

Safe and Steerable Geometric Motion Policies for Robotic Dexterous Manipulation

Albert Wu¹, Riccardo Bonalli², Thomas Lew³, C. Karen Liu¹

Abstract—Robotic dexterous manipulation requires continuously reconciling objectives and constraints defined on heterogeneous geometric spaces: a robot controlled on a \mathbb{R}^7 configuration manifold may need to track end effector poses on $SE(3)$ while satisfying obstacle avoidance margins in \mathbb{R} . We present Safe Pullback Bundle Dynamical Systems (SafePBDS), a geometrically consistent motion generation framework that computes optimal, certifiably safe configuration manifold accelerations from objectives and safety requirements defined on arbitrary task manifolds. SafePBDS builds on prior work that combines predefined task manifold dynamical systems to produce autonomous motion. Its first innovation is a pullback control barrier function construction, which converts task manifold safety conditions into linear constraints on configuration manifold accelerations. The second innovation is a task manifold action interface that allows a high-level policy to inject low dimensional residual motions; zero input recovers the autonomous behavior, while safety is preserved under arbitrary inputs. This enables high-level policies to efficiently steer exploration while leaving precise motion generation to the autonomous behavior. We validate SafePBDS in simulation and on a 23-DOF Franka Panda-Allegro Hand platform. On dexterous grasping, SafePBDS achieves a 92.5% success rate across 20 household objects and 120 trials. By leveraging the action interface, the method can exclude any one of the four fingers during grasping using a one-dimensional action, achieving 94.4% 3-finger grasp success across 3 objects and 36 trials. The efficient planning and safety guarantee of SafePBDS also enables the first model-based, fully actuated palm-down in-hand reorientation, exceeding 360° of yaw rotation in both directions under varying object weight and wrist motion. A demo video and additional details are available at tml.stanford.edu/safe-pbds.

Index Terms—Constrained motion planning, dexterous manipulation, grasping, robot safety, in-hand manipulation, multifingered hands.

I. INTRODUCTION

Robotic dexterous manipulation requires reasoning over quantities defined on different geometric spaces, including positions in Cartesian space, orientations on $SO(3)$, and joint limits in generalized coordinates. Some of these quantities correspond to soft objectives, such as null-space redundancy resolution and task prioritization, while others impose hard constraints that must never be violated, such as collision avoidance, closed-loop kinematic constraints, and force closure. Moreover, real-world perturbations and modeling errors require the system to respond quickly via feedback during execution. Successful manipulation therefore demands continuously

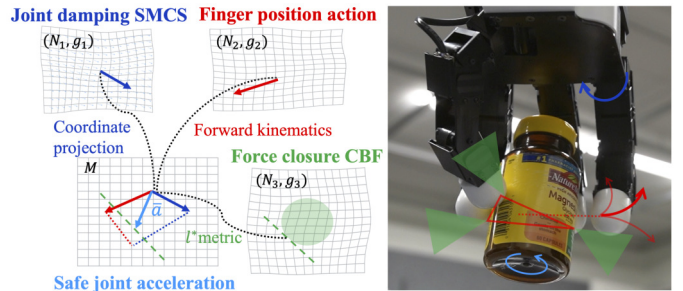


Fig. 1. Overview of SafePBDS. At each control step, tasks on heterogeneous manifolds (N_i, g_i) are pulled back to the configuration manifold M and composed into a safe acceleration \bar{a} via quadratic programs. (Left) Representative tasks include an autonomous joint-damping SMCS (blue), a high-level finger-position action (red), and a force-closure CBF safety constraint (green), each connected to M by a task map. (Right) On a 16-DOF Allegro Hand, these components combine to achieve palm-down in-hand reorientation (Section VIII-D).

resolving objectives and constraints defined on heterogeneous geometric spaces in real time.

Contemporary data driven methods attempt to sidestep these challenges by learning end-to-end policies that implicitly encode all objectives and constraints. Deep reinforcement learning (DRL) [28, 14, 8, 33] and behavior cloning [60, 10, 58] can capture globally coordinated behaviors and are often straightforward to deploy. However, the resulting policies often remain specialized to a single task or embodiment [51]. Vision-language-action models (VLAs) [13, 30, 47] have shown impressive generalization on arm-gripper manipulation, but their extension to multi-fingered hands remains limited by the complexity of dexterous motion and the difficulty of collecting multi-fingered manipulation data. Additionally, all aforementioned black-box policies can fail catastrophically outside the training distribution, motivating the need to incorporate certifiable structure for safety guarantees.

At the other end of the spectrum, trajectory optimization [32, 39, 29, 24] can explicitly reason about constraints and dynamics to produce certifiable motion plans. However, solving these optimizations is computationally expensive, prone to getting stuck in local minima, require specific engineering tricks such as fixed contact pairs, and the resulting plans are essentially one-shot. Adapting them to online perturbations requires expensive replanning at rates that are typically incompatible with high-frequency dexterous manipulation control.

A third family of approaches is the *vector field policy*, which maps each state to a desired velocity or acceleration through a globally defined vector field [34, 6]. Because these policies are defined analytically, they enable fast reactive behavior, and properties such as stability can often be certified by construction. Several frameworks compose per-task vector

¹Computer Science Department, Stanford University, Stanford, CA 94305, USA. {amhwu, karenliu}@cs.stanford.edu

²Laboratory of Signals and Systems, University of Paris-Saclay, CNRS, CentraleSupélec, France.

riccardo.bonalli@12s.centralesupelec.fr

³Toyota Research Institute, Los Altos, CA 94022, USA.

thomas.lew@tri.global

fields using Riemannian geometry [34, 9, 35, 50, 6], but existing formulations either lack geometric consistency or treat safety as a soft property [35, 50]. Moreover, because most vector field policies are inherently local, they can become trapped in local minima. A recent example, DextrAH-G [23], combines vector field policies with learning-based methods to mitigate this lack of global planning, but the resulting policies remain limited by the vector field formulation.

Among vector field policies, Pullback Bundle Dynamical Systems (PBDS) [6] stands out for achieving geometric consistency. By building on the coordinate invariant simple mechanical control system (SMCS), PBDS provides a fully geometric formulation that works on arbitrary Riemannian manifolds. However, PBDS has two important limitations. First, *safety constraints are soft*: metric-based constraints enter the optimization as additional weighted objectives and can therefore be violated when they conflict with other tasks; moreover, these objectives may become ill-defined in unsafe regions. Second, *there is no action interface*: the system provides no mechanism for a higher-level policy to inject task manifold commands, limiting its use to autonomous policies fixed at design time.

We present *SafePBDS* (Safe Pullback Bundle Dynamical Systems), a framework that addresses both limitations of PBDS while preserving the geometric and compositional properties. *SafePBDS* extends PBDS with hard safety guarantees and external controllability while retaining its geometric consistency. At each control step, a constrained quadratic program takes in real-time observations and composes desired autonomous behaviors, safety constraints, and action inputs into a single acceleration target. This is enabled by the following:

- *Pullback control barrier functions (CBFs)* that enforce task manifold safety as hard constraints on the configuration manifold acceleration by pulling back the constraints through smooth task maps. We derive formulations for two prominent higher-order CBF variants, exponential (ECBF) and backstepping (BCBF), and show their respective dependencies on task manifold geometries.
- *A task manifold action interface* that lets a higher-level policy inject inputs on selected task manifolds. Our formulation guarantees that zero input recovers the autonomous behavior and that safety is preserved under arbitrary inputs.

We also provide extensive evaluations of *SafePBDS*:

- *Simulation experiments* on an \mathbb{S}^2 double integrator and a 7-DOF robot arm validate the theoretical properties of our framework: chart invariance, task manifold metric effects, recovery from unsafe states, and the action interface.
- *Hardware experiments* on a 23-DOF arm–hand system (Franka Panda + Allegro Hand) demonstrate (a) autonomous dexterous grasping of 20 household objects at a 92.5% success rate (111/120 trials), including a 3-finger ablation at 94.4% (34/36); and (b) robust in-hand reorientation under palm-down and variable wrist orientations, achieving over 360° rotation in both directions.

The remainder of this paper is organized as follows. Section II reviews related literature. Section III introduces the

necessary theoretical background. Section IV derives the pullback ECBF and BCBF formulations. Section V presents the task manifold action interface. Sections VII and VIII present simulation and hardware experiments, respectively.

II. RELATED WORK

Given the breadth of the motion planning and robotic manipulation literature, we restrict our attention to topics directly related to this work. For the adjacent literature, we refer the reader to recent surveys on optimization-based task and motion planning [59], robot manipulation in contact [43], motion planning for manipulators in dynamic environments [21], deep reinforcement learning for robotics [44], and imitation learning for contact-rich manipulation [48].

