraction, e.g., when a robot arm holds different tools. The proposed approach is demonstrated on a 7-DoF robot arm (14 states) both in simulation and real-world. Our method always finds feasible control and achieves zero safety violations in 500 trials on 5 different systems.**

I. Introduction

Energy function-based methods have been extensively studied to ensure control-level safety for robotic systems in various applications, such as industrial robots in manufacturing or autonomous vehicles in transportation [1]. Typical approaches include the control barrier function method and the safe set algorithm (with safety index). These techniques aim to map dangerous states to high energy and safe states to low energy. Safety is guaranteed if a realizable control always exists that dissipates the energy whenever the state is in danger, which is known as *persistent feasibility*. Persistent feasibility can be guaranteed by offline energy function synthesis (known as the barrier function or the safety index) [2]. Formal guarantees can be provided for general nonlinear systems with up to seven dimensions in states [3].

However, existing energy function-based methods face challenges in ensuring persistent feasibility for general high-dimensional applications. For example, rule-based methods [4] only apply for specific types of system dynamics; evolutionary optimization-based synthesis does not scale well for high dimensions [2] due to the curse of dimensionality; adversarial optimization lacks formal guarantees [5]; and Sum-of-Square-based methods are restricted to certain polynomial specifications [6], [7]. Thus, methods to formally ensure safety for general high-dimensional applications are needed.

*This material is based upon work supported by the National Science Foundation under Grant No. 2144489.

[1]These authors are with Robotics Institute, Carnegie Mellon University, `twei2, ruixuanl, cliu6@andrew.cmu.edu`

[2]Shucheng Kang is with the Department of Electrical Engineering, Tsinghua University. Work done during an internship at Carnegie Mellon. `ksc19@mails.tsinghua.edu.cn`

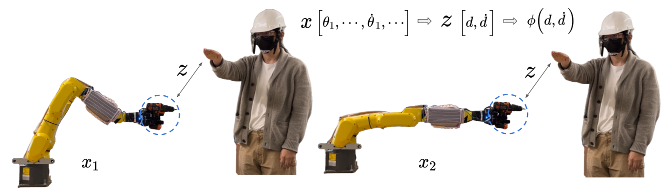


Fig. 1: An illustration of abstract safe control to prevent collision between a drill and a human. The conventional safe control approach is to model the full kinematic chain of the robot system with state x. Our proposed approach considers abstract states $z = [d, \dot{d}]$ (relative distance and velocity) and a scalar M that considers constraints imposed by the full kinematic chain (whose values are different for x_1 and x_2).

Our observation, as illustrated in fig. 1, is that high-dimensional system models are typically redundant with respect to the safety specification. For instance, if we only need to ensure collision avoidance between a drill and a human, there is no need to verify the feasibility of safe control for all robot states. Similarly, when addressing collision avoidance for a legged robot, the focus is primarily on its center of mass rather than each leg's states [8]. Hence, a simplified model of the robot can be used to replace the full dynamic model for safe control to reduce dimensionality.

A common approach to simplify dynamics is by employing a two-layer hierarchical architecture through system abstraction [9]. The high-level system uses abstracted dynamics, while the low-level system uses the original concrete dynamics. However, existing system abstraction methods [10], [11] can not be used for ensuring persistent feasibility. Because they usually assume the abstract model has uniform control constraints, which may lead to an unrealizable abstract control that breaks the feasibility. For instance, in fig. 1, although both situations correspond to the same abstract state z, the feasible abstract control (Cartesian acceleration of the drill) in these two situations is different. In particular, at x_2, the horizontal acceleration of the drill is unrealizable due to singularity. To enable safe control through system abstraction, the notion of *abstraction consistency* is essential to ensure that the high-level objective is always realizable by the concrete controller. Consistent abstraction of controllability [9] and local accessibility [12] have been studied. However, it remains unclear on how to design a consistent abstraction for persistent feasibility, since the constraint is state-dependent, and control limits must be considered.

To address these challenges, we propose a consistent abstraction for persistent feasibility, which allows us to design and verify the energy function (in the following discussion, we call it as the safety index) on a low-dimensional abstracted system. We specifically augment the abstract state with a scalar to account for different control constraints imposed by the concrete system for same abstract states. For example, we extend z in fig. 1 to $\hat{z} = [d, \dot{d}, M]$, where $M = \max |\ddot{d}|$. By mapping x_1 and x_2 to different $\hat{z}$, we can ensure that $\ddot{d}$ is always realizable by choosing it from the range $[-M, M]$. We will present a general method for designing this extended abstraction. Then we prove that a persistent feasible safety index synthesized on the extended abstraction guarantees persistent feasibility in the concrete system because all abstract controls are realizable. Lastly, we discuss how to design a persistent feasible safety index on the extended abstraction.