A. Multi-task motion planning on manifolds

Reactive multi-objective control in Euclidean task manifolds is classically addressed by operational space control [15, 37, 40], which composes Cartesian objectives through null space projection. This line of work establishes key ingredients but does not provide a geometrically consistent or coordinate free framework for composing manifold valued tasks.

A major step toward manifold-valued policy composition is the introduction of *Riemannian Motion Policies* (RMPs) [34], which compose acceleration-level policies through metric-weighted pullback; RMPflow [9] later extends this construction to tree-structured task hierarchies. However, the original RMP formulation is not chart-invariant, so the resulting policy depends on the choice of local coordinates [6]. A related line of work is the *Optimization Fabrics* and *Geometric Fabrics* framework [35, 50], which provides a Finsler-geometric foundation for stable reactive policy design. Across this family of methods, however, safety-related behaviors are typically encoded in the reactive policy itself rather than enforced as hard constraints at runtime. Unresolved geometric consistency issues also leave them short of a complete solution to composing tasks defined on heterogeneous manifolds.

Pullback Bundle Dynamical Systems (PBDS) [6] resolves the geometric consistency issue by realizing simple mechanical control systems (SMCSs) [5] on task manifolds and combining them through a metric-weighted least-squares problem. In doing so, PBDS establishes a principled foundation for geometrically consistent control synthesis on manifolds. However, PBDS inherits the broader limitation above: safety-related tasks enter the least-squares objective as soft costs and may therefore be violated under competing task pressures. PBDS also remains purely autonomous, with no mechanism for a higher-level planner or learned policy to steer the system at runtime. Our work addresses both limitations.

B. Control barrier functions on manifolds

Control barrier functions (CBFs) certify forward invariance of a safe set by imposing an affine inequality on the control input at runtime [2]. The earliest formulations assume the safety function has relative degree one with respect to the input. This assumption fails for systems where position-level safety must be enforced through acceleration-level control. Two families of methods address this higher-relative-degree

setting, *Exponential CBFs* (ECBFs) [27] apply pole-placement design to the chain of Lie derivatives of the safety function, collapsing the higher-order condition into a single linear constraint on the input. A related generalization, high-order CBFs (HOCBFs) [55], relaxes the linear pole-placement structure of ECBFs to a sequence of class- \mathcal{K} comparison functions. Backstepping CBFs (BCBFs) instead build a CBF from a safe virtual controller for the lower-order subsystem and lift it to the full system [45]; for second-order systems with position constraints, this construction reduces to lifting a safe velocity field to an acceleration-level constraint.

Recent work has generalized the CBF methodology beyond Euclidean spaces to manifold-valued states. Wu and Sreenath [54] extend CBF synthesis to mechanical systems evolving on Riemannian configuration manifolds, with demonstrations on the spherical pendulum (\mathbb{S}^2) and the 3D pendulum ($\text{SO}(3)$). More recently, De Sa et al. [36] develop a general theory of geometric CBFs on control systems defined over bundles, and use it to generalize kinetic-energy CBF backstepping to SMCSs. These works establish geometric CBFs on individual *configuration* manifolds; `SafePBDS` instead defines safety on *task* manifolds so that hard constraints naturally expressed in different geometric spaces (e.g. end effector poses and joint angle limits) can be composed with the PBDS task structure.

C. Dexterous manipulation

We focus on two dexterous manipulation problems targeted in our hardware evaluation: multi-fingered *grasping* and *in-hand reorientation*. Both require coordinating objectives and constraints defined on heterogeneous spaces, including force closure and friction-cone conditions in contact space, fingertip and link clearance in Cartesian space, joint limits in the configuration manifold, and reachability in $\text{SE}(3)$.

1) *Grasping*: The theory of static multi-fingered grasping is well established [25]. We refer the reader to [4, 26, 19] for broader surveys of grasp synthesis and focus on methods that provide explicit physical-feasibility guarantees, as in our work. A central challenge in certifiable grasp synthesis is that precise local constraints must be satisfied while searching over multiple global grasp modalities (see [12] for a grasp taxonomy). This challenge has motivated optimization-based pipelines that incorporate analytical grasp-quality objectives, sometimes used to refine learned grasp predictions [22, 49, 61]. In most such methods, however, the relevant physical conditions enter as relaxed scalar penalties and therefore do not provide strict guarantees on the final grasp. Wu et al. [53] address this limitation by formulating grasp refinement as a bilevel optimization in which force closure is imposed as an exact inner constraint, yielding grasps with certified physical feasibility at the solution. Li et al. [20] further introduce the min-weight metric as a surrogate objective for force closure certification. However, both [53] and [20] certify grasp feasibility offline and do not maintain it online under execution-time perturbations. Lum et al. [23] combine reinforcement learning with a Geometric Fabric to enforce arm joint limits and collision avoidance during grasp execution, but do not certify grasp conditions such as force closure. Shaw Cortez

et al. [38] use a CBF-based safety filter to preserve grasp validity (e.g., preventing slip and singularities) during in-hand adjustments, but do not discuss how the grasp is initially achieved.

2) *In-hand reorientation*: In-hand reorientation can be broadly divided by palm orientation. In the *palm-up* setting, gravity helps retain the object, allowing brief release-and-regrasp maneuvers. In the *palm-down* setting, the object must remain securely grasped throughout the motion. Both have proven challenging, and most prior work has relied on deep reinforcement learning with sim-to-real transfer, beginning with palm-up [28, 14, 33, 57] and extending to palm-down more recently [7, 41, 8, 31, 56]; see [51] for a broader survey. While these learned methods demonstrate impressive dexterity, they do not provide formal safety guarantees and typically require substantial tuning and reward engineering to work on hardware. Analytical approaches are much less common; existing work often exploits mechanical compliance and underactuation to reduce the planning burden [24, 3]. Suh et al. [42] propose a contact-trust-region formulation for contact-rich MPC and demonstrate palm-up cube reorientation with a fully actuated Allegro hand. To the best of our knowledge, our work is the first to achieve palm-down and variable-wrist-pose in-hand reorientation on a fully actuated general-purpose dexterous hand without relying on either machine learning or mechanical compliance.

III. PRELIMINARIES

Our method draws on differential geometry, pullback bundle dynamical systems, and control barrier functions, which we introduce in this section. Before formalizing each construct, we anchor the exposition with the following running example.

Example 1 (Reaching with obstacle avoidance). A 7-DOF Franka Emika Panda arm is to reach a goal end effector pose while avoiding a fixed workspace obstacle. The configuration is given by seven joint angles $\sigma = (\sigma^1, \dots, \sigma^7)$ with joint limits $\sigma_-^j \leq \sigma^j \leq \sigma_+^j$. Forward kinematics yields the end effector pose and the pose of each link for collision checking. The motion policy must reconcile objectives on heterogeneous spaces:

- *End effector tracking* in end effector pose.
- *Obstacle avoidance* in link-obstacle distances.
- *Joint limits* in the configuration manifold.

This system is implemented in simulation in Section VII.

A. Differential geometry and geometric mechanical systems

We present a brief introduction to Riemannian geometry; for further details, we refer the reader to [5, 18]. Let M be a smooth m -dimensional manifold with tangent bundle TM and cotangent bundle T^*M . A *Riemannian metric* g is a smooth assignment of an inner product $g_p: T_pM \times T_pM \rightarrow \mathbb{R}$ to each point $p \in M$; the pair (M, g) is called a *Riemannian manifold*. The metric induces the sharp isomorphism $g^\sharp: T^*M \rightarrow TM$, which maps each covector $\alpha \in T_x^*M$ to the unique vector $g^\sharp(\alpha) \in T_xM$ satisfying

$$g_x(g^\sharp(\alpha), v) = \alpha(v) \quad \text{for all } v \in T_xM.$$

TABLE I
NOTATION SUMMARY.

| Symbol | Meaning / Definition |
|---|---|
| <i>Manifolds and Maps</i> | |
| M | Configuration manifold (dimension m) |
| N_i | Task manifold for task i (dimension n_i) |
| TM, T^*M | Tangent and cotangent bundles of M |
| $(p, v) \in TM$ | Tangent-bundle point; $v \in T_pM$ |
| $f_i : M \rightarrow N_i$ | Task map |
| df_i, Jf_i | Differential and Jacobian of f_i |
| $\dot{J}f_i$ | Time derivative of Jf_i , i.e. $v^\ell \partial^2 f_i / \partial x^\ell \partial x^j$ |
| $F_i : TM \rightarrow TN_i$ | Higher-order task map, $(p, v) \mapsto (f_i(p), (df_i)_p(v))$ |
| (∇df) | Second fundamental form of f |
| <i>Riemannian Geometry</i> | |
| g_M, g_i | Riemannian metrics on M and N_i |
| $\langle \cdot, \cdot \rangle_g$ | Inner product of metric g ; on N also $\langle \cdot, \cdot \rangle_N$ |
| ${}^M\nabla, {}^{N_i}\nabla$ | Levi-Civita connections on M and N |
| ${}^M\Gamma_{ij}^k, {}^{N_i}\Gamma_{\beta\gamma}^\alpha$ | Christoffel symbols on M and N |
| \sharp | Sharp operator, $g^\sharp : T^*M \rightarrow TM$ (and task manifold analogue) |
| $\text{grad } h$ | Riemannian gradient: $\langle \text{grad } h, v \rangle_g = dh(v)$ |
| <i>PBDS Framework</i> | |
| $\sigma(t), \dot{\sigma}, \ddot{\sigma}$ | Configuration curve on M ; velocity and coordinate acceleration |
| $x(t)$ | task manifold curve on N |
| Φ_i | Potential function on N_i |
| $\mathcal{F}_{D,i}$ | Dissipative force map $TN_i \rightarrow T^*N_i$ |
| $\mathcal{F} \subset T^*M$ | SMCS input codistribution |
| w_i | Weighting pseudometric (task priority) |
| $w_i^\alpha \in \mathbb{R}^{n_i \times n_i}$ | Lower-right block of w_i in local coordinates |
| $u \in T^*M$ | Control input (generalized force one-form) |
| $\mathcal{D}_{(p,v)}$ | Affine distribution of second-order vectors at (p, v) |
| $\mathcal{A}_{\text{safe}}$ | Safe acceleration set (pullback CBFs) |
| π_{VB}, π_A | Vertical-bundle and actuation projections |
| \bar{a} | Optimal autonomous acceleration (27) |
| f_l, u_l, g_l, w_l | Control task map, input $u_l \in T^*N_l$, behavior metric, weighting pseudometric |
| <i>Control Barrier Functions (ECBF)</i> | |
| $h_0 : N \rightarrow \mathbb{R}$ | Safety function (indexed $h_{0,j}$ for multiple constraints) |
| $\mathcal{C}_0 \subseteq \mathcal{S}$ | Safe set $\{x : h_0(x) \geq 0\}$, contained in informal safe region $\mathcal{S} \subseteq N$ |
| $\alpha \in \mathcal{K}_\infty^e$ | Extended class- \mathcal{K}_∞ function |
| η_b | Barrier state $[h_0, \dot{h}_0, \dots, h_0^{(r-1)}]^\top$ |
| κ | ECBF gain vector; $\kappa^\top \in \mathbb{R}^{1 \times r}$ (scalars κ_1, κ_2 for $r=2$) |
| ν_i, \mathcal{C}_i^ν | Auxiliary functions and sets in the ECBF recursion; p_i are negated eigenvalues of $F - G\kappa^\top$ |
| <i>Backstepping CBF (BCBF)</i> | |
| $h(x, \dot{x})$ | Lifted BCBF candidate on TN |
| $\xi, \tilde{\xi}$ | Nominal and safe velocity fields on N (pre/post-filter) |
| $\xi_{\text{HS}}, \alpha_{\text{HS}}, \beta_{\text{HS}}, \lambda_{\text{HS}}$ | Half-Sontag safety filter quantities |
| $\Omega_0 \supset \mathcal{C}_0$ | Open domain on which ξ is a strict CBF |
| δ, ε | Strict-margin augmentation and BCBF scaling |
| ϵ_{pad} | Numerical padding for h_0 (distinct from ε) |
| e_x | Velocity error $\dot{x} - \tilde{\xi}$ |

The *Riemannian gradient* of a smooth function h is defined by $\text{grad } h = g^\sharp(dh)$, or equivalently, $\langle \text{grad } h, v \rangle_g = dh(v)$ for all tangent vectors v . The Levi-Civita connection induced by g is denoted by ∇ , so that the covariant acceleration along a curve $\sigma(t)$ is given by $\nabla_{\dot{\sigma}}\dot{\sigma}$. When two manifolds M and N must be disambiguated, we write ${}^M\nabla, {}^N\nabla$ for their Levi-Civita connections and ${}^M\Gamma, {}^N\Gamma$ for the corresponding Christoffel symbols.

The Euler-Lagrange equations for a mechanical system admit a coordinate-free formulation on Riemannian manifolds, known as a *simple mechanical control system*:

Definition III.1 (Simple Mechanical Control System). A *simple mechanical control system* (SMCS) [5] is a tuple $(M, g, \Phi, \mathcal{F})$, where M is an m -dimensional smooth manifold (the *configuration manifold*), g is a Riemannian metric on M (the *kinetic energy metric*), $\Phi \in C^\infty(M)$ is the *potential function*, and $\mathcal{F} \subset T^*M$ is a rank- p codistribution on M (the *input codistribution*). The equations of motion are

$$\nabla_{\dot{\sigma}}\dot{\sigma} = -\text{grad } \Phi \circ \sigma + u^\sharp, \quad (1)$$

where $u \in \mathcal{E}(\sigma, \mathcal{F})$, the space of sections of T^*M along σ such that $u(t) \in \mathcal{F}_{\sigma(t)}$, is the control one-form. In coordinates, (1) becomes

$$\begin{aligned} \ddot{\sigma}^k(t) + \Gamma_{ij}^k(\sigma(t)) \dot{\sigma}^i(t) \dot{\sigma}^j(t) \\ = -g^{kl}(\sigma(t)) \frac{\partial \Phi}{\partial \sigma^l}(\sigma(t)) + g^{kl}(\sigma(t)) u_l(t), \end{aligned} \quad (2)$$

where Γ_{ij}^k are the Christoffel symbols of g and u_l are the components of u .

Remark 1. SMCS is control affine since the sharp map is a linear operator.

Remark 2. g^{kl} converts the force covectors u and $d\Phi$ into accelerations. When g is the kinetic energy metric, g^{kl} plays the role of an inverse mass matrix. $\Gamma_{ij}^k \dot{\sigma}^i \dot{\sigma}^j$ corrects the coordinate acceleration $\ddot{\sigma}^k$ into a true acceleration on the curved manifold similar to the Coriolis and centripetal terms.

Example 1 (continued). For the 7-DOF arm reaching task, the configuration manifold is the open box of joint limits

$$M = \prod_{j=1}^7 (\sigma_-^j, \sigma_+^j) \subset \mathbb{R}^7,$$

equipped with the flat product metric $g_M = \delta_{ij}$. Although the physical joint limits are closed, we use strict inequalities so that M is a smooth manifold without boundary; the distinction makes no difference in practice.

B. Pullback bundle dynamical system (PBDS)

In order to construct a motion policy that combines multiple desired task behaviors, [6] proposes designing a stable mechanical control system (SMCS) on each task manifold of interest and combining them via a weighted least-squares optimization over the resulting task accelerations. Consider smooth *task maps* $f_i : M \rightarrow N_i$, $i = 1, \dots, K$, that map the configuration manifold M (of dimension m) to task manifolds N_i (of dimension n_i). On each N_i one designs an SMCS that exhibits a desired behavior by choosing a potential function $\Phi_i : N_i \rightarrow \mathbb{R}$, a Riemannian metric g_i on N_i (together with its Levi-Civita connection ∇_i), and a dissipative force

map $\mathcal{F}_{D,i} : TN_i \rightarrow T^*N_i$. We illustrate the role of f_i , g_i , Φ_i , and $\mathcal{F}_{D,i}$ on the running 7-DOF arm scenario (Example 1) in the continuation block following Definition III.2. Together, these determine a second-order dynamical system on TN_i , which can be pulled back to a dynamical system on the *pullback bundle*

$$f_i^*TN_i = \coprod_{p \in M} \pi_{N_i}^{-1}(f_i(p)), \quad (3)$$

where \coprod denotes the disjoint union and $\pi_{N_i} : TN_i \rightarrow N_i$ is the tangent bundle projection. Concretely, the fiber of $f_i^*TN_i$ over a point $p \in M$ is the tangent space $T_{f_i(p)}N_i$. The framework of [6] constructs on $f_i^*TN_i$ a pullback connection $f_i^*\nabla_i$, a compatible pullback metric $f_i^*g_i$, pullback dissipative forces $f_i^*\mathcal{F}_{D,i}$, and a pullback gradient operator $f_i^*\text{grad } \Phi_i$, all of which are globally well-defined.