The abstraction not only reduces dimensionality but also enhances the transferability of the synthesized safety index. The safety index synthesized on such an extended abstraction can be directly applied to other concrete systems that have the same safety specifications (which implies same abstraction), as long as certain criteria is met. This transferability is especially useful for systems with time varying structures, such as a robot arm with changing end effectors. Once the safety index is feasible for a robot arm on the extended abstraction, it can be directly applied to the robot arm with different end effectors, to be shown in section V.

II. Formulation

A. Safety index synthesis

Consider a control affine system

$$(\Sigma_1) \quad \dot{x} = f(x) + g(x)u \tag{1}$$

where $x \in X \subseteq \mathbb{R}^{n_x}$, $u \in U \subseteq \mathbb{R}^{n_u}$. We assume U is a polytope, which is a common case in practice.

Assumption 1 (Polytope U)**.** The control limits U of the concrete system is a polytope: $U = \{u \mid Au < b\}$.

A user-defined safety specification $\phi_0(x) : \mathbb{R}^n \to \mathbb{R}$ is a continuous function and implicitly defines a connected and closed set $\mathcal{X}_s := \{\phi_0\}_{<0}$ called the safe set. We are interested in keeping the state in a subset of the user-defined safe set: $S \subseteq \mathcal{X}_s$. The problem can be expressed as a forward invariance problem:

$$x(t_0) \in S \implies \forall t > t_0, x(t) \in S. \tag{2}$$

In energy function-based methods, S is defined by a designable safety index ϕ: $S := \{\phi\}_{<0}$. The forward invariance can be guaranteed if the safe control constraint: $\dot{\phi}(x,u) < \gamma(x)$ is persistently feasible, where $\gamma(x)$ depends on the method. This paper considers the safe control constraint used in SSA [4], corresponding to the following persistent feasibility condition:

Definition 1 (Persistent feasibility)**.** A safety index ϕ is persistently feasible if $\forall x$ that $\phi(x) = 0$, there always exists $u \in U$ such that $\dot{\phi}(x,u) < 0$.

However, designing a persistently feasible safety index to ensure forward invariance is not easy, especially for high-dimensional applications. Suppose ϕ is parameterized by θ, the problem can be formulated as

$$\min_\theta |B_\theta^*| := \min_\theta \left| \{x \mid \phi_\theta(x) = 0, \inf_u \dot{\phi}_\theta(x,u) \geq 0\} \right|, \tag{3}$$

where B_θ^* denotes the set of states on the boundary of ϕ_θ that have no feasible safe control. The goal is to optimize θ such that $|B_\theta^*| = 0$. The task is difficult because computing $|B_\theta^*|$ is generally intractable for high dimensional systems.

B. System abstraction

To design a persistent feasible safety index for high-dimensional applications. We observe that, often, not all states are needed to check the satisfaction or feasibility of a safety specification as in eq. (3). For instance, when considering collision avoidance of a tool held by a robot arm (fig. 1), only the relative distance and velocity from the tool to the obstacle (2 dimensions) are required, instead of all 14 dimensions of the robot arm.

System Σ_1 is called the *concrete system*. Suppose we want to design the safety index on a space Z which is defined by a smooth, surjective map $z = \Phi(x)$, where $\Phi : \mathbb{R}^{n_x} \to \mathbb{R}^{n_z}$, $n_z \leq n_x$. Then we can define a system

$$(\Sigma_2) \quad \dot{z} = f_z(z) + g_z(z)v, \tag{4}$$

where $z \in Z \subseteq \mathbb{R}^{n_z}$, $v \in V \subseteq \mathbb{R}^{n_v}$.

Assumption 2. We assume g_z is of full column rank. Otherwise, the dimension of abstract safe control can be reduced. Therefore, $g_z(z)^{-1}$ always exists.

Definition 2 (Φ-related)**.** A system Σ_2 is Φ-related to a system Σ_1 if for every trajectory $x(t)$, $z(t) = \Phi(x(t))$ is a trajectory of Σ_2.

Σ_2 is an *abstraction* of Σ_1 if it is Φ-related to Σ_1 [12]. [9] proves that given a control system Σ and any smooth map Φ, there always exists a control system which is Φ-related to Σ. [12] provides a method to construct the smallest control system Σ_2 on Z that is Φ-related to Σ_1. If $f_z(z)$ and $g_z(z)$ are unknown, they can be constructed with this method.