Given these pullback constructions, two operators are introduced that capture, respectively, the *desired* task manifold acceleration produced by each local PBDS and the *resulting* task manifold acceleration induced by a candidate configuration manifold acceleration. The first operator,

$$S_i : TM \rightarrow T(f_i^*TN_i), \quad (4)$$

maps a state $(p, v) \in TM$ to the *desired* pullback task acceleration. It is defined as the vertical-bundle projection of the velocity of the local PBDS curve at (p, v) :

$$S_i(p, v) = \pi_{\text{VB}}(G_i(p, v)) = \left((f_i^*df_i)_p(v), (0, \dot{\gamma}_{v,p,i}^a(0)) \right), \quad (5)$$

where G_i encodes the dynamics of the local PBDS defined by $(f_i, g_i, \Phi_i, \mathcal{F}_{D,i})$, and $\dot{\gamma}_{v,p,i}^a(0)$ is the resulting task manifold acceleration. The pullback construction ensures that S_i is globally well-defined; see [6] for a detailed derivation. In local coordinates, the desired acceleration evaluates to

$$\begin{aligned} (\dot{\gamma}_{v,p,i}^a)^\alpha(0) &= g_i^{\alpha\beta}(f_i(p)) \mathcal{F}_{D,i}^\beta((df_i)_p(v)) \\ &\quad - \frac{\partial \Phi_i}{\partial y_i^\beta}(f_i(p)) g_i^{\alpha\beta}(f_i(p)) \\ &\quad - v^\ell (Jf_i)_h^\beta(p) v^h (Jf_i)_\ell^\gamma(p) (\Gamma_i)_{\gamma\beta}^\alpha(f_i(p)). \end{aligned} \quad (6)$$

The second operator is defined as

$$Z_i : TTM \rightarrow T(f_i^*TN_i), \quad (7)$$

maps a candidate configuration manifold acceleration to its *resulting* task manifold acceleration. It is defined as the vertical-bundle projection of the differential of the pullback differential $f_i^*df_i$:

$$Z_i((p, v), a) = \pi_{\text{VB}}\left(d(f_i^*df_i)_{(p,v)}(a)\right). \quad (8)$$

For $a = ((p, v), (v, a^a)) \in \mathcal{D}_{(p,v)}$ (defined below), the local expression is

$$Z_i(a) = \left((f_i^*df_i)_p(v), (0, Jf_i(p)a^a + \dot{J}f_i(p, v)v) \right), \quad (9)$$

where $(\dot{J}f_i)_j^\alpha(p, v) = v^\ell \partial^2 f_i^\alpha / \partial x^\ell \partial x^j$.

Let $\mathcal{D}_{\dot{\sigma}(t)} \subseteq T_{\dot{\sigma}(t)}TM$ denote the affine distribution of second-order vectors at $\dot{\sigma}(t) = (p, v) \in TM$, i.e. $\mathcal{D}_{(p,v)} = \{((p, v), (v, a^a)) : a^a \in \mathbb{R}^n\}$. Additionally, let w_i be a Riemannian *weighting pseudometric* on TN_i that controls the priority of task i relative to the other tasks, and let $F_i : TM \rightarrow TN_i$, $(p, v) \mapsto (f_i(p), (df_i)_p(v))$ be the

higher-order task map. A *best-compromise* acceleration is then obtained by minimizing the weighted sum of task-acceleration tracking errors:

Definition III.2 (Multi-Task PBDS, [6, Definition III.2]). Let $\{f_i : M \rightarrow N_i\}_{i=1,\dots,K}$ be smooth task maps for Riemannian task manifolds (N_i, g_i) , with corresponding smooth potential functions $\Phi_i : N_i \rightarrow \mathbb{R}$, dissipative forces $\mathcal{F}_{D,i} : TN_i \rightarrow T^*N_i$, and weighting pseudometrics (positive semidefinite Riemannian metric) w_i on TN_i . Then the set $\{(f_i, g_i, \Phi_i, \mathcal{F}_{D,i}, w_i)\}_{i=1,\dots,K}$ forms a *multi-task PBDS* with curves $\sigma : [0, \infty) \rightarrow M$ satisfying

$$\begin{cases} \ddot{\sigma}(t) = \arg \min_{a \in \mathcal{D}_{\dot{\sigma}(t)}} \sum_{i=1}^K \frac{1}{2} \|Z_i(a) - S_i(\dot{\sigma}(t))\|_{F_i^*w_i}^2, \\ \dot{\sigma}(0) = (p_0, v_0). \end{cases} \quad (10)$$

The dynamics (10) are globally well-defined and smooth on TM ; see [6] for details. Since (10) is a least-squares problem over the free variable $a^a \in T_pM$, it admits the local closed-form solution

$$\ddot{\sigma} = \left(\sum_{i=1}^K Jf_i^\top(w_i^a \circ F_i) Jf_i \right)^\dagger \left(\sum_{i=1}^K Jf_i^\top(w_i^a \circ F_i) A_i \right), \quad (11)$$

where $(\cdot)^\dagger$ is the pseudoinverse, $w_i^a \in \mathbb{R}^{n_i \times n_i}$ is the lower-right block of the local matrix representation of w_i (the only block that contributes after the vertical-bundle projections in S_i and Z_i), and

$$A_i = \dot{\gamma}_{\dot{\sigma},i}^a(0) - \dot{J}f_i(\sigma, \dot{\sigma})\dot{\sigma}, \quad (12)$$

with $\dot{\gamma}_{\dot{\sigma},i}^a(0)$ given by (6) evaluated at $p = \sigma(t)$, $v = \dot{\sigma}(t)$.

Remark 3. The multi-task PBDS (10) outputs the coordinate acceleration $\ddot{\sigma}$. The optimization variable $a^a \in T_pM$ in (10) represents the coordinate second derivative $d^2\sigma^k/dt^2$, and the operators S_i and Z_i involve only the task manifold metrics g_i and weights w_i . Consequently, the PBDS framework is oblivious to the geometry of M . Intuitively, PBDS produces the *task manifold* desired behavior, which is independent of the robot's *configuration manifold* dynamics. In practice, an acceleration tracking controller must be used to track $\ddot{\sigma}$ and account for the configuration manifold dynamics, which stems from the physical system dynamics.

Example 1 (continued). The end effector tracking objective decomposes into orientation tracking on $\text{SO}(3)$ and position tracking on \mathbb{R}^3 ; per [6], an additional joint space damping task is required for stability. This yields 3 SMCSs on a different manifold:

- *Orientation tracking* on $N_{\text{ori}} = \text{SO}(3)$, task map $f_{\text{ori}} : M \rightarrow \text{SO}(3)$ given by the end effector orientation from forward kinematics, attractor potential Φ_{ori} centered at a goal orientation, and linear damping $\mathcal{F}_{D,\text{ori}}$. In the implementation we use the unit quaternion parameterization (i.e. the universal double cover $\mathbb{S}^3 \rightarrow \text{SO}(3)$); see Appendix C.
- *Position tracking* on $N_{\text{pos}} = \mathbb{R}^3$ with the identity metric, task map $f_{\text{pos}} : M \rightarrow \mathbb{R}^3$ given by the end effector position, attractor potential Φ_{pos} centered at a goal position, and linear damping $\mathcal{F}_{D,\text{pos}}$.

- *Joint space damping* with identity task map $f_{\text{jd}} = \text{id}_M$, no potential, and linear damping $\mathcal{F}_{D,\text{jd}}$.

In the implementation, the attractor and damping of each tracking SMCS are realized as separate PBDS tasks on the same manifold, so the multi-task PBDS comprises 5 tasks total (orientation attractor, orientation damping, position attractor, position damping, and joint space damping). These tasks compose into a multi-task PBDS through Definition III.2 with weighting pseudometrics w_i tuned by task priority.

C. Control barrier functions (CBFs)

While multi-task PBDS (11) allows encouraging safe behavior through metric-based constraints [6, Section III.C], it does not provide a hard guarantee. Additionally, metric-based constraints are ill-defined in unsafe regions. We now review control barrier functions, which provide the tools to overcome this limitation.

Consider a safety problem on a task manifold N (e.g. obstacle avoidance constraints for the end effector), where $\mathcal{S} \subseteq N$ denotes a safe region: the system is safe if the state $x \in \mathcal{S}$.

Definition III.3 (Safety function). A smooth function $h_0 : N \rightarrow \mathbb{R}$ with 0 as a regular value and whose superlevel set satisfies

$$\mathcal{C}_0 \triangleq \{x \in N \mid h_0(x) \geq 0\} \subseteq \mathcal{S}, \quad (13)$$

is called a *safety function*.

Forward invariance of the set \mathcal{C}_0 implies that any trajectory initialized in \mathcal{C}_0 remains in \mathcal{C}_0 for all future times, ensuring that the system always satisfies safety constraints. Safety may be enforced by selecting appropriate control inputs u that govern the evolution of h_0 . To characterize how u relates to h_0 , we present two families of techniques from the literature. Throughout, we let $\alpha \in \mathcal{K}_\infty^e$ denote an extended class- \mathcal{K}_∞ function, i.e., a continuous, strictly increasing function $\alpha : \mathbb{R} \rightarrow \mathbb{R}$ satisfying $\alpha(0) = 0$ and $\lim_{r \rightarrow \pm\infty} \alpha(r) = \pm\infty$. For a task manifold CBF we write $(x, \dot{x}) \in TN$ for a tangent-bundle point, where $\dot{x} \in T_x N$; we fix a Riemannian metric g on N , and the norm $\|\cdot\|_N$, gradient grad , and inner product $\langle \cdot, \cdot \rangle_N$ below are all taken with respect to g .

We first introduce the class of systems under consideration. A *control-affine system* on a manifold N takes the form

$$\dot{x} = f_0(x) + B(x)u, \quad x \in N, u \in U \subseteq \mathbb{R}^{m_u}, \quad (14)$$

where f_0 is a drift vector field on N , $B(x) = (B_1(x), \dots, B_{m_u}(x))$ collects m_u input vector fields with $B(x)u = \sum_{k=1}^{m_u} B_k(x)u_k \in T_x N$, and U is the set of admissible inputs. In the context of the SMCS (1), the state is $(x, \dot{x}) \in TN$ and the control input u enters through the acceleration, so the system has relative degree two with respect to a safety function $h_0(x)$.

1) *Exponential CBF (ECBF)*: When the safety constraint h_0 has relative degree $r \geq 2$ with respect to the control input, i.e. the first $r-1$ time derivatives of h_0 are independent of u and u first appears explicitly in $h_0^{(r)}$, the standard relative-degree-one CBF condition cannot be applied directly. The

exponential CBF (ECBF) framework of [27] addresses this by enforcing a linear constraint on the r th derivative of h_0 .

Define

$$\eta_b(x) = [h_0(x), \dot{h}_0(x), \dots, h_0^{(r-1)}(x)]^\top. \quad (15)$$

The input-output linearized dynamics of h_0 under the virtual input $\mu = h_0^{(r)}$ take the linear form $\dot{\eta}_b = F\eta_b + G\mu$, $h_0 = C\eta_b$, where (F, G, C) is in controllable canonical form. Consider the input constraint $\mu \geq -\kappa^\top \eta_b(x)$ with gains $\kappa^\top = [\kappa_1, \dots, \kappa_r] \in \mathbb{R}^{1 \times r}$. By the comparison lemma, under this constraint, $h_0(x(t)) \geq C e^{(F-G\kappa^\top)t} \eta_b(x_0)$. Therefore, safety ($h_0 \geq 0$) is guaranteed if the right-hand side remains nonnegative. To this end, [2] introduces a family of auxiliary functions ν_i and corresponding sets \mathcal{C}_i^ν :

$$\nu_0(x) = h_0(x), \quad \nu_i(x) = \dot{\nu}_{i-1}(x) + p_i \nu_{i-1}(x), \quad i = 1, \dots, r, \quad (16)$$

$$\mathcal{C}_i^\nu = \{x : \nu_i(x) \geq 0\}, \quad (17)$$

where p_1, \dots, p_r are the roots of the characteristic polynomial $F - G\kappa^\top$. Forward invariance of \mathcal{C}_0 then requires the following conditions on κ [2, 27]:

Definition III.4 (Exponential CBF [27, 2]). A safety function h_0 of relative degree r with respect to (14) is an *exponential CBF* if there exists a row vector $\kappa \in \mathbb{R}^r$ such that for all $x \in \text{Int}(\mathcal{C}_0)$,

$$\sup_{u \in U} h_0^{(r)}(x, u) \geq -\kappa^\top \eta_b(x). \quad (18)$$

results in $h_0(x(t)) \geq C e^{(F-G\kappa^\top)t} \eta_b(x_0) \geq 0$ whenever $h_0(x_0) \geq 0$.

Theorem III.1 ([2, Theorem 8]). *Suppose κ is chosen such that $F - G\kappa^\top$ is Hurwitz and totally negative (resulting in negative real poles), and its eigenvalues satisfy*

$$\lambda_i(F - G\kappa^\top) \geq -\frac{\dot{\nu}_{i-1}(x_0)}{\nu_{i-1}(x_0)}, \quad i = 1, \dots, r, \quad (19)$$

whenever $\nu_{i-1}(x_0) > 0$. Then $\mu \geq -\kappa^\top \eta_b(x)$ guarantees that $h_0(x)$ is an exponential CBF.

Remark 4. The totally-negative requirement is stronger than the standard Hurwitz condition, which permits complex eigenvalues with negative real parts. This is because the recursive argument in [2, Proposition 6 and Theorem 7] establishes forward invariance of each \mathcal{C}_i^ν by showing that $\dot{\nu}_{i-1} \geq 0$ on $\partial\mathcal{C}_{i-1}^\nu$ only when $p_i > 0$. The eigenvalue bound (19) couples the pole locations to the initial state: the poles cannot be placed arbitrarily fast without violating the requirement that $\nu_i(x_0) \geq 0$. In practice, the ECBF can be designed via pole placement, choosing p_i large enough for rapid convergence while satisfying (19). The constraint (18) is then enforced pointwise via a quadratic program (QP) at each time step.

Remark 5. Evaluating (18) requires computing time derivatives of h_0 up to order r , which in turn depend on time derivatives of the state up to order r . For the SMCS (1), which is second order, $r = 2$, and the two conditions reduce to: 1) $p_1, p_2 > 0$ with $\kappa_1 = p_1 p_2$ and $\kappa_2 = p_1 + p_2$, and 2) $p_1 \geq -\dot{h}_0(x_0)/h_0(x_0)$.

2) *Backstepping CBF (BCBF)*: An alternative to the ECBF that avoids computing higher-order derivatives, which amplify

measurement noise and incur additional autograd cost, is backstepping CBF (BCBF) [45]. In BCBF, the safety specification on N is lifted to a CBF on TN by introducing an auxiliary safe velocity field on N . Below we recite the key geometric generalization results in [36]. For simplicity, we assume the system is fully actuated ($\pi_{\mathcal{A}} = \text{Id}$) and refer the reader to [36] for the general (including underactuated) case and proof details. In practice, most manipulators are fully- or over-actuated.

Step 1: Construct a safe velocity field.: Given a nominal velocity field ξ on N (so $\xi(x) \in T_x N$), one applies a smooth safety filter to obtain a *safe velocity field* $\tilde{\xi} : N \rightarrow TN$, $x \mapsto \tilde{\xi}(x) \in T_x N$, that renders \mathcal{C}_0 forward invariant under first-order dynamics $\dot{x} = \tilde{\xi}(x)$. Following [36], one uses the *half-Sontag formula*:

$$\xi_{\text{HS}}(x) = \xi(x) + \lambda_{\text{HS}}(\alpha_{\text{HS}}(x), \beta_{\text{HS}}(x)) \text{grad } h_0, \quad (20)$$

where

$$\lambda_{\text{HS}}(\alpha_{\text{HS}}, \beta_{\text{HS}}) = \begin{cases} 0 & \text{if } \beta_{\text{HS}} = 0, \\ -\frac{\alpha_{\text{HS}} + \sqrt{\alpha_{\text{HS}}^2 + \beta_{\text{HS}}^2}}{2\beta_{\text{HS}}} & \text{if } \beta_{\text{HS}} \neq 0, \end{cases} \quad (21)$$

with

$$\begin{aligned} \alpha_{\text{HS}}(x) &= \alpha(h_0(x)) + \langle \text{grad } h_0, \xi(x) \rangle_N, \\ \beta_{\text{HS}}(x) &= \|\text{grad } h_0\|_N^2. \end{aligned} \quad (22)$$

The half-Sontag formula is a smooth, closed-form safety filter: it adds a correction along $\text{grad } h_0$ that is just large enough to enforce the CBF condition $d(h_0)_x \xi_{\text{HS}} \geq -\alpha(h_0(x))$, and vanishes when ξ already satisfies it. The nominal velocity field ξ can be chosen freely based on the application; a natural choice within the PBDS framework is $\xi(x) = -\text{grad } \Phi|_x$, which drives the system toward the minimum of the task potential.

To provide a strict margin for the backstepping step, one further augments ξ_{HS} with an additional gradient term:

$$\tilde{\xi} = \xi_{\text{HS}}(x) + \delta \text{grad } h_0, \quad (23)$$

for some $\delta > 0$.