However, Φ-relatedness is insufficient for designing the safety index on the abstraction. Because an abstract control at z may not be implementable at all states $x = \Phi^{-1}(z)$ by the concrete system. For example, as in fig. 1, a horizontal acceleration is implementable at x_1 but not at x_2. To enable designing safety index on the abstraction, we define consistent abstraction of constraint feasibility as follows:

Definition 3 (Feasibility consistent abstraction)**.** Let Σ_1 and Σ_2 be two control systems and $\Phi : X \to Z$ be a smooth map. Given a safety index ϕ defined on X. Σ_2 is a feasibility consistent abstraction of Σ_1 iff there exists a safety index ϕ_z defined on Z such that the following conditions are satisfied:

1) $\phi_z(z) = \phi_z(\Phi(x)) = \phi(x)$; 2) $\forall z$ *s.t.* $\phi_z(z) = 0$, if $\exists v \in V$, *s.t.* $\dot{\phi}_z(z, v) < 0$, then $\exists u \in U$, *s.t.* $\dot{\phi}(x, u) < 0$, $\forall x$, *s.t.* $\Phi(x) = z$.

Consistent abstraction w.r.t many properties has been studied when there is no control limits[12], [9]. That is, when $U = \mathbb{R}^{n_u}$ and $V = \mathbb{R}^{n_v}$. But in reality, U is usually a bounded set. How to design the abstraction under control limits is still unknown. Besides, the safety constraint $\dot{\phi}(x, u) < 0$ is state-dependent which introduces another difficulty.

In the following, we first show how to construct feasibility-consistent abstractions under control limits, and characterize when persistent feasibility on the concrete system can be guaranteed by ensuring persistent feasibility on the abstracted system, that is, when condition 2) in definition 3 can be satisfied. Then we show how to ensure persistent feasibility on the abstracted system by designing a safety index.

III. Consistent Abstraction Theory

A. *Consistent abstraction under control limits*

In order to construct a consistent abstraction under control limits, we first show how to choose Φ, the corresponding abstract space Z and ϕ_z. Then we pose a condition on the abstract control limits to make the abstraction consistent.

Suppose ϕ_0 is composed of some inner functions. That is, $\exists \varphi_1(x), \cdots, \varphi_k(x)$, *s.t.* $\phi_0(x) = \phi_0(\varphi_1(x), \cdots, \varphi_k(x))$. For example, $\varphi_i(x)$ can be the distance to obstacles, the center of mass, etc. Inspired by [4], we can define Φ and correspondingly Z by

$$\Phi = \varphi_1 \times \dot{\varphi}_1 \times \cdots \times \varphi_1^{(n_1)} \times \cdots \times \varphi_k \times \cdots \times \varphi_k^{(n_k)} \quad (5)$$

where the relative degree in the sense of Lie derivative from $\varphi_i^{(n_i)}$ to the concrete control u is one. This choice of Φ ensures the appearance of u in $\dot{z}$, which is a necessary condition of persistent feasibility. z_i are all features of x. Therefore, we can restrict the safety index $\phi(x)$ to be composite functions of Φ and let $\phi_z(z) := \phi(z)$. Then condition 1) in definition 3 is satisfied.

Definition 4 (Composition)**.** $\phi(x)$ is a composite function of Φ, that is, $\phi(x) = \phi(\Phi(x)) = \phi(z)$.

For example, as shown in fig. 1, we can define $\varphi(x) := d$, then $\phi_0(x) := 1 - d = 1 - \varphi(x)$. We can let $z = [\varphi(x), \dot{\varphi}(x)] = [d, \dot{d}]$ and $\phi(z) = \phi(d, \dot{d})$.

Next, we present the condition of consistent abstraction. We start by proving that the abstract control v is an affine transformation of the concrete control u.

Lemma 3 (Affine transformation of control)**.** Consider a control affine concrete system and a smooth map Φ. The abstract control can be represented as an affine transformation of the concrete control, specifically, we can let $v = C(x)u + d(x)$, where $C(x) = g_z(\Phi(x))^{-1}\nabla\Phi(x)g(x)$, and $d(x) = g_z(\Phi(x))^{-1}[\nabla\Phi(x)f(x) - f_z(\Phi(x))]$. The proof is in appendix A (on arxiv).

Definition 5 (Implementable abstract control limits)**.** Given a polytope concrete control limits U and a state x, based on lemma 3, the implementable abstract control limits at x is also a polytope set, defined as $\Phi_U(x) := \{C(x)u + d(x) \mid \forall u \in U\}$.

Theorem 4 (Feasibility consistency condition)**.** Let Σ_1 and Σ_2 be two control systems and $\Phi : X \to Z$ be a surjective smooth map. Suppose the abstract control limits V is a state-dependent set, then condition 2) in definition 3 can be satisfied if

$$\forall z,\ V(z) \subseteq \cap_{x\in\Phi^{-1}(z)}\Phi_U(x) \quad (6)$$

Proof. For arbitrary $x \in \{\phi(x)\}_{=0}$, we can let $z = \Phi(x)$ and have $\phi(z) = 0$. If there exists a $v \in V(z)$ such that $\dot{\phi}(z, v) < 0$. From eq. (6), we know this $v \in V(z) \subseteq \Phi_U(x)$. Therefore, based on definition 5, $\exists u \in U$ such that $v = C(x)u + d(x)$. And this u satisfies that $\dot{\phi}(x, u) = \frac{\partial\phi}{\partial x}\dot{x} = \frac{\partial\phi}{\partial z}\frac{\partial z}{\partial x}\dot{x} = \frac{\partial\phi}{\partial z}\dot{z} = \dot{\phi}(z, v) < 0$. $\square$

Theorem 4 states that the feasibility can be propagated from Σ_2 to Σ_1 if all abstract control $v \in V(z)$ at a given z is always implementable on the concrete system at arbitrary $x \in \Phi^{-1}(z)$. However, $V(z)$ is difficult to construct because $\Phi^{-1}(z)$ is difficult to compute. For example, $\Phi^{-1}(z)$ may correspond to all possible poses of a robot arm given an end-effector status z. Besides, $z = \Phi(x)$ may aggregate x with vastly different abstract control limits $\Phi_U(x)$ into the same class, making $V(z)$ very tight and even empty. As shown in fig. 1, if we design control using $V(z) = \Phi_U(x_1) \cap \Phi_U(x_2)$, we will lose horizontal acceleration for both cases. Therefore, we need a better method that is easy to construct and improves conservativeness.

B. *Non-conservative abstraction under control limits*

To address this issue, we propose an extended abstraction that only aggregates x with similar abstract control limits $\Phi_U(x)$. In the following, we first define a one-dimensional under-approximation of $\Phi_U(x)$, then define such an extended abstraction.

Definition 6 (state-dependent radius of control constraints)**.** The radius of the largest zero-centered inner Lp-norm ball of $\Phi_U(x)$ is defined as:

$$M(x) = \max_r r \quad \textit{s.t.}\ B_p(r) \subseteq \Phi_U(x), \quad (7)$$

where $B_p(r) = \{v \mid \|v\|_p \leq r\}$.

Definition 7 (Extended abstraction)**.** With $M(x)$, we define an extended abstraction for Σ_1 as $\hat{\Phi}(x) := \Phi(x) \times M(x)$, and $\hat{z} := z \oplus M$. The abstract control is defined as $\hat{v} = v \times m$. The abstract control limit is defined by $\hat{V}(\hat{z}) := B_p(M) \times \mathbb{R}$. The corresponding extended abstraction space is $\hat{Z} := Z \times \mathbb{R}$, which has only one more dimension than Z.

In this way, we keep the extended abstraction in low dimension and gain flexibility in designing abstract control under different control limits.

One prerequisite of this under-approximation is that $\Phi_U(x)$ must contain 0. Because $M(x) = 0$ when $0 \notin \Phi_U(x)$.

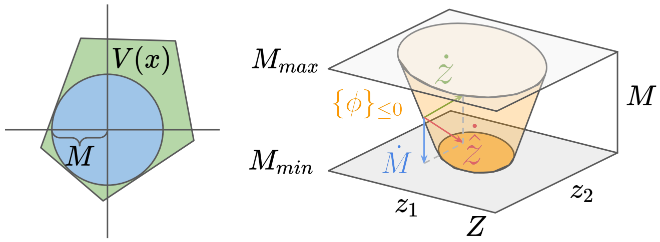


Fig. 2: Left: Abstract control limit $V(x)$ and its under-approximation M. Right: Extended abstraction and extended abstract safe set. The abstract space Z is extended with the scalar M, representing the maximal inner Lp-norm ball of $\Phi_U(x)$. In this way, the abstract safe set is lifted from Z to $\hat{Z} = Z \times \mathbb{R}$. Designing the safety index ϕ on $\hat{Z}$ enables different safe control under different control limits. Persistent feasibility requires that there always exists a control that leads to a system flow $\dot{\hat{z}}$ toward the interior of the invariance set.

In this case, $\hat{V}(\hat{z})$ is an empty set which makes Σ_2 ill-defined. Therefore, we provide sufficient conditions of $0 \in \Phi_U(x)$.

Assumption 5 ($0 \in \Phi_U(x)$)**.** We assume

$$\forall x, \exists u, \ \textit{s.t.}\ C(x)u + d(x) = 0. \tag{8}$$

To understand when the assumption holds, we present the following three case studies:

Case 1. If $0 \in U$ and systems are driftless [13], that is $f(x) = 0$ and $f_z(z) = 0$. Then $0 \in \Phi_U(x)$ always holds.

Case 2. When v is a scalar, a sufficient condition of $\Phi_U(x)$ contains 0 is $\min_{u\in\partial U} \|u\| > \max_{x\in X} \|d(x)\|/\|C(x)\|$.