Lemma III.2 (Lemma 2 in [36]). *Let h_0 be the smooth safety function and \mathcal{C}_0 be the safe set as in Definition III.3, and let $\alpha \in \mathcal{K}_{\infty}^e$. There exists an open set $\Omega_0 \supset \mathcal{C}_0$ on which the safe velocity field (23) satisfies the strict CBF condition*

$$d(h_0)_x \mu_x > -\alpha(h_0(x)), \quad \forall x \in \Omega_0. \quad (24)$$

Step 2: Lift to a CBF on TN .: Using the safe velocity field μ_x from Step 1, one defines a candidate CBF on the tangent bundle:

$$h(x, \dot{x}) = h_0(x) - \frac{\varepsilon}{2} \|\dot{x} - \mu_x\|_N^2, \quad (x, \dot{x}) \in TN, \quad (25)$$

where $\varepsilon > 0$ is a design parameter controlling the tradeoff between the safety margin and the allowable velocity deviation from μ_x .

Step 3: Enforce the BCBF constraint.:

Theorem III.3 (Backstepping CBF [45, 36]). *The function h in (25) is a CBF for the SMCS on $T\Omega_0$, i.e., there exist control*

inputs satisfying

$$\sup_{u \in U} \dot{h}(x, \dot{x}, u) > -\alpha(h(x, \dot{x})), \quad (26)$$

for all $(x, \dot{x}) \in T\Omega_0$.

Since $h(x, \dot{x}) \leq h_0(x)$ for all $(x, \dot{x}) \in TN$ by construction (25), forward invariance of the set $\{(x, \dot{x}) \in TN : h(x, \dot{x}) \geq 0\}$ immediately implies safety:

Corollary III.1 (Safety [36, Proposition 2]). *Let u be any locally Lipschitz control input satisfying $\dot{h}(x, \dot{x}, u) \geq -\alpha(h(x, \dot{x}))$ for all $(x, \dot{x}) \in T\Omega_0$. If $h(x(0), \dot{x}(0)) \geq 0$, then $x(t) \in \mathcal{C}_0$ for all $t \geq 0$.*

The BCBF approach has two practical advantages over the ECBF: it avoids computing second (or higher) derivatives of h_0 along the system dynamics, and the resulting CBF constraint (26) is affine in u , yielding a standard QP. Furthermore, the backstepping construction extends naturally to the Riemannian manifold setting via the geometric framework of [36]. Conversely, the ECBF has the advantage that its constraint (18) depends only on Lie derivatives of h_0 along the system vector fields and is therefore independent of any choice of Riemannian metric on N , whereas the BCBF relies on metric-dependent quantities ($\text{grad } h_0$, $\|\cdot\|_N$, $\nabla_{\dot{x}} \mu_x$, u^{\sharp}).

IV. PULLBACK CBF FOR TASK MANIFOLD SAFETY

While metric-based tasks for safety were proposed in [6], they present several limitations:

- 1) *Exit behavior:* due to the symmetry of the metric-based velocity field, the constraint task must be deactivated when attempting to leave the safe set.
- 2) *Undefined behavior upon violation:* the metric $g = \exp(1/(2x^2))$ used in [6] is undefined on the constraint boundary $x = 0$, so no recovery mechanism exists once the constraint is violated.
- 3) *No guaranteed constraint satisfaction:* the constraint is ultimately enforced as one task among many in the multi-task PBDS QP (10), and the desired acceleration produced by the constraint task may not be fully realized.

We address these limitations by incorporating *pullback control barrier functions*, which produce sufficient conditions for a configuration-controlled robot to enforce task manifold safety. For brevity, we drop the task manifold index in this section. Consider a safety set $\mathcal{S} \subseteq N$ and a safety function $h_0 : N \rightarrow \mathbb{R}$ that defines the safety of the system. We additionally make the following assumption, which is typically satisfied in fully actuated and redundant robotic manipulators:

Assumption 1 (Surjective submersive task map). The task map $f : M \rightarrow N$ is a surjective submersion, so the system is fully actuated on N .

Remark 6. The safe set in the configuration manifold is the preimage $f^{-1}(\mathcal{C}_0) = \{p \in M \mid (h_0 \circ f)(p) \geq 0\}$, which may not be connected. Since trajectories are continuous, forward invariance of $f^{-1}(\mathcal{C}_0)$ implies forward invariance of each connected component individually.

Since the PBDS framework builds on the SMCS (1), the system is second order. We present adaptations of both the

ECBF from [27, 2] and the BCBF from [45, 36] enforce task manifold safety constraints on configuration manifold acceleration. By constraining the configuration manifold acceleration to lie in a safe set $\mathcal{A}_{\text{safe}}$, we arrive at the following:

Definition IV.1 (Autonomous SafePBDS). The union of K PBDS behavior tasks and J pullback-CBF safety tasks,

$$\{(f_i, g_i, \Phi_i, \mathcal{F}_{D,i}, w_i)\}_{i=1,\dots,K} \cup \{(f_j, h_{0,j}, \kappa_j)\}_{j=1,\dots,J},$$

forms a multi-task PBDS with curves $\sigma : [0, \infty) \rightarrow M$ satisfying

$$\ddot{\sigma}(t) = \arg \min_{a \in \mathcal{D}_{\dot{\sigma}(t)} \cap \mathcal{A}_{\text{safe}}} \sum_{i=1}^K \frac{1}{2} \|Z_i(a) - S_i(\dot{\sigma}(t))\|_{F_i^* w_i}^2 \quad (27)$$

where $\mathcal{A}_{\text{safe}}$ is the intersection of the pullback-CBF constraints induced by the J safety tasks (Theorems IV.1 and IV.3). For a BCBF safety task, κ_j is replaced by its BCBF parameters (ξ_j, ε_j) .

The remainder of this section derives pullback CBFs under the ECBF and BCBF formulations. The key idea is to pullback the task manifold CBF conditions through the task map to obtain safety constraints expressed entirely in configuration manifold inputs and the robot's state.

For the purpose of SafePBDS, which is controlled by configuration manifold coordinate acceleration $\ddot{\sigma}$, we show that both the pullback ECBF and pullback BCBF yield linear constraints on $\ddot{\sigma}$. Therefore, safety can be enforced by adding linear constraints to the PBDS QP (10). By substituting in Equation (2), our derivation can be extended to general systems with non-flat configuration manifolds and SMCS dynamics (1).

A. Geometric Kinematics

While the safety constraint is defined in the task manifold N , the control input in PBDS is the configuration manifold coordinate acceleration $\ddot{\sigma}$. Thus, we begin by deriving the geometric kinematics relating the configuration and task manifold accelerations.

1) *Velocity Relationship*: Let $x(t)$ be a curve on N , $\sigma(t)$ be a curve on M where $f(\sigma(t)) = x(t)$. In local coordinates, differentiating $x^\alpha = f^\alpha(\sigma)$ with respect to time yields

$$\dot{x}^\alpha = \frac{\partial f^\alpha}{\partial \sigma^i} \dot{\sigma}^i = (Jf \dot{\sigma})^\alpha \quad (28)$$

where Jf is the Jacobian of the task map. In coordinate-free notation, this is $\dot{x} = df \dot{\sigma}$.

2) *Acceleration Relationship*: Differentiating the velocity relation gives the coordinate acceleration on N :

$$\ddot{x}^\alpha = \frac{d}{dt} \left(\frac{\partial f^\alpha}{\partial \sigma^j} \dot{\sigma}^j \right) \quad (29)$$

$$= \frac{\partial f^\alpha}{\partial \sigma^j} \ddot{\sigma}^j + \frac{\partial^2 f^\alpha}{\partial \sigma^k \partial \sigma^j} \dot{\sigma}^k \dot{\sigma}^j \quad (30)$$

The covariant accelerations on N and M are given in coordinates by

$$({}^N \nabla_{\dot{x}} \dot{x})^\alpha = \ddot{x}^\alpha + {}^N \Gamma_{\mu\nu}^\alpha \dot{x}^\mu \dot{x}^\nu \quad (31)$$

$$({}^M \nabla_{\dot{\sigma}} \dot{\sigma})^k = \ddot{\sigma}^k + {}^M \Gamma_{ij}^k \dot{\sigma}^i \dot{\sigma}^j \quad (32)$$

Rearranging (32) for $\ddot{\sigma}^k$ and substituting into (30) yields

$$\ddot{x}^\alpha = \frac{\partial f^\alpha}{\partial \sigma^k} ({}^M \nabla_{\dot{\sigma}} \dot{\sigma})^k + \left(\frac{\partial^2 f^\alpha}{\partial \sigma^i \partial \sigma^j} - \frac{\partial f^\alpha}{\partial \sigma^k} {}^M \Gamma_{ij}^k \right) \dot{\sigma}^i \dot{\sigma}^j \quad (33)$$

Substituting (33) into (31) and using $\dot{x}^\mu = \frac{\partial f^\mu}{\partial \sigma^i} \dot{\sigma}^i$ yields

$$\begin{aligned} ({}^N \nabla_{\dot{x}} \dot{x})^\alpha &= \frac{\partial f^\alpha}{\partial \sigma^k} \underbrace{({}^M \nabla_{\dot{\sigma}} \dot{\sigma})^k}_{df({}^M \nabla_{\dot{\sigma}} \dot{\sigma})} \\ &+ \underbrace{\left(\frac{\partial^2 f^\alpha}{\partial \sigma^i \partial \sigma^j} - \frac{\partial f^\alpha}{\partial \sigma^k} {}^M \Gamma_{ij}^k + {}^N \Gamma_{\mu\nu}^\alpha \frac{\partial f^\mu}{\partial \sigma^i} \frac{\partial f^\nu}{\partial \sigma^j} \right)}_{(\nabla df)(\dot{\sigma}, \dot{\sigma})} \dot{\sigma}^i \dot{\sigma}^j \end{aligned} \quad (34)$$

In coordinate-free notation:

$${}^N \nabla_{\dot{x}} \dot{x} = df ({}^M \nabla_{\dot{\sigma}} \dot{\sigma}) + (\nabla df)(\dot{\sigma}, \dot{\sigma}) \quad (35)$$

where $(\nabla df)(\dot{\sigma}, \dot{\sigma})$ denotes the second fundamental form of the map f as in [11].

B. Task manifold ECBF

As (35) is a relative degree 2 system, we examine the first and second derivatives of the constraint function h_0 along the system trajectories.

The first derivative of $h_0(x(t))$ is computed by the chain rule:

$$\dot{h}_0 = \frac{\partial h_0}{\partial x^\alpha} \dot{x}^\alpha. \quad (36)$$

For the second derivative, we differentiate (36) and substitute $\dot{x}^\alpha = \frac{\partial f^\alpha}{\partial \sigma^i} \dot{\sigma}^i$:

$$\ddot{h}_0 = \frac{\partial^2 h_0}{\partial x^\alpha \partial x^\beta} \dot{x}^\alpha \dot{x}^\beta + \frac{\partial h_0}{\partial x^\gamma} \ddot{x}^\gamma. \quad (37)$$

Substituting the coordinate acceleration expansion (30) into (37) and applying the chain rule identities

$$\frac{\partial^2 (h_0 \circ f)}{\partial \sigma^i \partial \sigma^j} = \frac{\partial^2 h_0}{\partial x^\alpha \partial x^\beta} \frac{\partial f^\alpha}{\partial \sigma^i} \frac{\partial f^\beta}{\partial \sigma^j} + \frac{\partial h_0}{\partial x^\gamma} \frac{\partial^2 f^\gamma}{\partial \sigma^i \partial \sigma^j}, \quad (38)$$

$$\frac{\partial (h_0 \circ f)}{\partial \sigma^i} = \frac{\partial h_0}{\partial x^\gamma} \frac{\partial f^\gamma}{\partial \sigma^i}, \quad (39)$$

yields

$$\ddot{h}_0 = \underbrace{\frac{\partial^2 (h_0 \circ f)}{\partial \sigma^i \partial \sigma^j} \dot{\sigma}^i \dot{\sigma}^j}_{\text{Natural Dynamics / Drift}} + \underbrace{\frac{\partial (h_0 \circ f)}{\partial \sigma^i} \ddot{\sigma}^i}_{\text{Control Term}}. \quad (40)$$

Applying the ECBF constraint from Definition III.4 with $r = 2$ to (40) yields the task manifold formulation:

Theorem IV.1 (Task manifold ECBF). *A safety function h_0 (Definition III.3) is a task manifold ECBF for a PBDS system if there exists $\ddot{\sigma} \in T_\sigma M$ such that*

$$\frac{\partial (h_0 \circ f)}{\partial \sigma^i} \ddot{\sigma}^i \geq -\frac{\partial^2 (h_0 \circ f)}{\partial \sigma^i \partial \sigma^j} \dot{\sigma}^i \dot{\sigma}^j - \kappa_2 \dot{h}_0 - \kappa_1 h_0, \quad (41)$$

where the gains κ_1, κ_2 satisfy the pole placement and initial-condition requirements in Definition III.4.

Proof. Substituting (40) into the ECBF condition $\ddot{h}_0 \geq -\kappa_2 \dot{h}_0 - \kappa_1 h_0$ and rearranging yields (41). By the comparison lemma, the evolution of h_0 satisfies

$$h_0(x(t)) \geq [1 \ 0] e^{\begin{bmatrix} 0 & 1 \\ -\kappa_1 & -\kappa_2 \end{bmatrix} t} \eta_b(x_0). \quad (42)$$

where $\eta_b = [h_0, \dot{h}_0]^\top$. With $\kappa_1 = p_1 p_2$ and $\kappa_2 = p_1 + p_2$ for $p_1, p_2 > 0$, the matrix exponential decays and the right-hand side remains nonnegative provided the initial state satisfies $x_0 \in \mathcal{C}_0 \cap \mathcal{C}_1^\nu$, i.e. $h_0(x_0) \geq 0$ and $\dot{h}_0(x_0) + p_1 h_0(x_0) \geq 0$. \square

Remark 7 (Metric independence). The constraint (41) involves only coordinate partial derivatives of $h_0 \circ f$ and the coordinate acceleration $\ddot{\sigma}^i$. Intuitively, $h_0 \circ f$ has no knowledge of N , so no geometric quantities of N appear. Since SafePBDS controls $\ddot{\sigma}$, the ECBF constraint contains no geometric quantities altogether.

Remark 8 (Covariant form for non-flat M). For any choice of Riemannian metric g_M on M , substituting $\ddot{\sigma}^k = ({}^M\nabla_{\dot{\sigma}} \ddot{\sigma})^k - M\Gamma_{ij}^k \dot{\sigma}^i \dot{\sigma}^j$ from (32) into (41) and using the Riemannian Hessian identity

$$\text{Hess}_M(h_0 \circ f)(\dot{\sigma}, \dot{\sigma}) = \frac{\partial^2(h_0 \circ f)}{\partial \sigma^i \partial \sigma^j} \dot{\sigma}^i \dot{\sigma}^j - \frac{\partial(h_0 \circ f)}{\partial \sigma^k} M\Gamma_{ij}^k \dot{\sigma}^i \dot{\sigma}^j, \quad (43)$$

yields the equivalent geometric form

$$d(h_0 \circ f)({}^M\nabla_{\dot{\sigma}} \dot{\sigma}) \geq -\text{Hess}_M(h_0 \circ f)(\dot{\sigma}, \dot{\sigma}) - \kappa_2 \dot{h}_0 - \kappa_1 h_0. \quad (44)$$

Remark 9 (Behavior outside \mathcal{C}_0). As long as h_0 is smooth and well defined outside \mathcal{C}_0 , the ECBF constraint guarantees the system will converge asymptotically to $\{x \mid h_0(x) \geq 0\}$ from any initial condition, with convergence rate governed by the eigenvalues $-p_1, -p_2$ via (42). This contrasts with the metric-based constraint enforcement in [6], where the metric $g = \exp(1/(2x^2))$ is undefined on the constraint boundary $h_0(x) = 0$. In practice, one may offset h_0 so that $h_0(x) \geq -\epsilon$ represents safety; the ECBF then drives the system to $\{x \mid h_0(x) \geq -\epsilon\}$ in finite time.

Example 1 (continued). *Workspace obstacle clearance* for robot link i is encoded on the task manifold $N_{\text{obs},i} = \mathbb{R}$ with task map $f_{\text{obs},i}: M \rightarrow N_{\text{obs},i}: \sigma \mapsto d(\text{geom}_i, \text{obs})$, the signed distance between the link i collision geometry and the obstacle. The safety function $h_{0,\text{obs},i}: N_{\text{obs},i} \rightarrow \mathbb{R}$ is given by $h_{0,\text{obs},i}(x) = x - d_{\min}$, where $d_{\min} \geq 0$ is a minimum clearance margin.

Joint limits for joint j are encoded on the task manifold $N_{\text{lim},j} = \mathbb{R}$ with task map $f_{\text{lim},j}: M \rightarrow N_{\text{lim},j}: \sigma \mapsto \sigma^j$. The safety functions $h_{0,j}^-, h_{0,j}^+ : N_{\text{lim},j} \rightarrow \mathbb{R}$ are $h_{0,j}^-(x) = x - \sigma_j^-$ and $h_{0,j}^+(x) = \sigma_j^+ - x$, for $j = 1, \dots, 7$. Both families have relative degree 2 with respect to the task state. Each safety function yields a linear constraint on $\ddot{\sigma}$ via Theorem IV.1.

C. Task manifold BCBF

Here we derive the BCBF safety constraint for the PBDS system. Note that by Assumption 1, the task map f is a surjective submersion, so the system is fully actuated on N .