Case 3. When v is a vector, a sufficient condition that $0 \in \Phi_U(x)$ is that $\min_{u\in\partial U} \|u\| > \max_{x\in X} \|C(x)^T w(x)\|$, where $w(x)$ is a solution of $C(x)C(x)^T w(x) + d(x) = 0$.

The extended abstraction enables the design of abstract controllers under varying control limits. Without the extension, an abstract controller will give us the same control under different abstract control limits. But after the extension, we can define the safety index on $\hat{Z}$, which is abstract control limits aware. For example, as in fig. 1, if the only considering $d, \dot{d}$, an abstract controller will suggest the same control for x_1 and x_2, but after the extension, it will suggest different actions for x_1 and x_2.

IV. Abstract Safe Control

The consistent abstraction theory shows that a persistent feasible safety index defined on the extended abstraction guarantees persistent feasibility for the concrete system. Next, we show how to design a persistent feasible safety index on the extended abstract space $\hat{Z}$.

A. Persistent feasibility on extended abstraction

As shown in fig. 2, a safety index defined on Z corresponds to a disk invariance set on the Z plane. But a safety index defined on $\hat{Z} = Z \times \mathbb{R}$ corresponds to a bucket shape invariance set in $\hat{Z}$. Persistent feasibility requires that we can always find a control on the boundary of the invariance set that leads to a system flow towards the inside of the invariant set. Formally, persistent feasibility requires the following inequality holds for all $x \in \{\phi\}_{=0}$:

$$\min_{\|v\|<M} \dot{\phi} + \gamma(\phi) = \frac{\partial \phi}{\partial z}\dot{z} + \frac{\partial \phi}{\partial M}\dot{M} \quad < 0 \tag{9}$$

$$= \frac{\partial \phi}{\partial z(x)}[f_z(z(x)) + g_z(z(x))v] + \frac{\partial \phi}{\partial M}\dot{M}(x) \quad < 0 \tag{10}$$

The numerical method to verify persistent feasibility for general nonlinear systems has an exponential growth time complexity $\mathcal{O}(2^n)$ [2], where n is the dimension of the state. Therefore we wish to verify the feasibility on the abstracted system and propagate the feasibility back to concrete system. However, notice that $\dot{M}(x)$ does not depend on z but on x. To guarantee the abstract control is consistent for all $x \in \Phi^{-1}(\hat{z})$. We propose the following method.

Suppose $M(x) \in [M_{min}, M_{max}] := R_v$, we instead verify the following stricter inequality: $\forall z \in \{\phi\}_{=0}, \forall M, \exists v$, such that

$$\min_{\|v\|<M} \frac{\partial \phi}{\partial z}\left[f_z(z) + g_z(z)v\right] + \left|\frac{\partial \phi}{\partial M}\dot{M}_{max}\right| < 0. \tag{11}$$

It is easy to see that eq. (11) $\implies$ eq. (10). Intuitively, eq. (11) says that we can always find a control u that corresponds to a $\dot{z}$, whose combination with the largest possible $\dot{M}$ (which is $\dot{\hat{z}}$) still points towards the interior of the invariant set. And because eq. (11) is x-independent, we can verify it on the space $\hat{Z}$, which is a much smaller space than X, therefore can be verified by numerical methods. We can derive the following lemma from eq. (11):

Lemma 6. A safety index is persistently feasible if $\dot{M}_{max}$ satisfies the following condition:

$$\dot{M}_{max} < \inf_{z,M\in Z\times R_v} -\frac{\min_{\|v\|<M} \frac{\partial \phi}{\partial z}\left[f_z(z) + g_z(z)v\right] + \gamma(\phi)}{\left|\frac{\partial \phi}{\partial M}\right|}.$$

In addition, eq. (11) can be relaxed when $M = M_{min}$. Because $\frac{\partial \phi}{\partial M}$ is expected to be always negative (the system is safer when the control limit is larger), and $\dot{M} \ge 0$ because M can not be less. Therefore $\frac{\partial \phi}{\partial M}|_{M_{min}} \dot{M}|_{M_{min}} \le 0$. Therefore, we only need to verify that $\min_{\|v\|<M} \frac{\partial \phi}{\partial z}\left[f_z(z) + g_z(z)v\right] < 0$ when $M = M_{min}$. We can estimate M_{min} and M_{max} by sampling the state space and computing M with the following lemma.

Lemma 7 (M computation)**.** Suppose $\Phi_U(x)$ is composed of linear constraints $\Phi_U(x) := \{v \mid a_i(x)v \le b_i(x), \forall i\}$, the maximal radius of the L_p-norm ball can be found by solving the following convex optimization.

$$\max_M M \quad \textit{s.t.}\ \|a_i(x)\|_{1/p} M \le b_i(x), \forall i. \tag{12}$$

The proof is in appendix B (on arxiv).

B. Safety Index Synthesis and Execution

With dynamics abstraction, we can synthesize ϕ on a low dimensional space. Therefore, we can use analytical or numerical methods [2] to compute $|B_\theta^*|$ in eq. (3). During online execution, we can find a safe control in the original space by solving the following linear inequality:

$$\dot{\phi}(x,u) = \frac{\partial \phi}{\partial z}\frac{\partial z}{\partial x}\dot{x} + \frac{\partial \phi}{\partial M}\frac{\partial M}{\partial x}\dot{x} \tag{13}$$

$$= (\frac{\partial \phi}{\partial z}\frac{\partial z}{\partial x} + \frac{\partial \phi}{\partial M}\frac{\partial M}{\partial x})[f(x) + g(x)u] < 0. \tag{14}$$

Due to the complex transformation, it can be difficult to compute $\frac{\partial M}{\partial x}$ by analytical methods. But we can compute $\frac{\partial M}{\partial x}$ numerically by perturbing x: $\frac{\partial M}{\partial x} = \lim_{\delta \to 0}[M(x+\delta) - M(x)]/\delta$. The approximation error can be made up by relaxing $\frac{\partial M}{\partial x}\dot{x}$. Because $\dot{M}$ is bounded and the persistent feasibility guarantees a safe control exists even for $\dot{M}_{max}$, therefore safety always can be guaranteed by choosing the most conservative safe control in the worst case.

With this safety constraint, we can choose a control u that is close to a given reference control u_0 with a QP objective function as in most energy function based method [1]: $\min_u \|u - u_0\|$ *s.t.* $\dot{\phi}(x,u) < 0$.

V. Experiment

Experiments are designed to show that 1. the prerequisite of the method can be easily achieved; 2. the method ensures persistent feasibility for high dimensional systems; 3. the synthesized safety index can be transferred to systems with different dynamics.

Our experiments are tested on a 7 degrees-of-freedom (DoF) Franka Panda robot arm simulation platform with a 1.5GHz AMD EPYC 7H12 64-Core Processor, and a 7 DoF FANUC LR Mate 200i real robot.

The robot's task is to reach a goal while avoiding collision with obstacles or humans. We consider a collision avoidance constraint $\phi_0 = d_{min} - d$, where $d_{min} = 0.05$ and d is the relative distance.

A. Distribution of M and $\dot{M}$

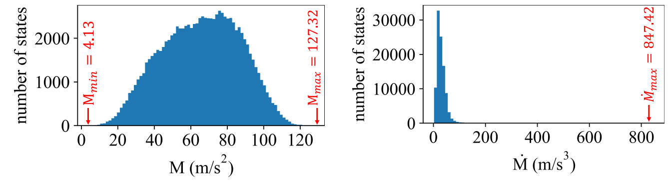
We first show the distribution of M and $\dot{M}$. The distribution in fig. 3a reveals that M is always above 0. Very few states have a small range. Therefore, the extended abstraction greatly reduced conservativeness. The distribution of $\dot{M}$ as shown in fig. 4c shows that most states have a small $\dot{M}_{max}$, which eases finding feasible safe control. It takes 14 hours to sample 100000 $\dot{M}$ and estimate $\dot{M}_{max}$.

The figures in fig. 4 show the corresponding robot arm poses for the extreme values of M and $\dot{M}$. We can see that these poses usually corresponds to singular states, such as fully extended. Therefore, users may accelerate the system property verification with their knowledge.

B. Persistent Feasibility

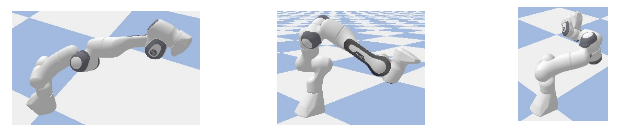
The safety index is designed as the following form:

$$\phi = \max(\phi_0, \phi^*), \text{ where } \phi^* = d_{min}^2 - d^2 - k\dot{d}/M. \tag{15}$$



(a) M distribution (b) $\dot{M}$ distribution

Fig. 3: Distribution of M and $\dot{M}$



(a) M_{min} pose (b) M_{max} pose (c) $\dot{M}_{max}$ pose

Fig. 4: Poses for extreme values of M and $\dot{M}$

The persistent feasibility of ϕ can be guaranteed by the persistent feasibility of ϕ^* as shown in [4]. Therefore, we can use system abstraction to directly determine the value of k that guarantees feasibility of ϕ^* and ϕ. Assuming $d \in [0.0, 0.8]m$ and $\dot{d} \in [-1.0, 1.0]m/s$: $\forall(d, \dot{d}, M)$, s.t. $\phi^* = 0$, there exists $\ddot{d} \in [-M, M]$ such that:

$$\dot{\phi}^* = -2d\dot{d} - k\ddot{d}/M + k\dot{d}\dot{M}/M^2 \tag{16}$$

$$\leq -2d\dot{d} - k\ddot{d}/M + k|\dot{d}|\dot{M}_{\max}/M^2 \leq 0 \tag{17}$$

We choose $\ddot{d} = -M$, then $k \geq -2d\dot{d} + k|\dot{d}|\dot{M}_{\max}/M^2$. Since $\phi^* = 0$, we have $k = M(d_{min}^2 - d^2)/\dot{d}$ and

$$k \geq -2d\dot{d} + |\dot{d}|/\left(d_{min}^2 - d^2\right) \cdot \frac{\dot{M}_{\max}}{\dot{d}M} \tag{18}$$

$$\geq 2 \mid d\dot{d} \mid_{\max} + \mid d_{min}^2 - d^2 \mid_{\max} \cdot \frac{\dot{M}_{\max}}{M_{\min}} \tag{19}$$

which implies $k \geq 133.31$. As shown in Table I, this value of k ensures no collisions in 100 randomly generated goal-obstacle pairs. In contrast, it is too complex to derive the analytical condition for the concrete system, which has hundreds of terms with nonlinear transformations such as trigonometric functions and multiplication.

and the exponential time complexity of numerical methods makes it infeasible to check for feasibility. For example, for a 7 degree-of-freedom robot arm, even with only 10 samples per dimension, it would require checking 10^{14} samples.

Case	Collision	Range of M	$\dot{M}_{max}$
5 DoF + EE 1	0	[1.67, 114.37]	1052.93
6 DoF + EE 1	0	[3.26, 121.34]	891.80
7 DoF + EE 1	0	[4.13, 127.32]	847.42
7 DoF + EE 2	0	[4.53, 120.19]	710.97
7 DoF + EE 3	0	[3.82, 131.55]	856.18

TABLE I: Collision count in 100 randomly generated scenarios and dynamics sensitivity to DoF and End Effectors.

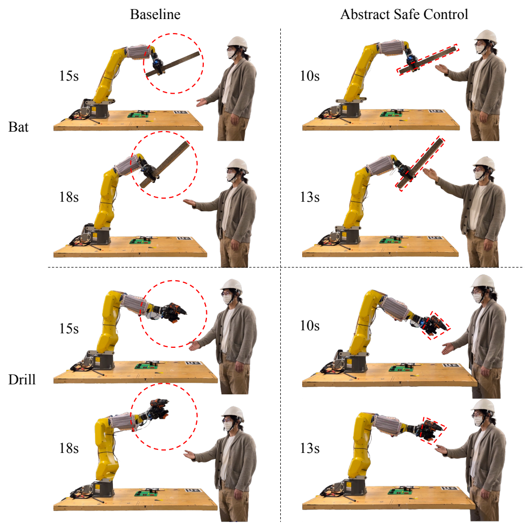


Fig. 5: Experiments on FANUC with two different tools. The baseline method ensures safety by considering a large sphere that contains all possible tools. But our method can guarantee safety by considering the nearest point from the tool to the human, which is much less conservative than the baseline.

C. Transplant to other systems

In this experiment, we change robot's degree-of-freedoms (DoF) and end-effectors (EE) to discuss when the safety index can be directly transferred to other systems. EE 1 is the baseline. In EE 2, a $0.1kg$-rod is attached to the end-effector. In EE 3, the offset of the seventh joint has a 0.05m increment in y-axis. In 6 DOF robot, the seventh joint is fixed. In the 5 DOF robot, both the sixth and seventh joints are fixed. As shown in table I, $\dot{M}_{max}$ decreases when the DoF increases, and the range of M does not change too much with the dynamics. We can conclude that if we consider a wide range of M and a large enough $\dot{M}_{max}$ during the design of the safety index, the same safety index can ensure safety for a wide range of unseen systems. This method is particularly useful for robot arms with real-time tool switching.

D. Experiment on real robots

We also test the method on a real FANUC LR Mate 200i robot with two different tools: a drill and a bat. These two tools have different shapes and different kinematics. Our method significantly reduces conservativeness when the tool is unknown. As shown in fig. 5, to ensure safety during tool use, the baseline method constructs a large sphere space that covers all possible tools. However, this method leads to conservative interaction with humans. But with our method, we can consider the nearest point from the robot to the human because the nearest points can be viewed as different end effectors that our safety index can be directly transplanted. Therefore we can provide safety guarantees for both tools while remaining non-conservative, the profiles of ϕ_0 are shown in fig. 6.

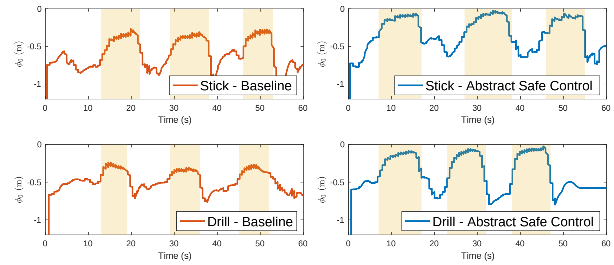


Fig. 6: ϕ_0 profiles $(-d)$. Our abstract safe control method is much less conservative, allowing a closer distance to the human while maintaining safety. The interaction sessions are shaded in yellow.

VI. Discussion

In conclusion, we propose a new method called abstract safe control that enables ensuring persistent feasibility for high dimensional systems by consistent system abstraction. The method provides a promising solution for ensuring safety in high-dimensional robotic tasks, with potential applications in various domains. In the future, we will explore how to derive the analytical upper bound of $\dot{M}_{max}$ to avoid the sampling-based estimation.

References

[1] T. Wei and C. Liu, "Safe control algorithms using energy functions: A unified framework, benchmark, and new directions," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 238–243.

[2] ——, "Safe control with neural network dynamic models," in *Learning for Dynamics and Control Conference*. PMLR, 2022, pp. 739–750.

[3] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, "Hamilton-jacobi reachability: A brief overview and recent advances," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 2242–2253.

[4] C. Liu and M. Tomizuka, "Control in a safe set: Addressing safety in human-robot interactions," in *Dynamic Systems and Control Conference*, vol. 46209. American Society of Mechanical Engineers, 2014, p. V003T42A003.

[5] S. Liu, J. Dolan, and C. Liu, "Safe control under input saturation with neural control barrier functions," in *6th Annual Conference on Robot Learning*, 2022.

[6] A. Clark, "Verification and synthesis of control barrier functions," *arXiv preprint arXiv:2104.14001*, 2021.

[7] W. Zhao, T. He, T. Wei, S. Liu, and C. Liu, "Safety index synthesis via sum-of-squares programming," *arXiv preprint arXiv:2209.09134*, 2022.

[8] W. Zhao, T. He, and C. Liu, "Model-free safe control for zero-violation reinforcement learning," in *5th Annual Conference on Robot Learning*, 2021.

[9] G. J. Pappas, G. Lafferriere, and S. Sastry, "Hierarchically consistent control systems," *IEEE transactions on automatic control*, vol. 45, no. 6, pp. 1144–1160, 2000.

[10] S. W. Smith, H. Yin, and M. Arcak, "Continuous abstraction of nonlinear systems using sum-of-squares programming," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 8093–8098.

[11] H. Yin, M. Bujarbaruah, M. Arcak, and A. Packard, "Optimization based planner–tracker design for safety guarantees," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 5194–5200.

[12] G. J. Pappas and S. Simic, "Consistent abstractions of affine control systems," *IEEE Transactions on Automatic Control*, vol. 47, no. 5, pp. 745–756, 2002.

[13] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

APPENDIX

A. Lemma 3 proof

Proof. Based on eq. (4), we have

$$\dot{z} = f_z(z) + g_z(z)v = f_z(\Phi(x)) + g_z(\Phi(x))v \tag{20}$$

And in the meantime, we have

$$\dot{z} = \frac{\partial \Phi(x)}{\partial t} = \nabla\Phi(x)\dot{x} = \nabla\Phi(x)f(x) + \nabla\Phi(x)g(x)u \tag{21}$$

Therefore, connect eq. (20) and eq. (21) we have

$$\begin{aligned} v =& g_z(\Phi(x))^{-1}[\nabla\Phi(x)f(x) - f_z(\Phi(x))] \\ &+ g_z(\Phi(x))^{-1}\nabla\Phi(x)g(x)u && \text{(22a)} \\ =& C(x)u + d(x) && \text{(22b)} \end{aligned}$$

$\square$

B. Lemma 7 proof

Proof. The radius of the maximum L_p-norm inner ball can be found by

$$\max_{r} r \quad \textit{s.t.}\ a_i(x)v \leq b_i(x), \forall \|v\|_p \leq r. \tag{23}$$

Based on Hölder's inequality, for p, q that satisfy $\frac{1}{p} + \frac{1}{q} = 1$, we have

$$|a_i(x)v| \leq \|a_i(x)\|_q \|v\|_p \tag{24}$$

Therefore

$$\max_{\|v\|_p \leq r} a_i(x)v = \max_{\|v\|_p \leq r} |a_i(x)v| \leq \max_{M} \|a_i(x)\|_q M \tag{25}$$

$\square$