Lemma IV.2. *In the task manifold PBDS setup, the BCBF candidate h given in (25) has time derivative*

$$\begin{aligned} \dot{h} = & \langle \text{grad } h_0, \dot{x} \rangle_N + \varepsilon \langle e_x, \nabla_{\dot{x}} \tilde{\xi} \rangle_N \\ & - \varepsilon \langle e_x, df({}^M\nabla_{\dot{\sigma}} \dot{\sigma}) + (\nabla df)(\dot{\sigma}, \dot{\sigma}) \rangle_N. \end{aligned} \quad (45)$$

The proof is given in Appendix A.

Theorem IV.3 (Task manifold BCBF). *The function h in (25) is a task manifold BCBF on $T\Omega_0$ for a PBDS system if*

$$\sup_{\ddot{\sigma} \in T_{\sigma} M} \dot{h}(\sigma, \dot{\sigma}, \ddot{\sigma}) > -\alpha(h(x, \dot{x})), \quad (46)$$

for all $(\sigma, \dot{\sigma}) \in TM$ with $(f(\sigma), df \dot{\sigma}) \in T\Omega_0$, where $x = f(\sigma)$, $\dot{x} = df \dot{\sigma}$, and h, \dot{h} are given by (25) and (45).

Proof. We verify (46) for all $(\sigma, \dot{\sigma}) \in TM$ with $(f(\sigma), df \dot{\sigma}) \in T\Omega_0$. Let $x = f(\sigma)$ and $\dot{x} = df \dot{\sigma}$. If $e_x = 0$ (i.e. $\dot{x} = \xi$), the ε -terms in (45) vanish and

$$\dot{h} = \langle \text{grad } h_0, \tilde{\xi} \rangle_N > -\alpha(h_0(x)) = -\alpha(h(x, \dot{x})), \quad (47)$$

where the inequality follows from Lemma III.2, and the last equality from $h(x, \dot{x}) = h_0(x)$ whenever $e_x = 0$ by (25). If $e_x \neq 0$, the term $-\varepsilon \langle e_x, df({}^M\nabla_{\dot{\sigma}} \dot{\sigma}) \rangle_N$ in (45) is affine in $\ddot{\sigma}$ by (32), with linear coefficient $-\varepsilon g_{\alpha\beta} e^\beta \frac{\partial f^\alpha}{\partial \sigma^k}$ (free index k on M). Since f is a submersion (Jf has full row rank) and $e_x \neq 0$, this coefficient is nonzero, so \dot{h} can be made arbitrarily large by choosing $\ddot{\sigma} \in T_{\sigma} M$, giving $\sup \dot{h} = +\infty$. \square

Remark 10. Assumption 1 ensures the system can be controlled in any direction on the tangent space of the task manifold, so the supremum over accelerations is unbounded if $e_x \neq 0$. This significantly simplifies the CBF design process.

Proposition 1 (Task manifold safety under BCBF). Let $\tilde{\xi}$ be a safe velocity field as in Lemma III.2, and let $\ddot{\sigma}$ satisfy the non-strict BCBF constraint $\dot{h} \geq -\alpha(h(x, \dot{x}))$ for all $(x, \dot{x}) \in T\Omega_0$. For any initial state with $h(x(0), \dot{x}(0)) \geq 0$, the closed-loop trajectory satisfies $x(t) \in \mathcal{C}_0$ for all $t \geq 0$.

Proof. Since $\alpha \in \mathcal{K}_\infty^e$ satisfies $\alpha(0) = 0$ and is strictly increasing, the non-strict constraint $\dot{h} \geq -\alpha(h)$ implies forward invariance of $\{h \geq 0\}$ by the comparison lemma, so $h(x(t), \dot{x}(t)) \geq 0$ for all $t \geq 0$. Since $h_0(x) \geq h(x, \dot{x})$ by (25), we conclude $h_0(x(t)) \geq 0$, and hence $x(t) \in \mathcal{C}_0$ for all $t \geq 0$. \square

Remark 11. Unlike the ECBF formulation, the BCBF depends on the choice of Riemannian metric on N : the safe velocity field $\tilde{\xi}$, the velocity error norm $\|e_x\|_N$, the covariant derivative $\nabla_{\dot{x}} \tilde{\xi}$, and the inner product in (45) all involve the metric. Intuitively, this is because the backstepping construction compares the current velocity \dot{x} against the safe velocity field $\tilde{\xi}$ using the metric on N .

Remark 12 (Strict vs. non-strict inequalities). The ECBF constraint (41) uses a non-strict inequality (\geq), following [2], with the regularity condition (that 0 is a regular value of h_0) assumed in Definition III.3. The BCBF constraint (46) uses a strict inequality ($>$), following [36]. The strict inequality is a stronger condition: it implies regularity [36, Remark 1], but not vice versa, since it additionally requires the existence of a control input that makes h strictly increase on $\{h = 0\}$. In the backstepping construction, this strict margin arises naturally from the $\delta \text{grad } h_0$ augmentation in (23). Note that the actual enforcement constraint used in Proposition 1 is the non-strict $\dot{h} \geq -\alpha(h)$; the strict inequality in Theorem IV.3 establishes that valid controls exist.

Remark 13 (Recovery outside \mathcal{C}_0). For the BCBF, if h_0 is smooth on all of N , the half-Sontag formula (20) produces a globally smooth safe velocity field $\tilde{\xi}$, so the candidate h (25) is well-defined on all of TN . The non-strict enforcement constraint $\dot{h} \geq -\alpha(h)$ is then satisfiable everywhere: when $e_x \neq 0$, full actuation makes the supremum unbounded; when $e_x = 0$, the half-Sontag formula guarantees $d(h_0)_x \tilde{\xi} \geq -\alpha(h_0(x))$ globally. If $h < 0$, a comparison argument using $\alpha \in \mathcal{K}_\infty^e$ shows $h(t) \rightarrow 0$, recovering safety. The domain Ω_0

from Lemma III.2 is only required for the strict inequality in Theorem IV.3, which establishes h as a CBF in the formal sense, but does not limit the practical recovery behavior.

Remark 14 (Practical considerations). In practice, finite differencing is often used to compute derivatives and smooth functions that are Lipschitz continuous but not everywhere differentiable. Additionally, due to numerical errors and latency, it is possible to get small CBF violations despite satisfying CBF constraints. Such issues may be mitigated by padding the safety function in (13) by a small margin $\epsilon_{\text{pad}} > 0$, which we apply in our implementation.

V. TASK MANIFOLD ACTIONS

In the autonomous SafePBDS framework (Definition IV.1), runtime decision-making is limited to adjusting the weighting pseudometrics w_i . However, one may wish to steer the system directly in a task manifold N , for example to incorporate commands from a higher-level planner or a learned policy. We therefore propose a mechanism that exposes control inputs while preserving the safety and compositional structure of the PBDS framework. We seek the following desiderata:

- 1) The action space may be lower dimensional than the configuration dimension m .
- 2) If so desired, all m dimensions may be controlled.
- 3) The action represents a residual force on top of the autonomous PBDS dynamics: when the action is zero, the system follows the autonomous behavior.

To this end, consider L control task maps $f_l : M \rightarrow N_l$, $l = 1, \dots, L$, each equipped with a weighting pseudometric w_l on TN_l and a Riemannian behavior metric g_l on N_l . These may coincide with some of the K autonomous task maps, or they may be entirely separate. For each control task, the user supplies a force-like input $u_l \in T^*N_l$, which is converted to an acceleration-level quantity via the sharp (musical isomorphism) $u_l^\sharp = g_l^\sharp(u_l) \in TN_l$ (in coordinates, $g_l^{-1} u_l$).

Definition V.1 (Steerable SafePBDS). Let \bar{a} be the optimal autonomous acceleration (27). Given L control inputs $u_l \in T^*N_l$, each defined in control task manifolds $f_l : M \rightarrow N_l$, the acceleration under action input is

$$\begin{aligned} \ddot{\sigma}(t) = & \arg \min_{a \in \mathcal{D}_{\dot{\sigma}(t)} \cap \mathcal{A}_{\text{safe}}} \sum_{i=1}^K \frac{1}{2} \|Z_i(a) - S_i(\dot{\sigma}(t))\|_{F_i^* w_i}^2 \\ & + \sum_{l=1}^L \frac{1}{2} \|Z_l(a) - Z_l(\bar{a}) - u_l^\sharp\|_{F_l^* w_l}^2. \end{aligned} \quad (48)$$

where S_i and Z_i are defined in Section III-B.

The first term is identical to the autonomous objective (10); the second term shifts the autonomous acceleration by the control input u_l^\sharp . Since Z_i is affine in the acceleration and the constraint set $\mathcal{D}_{\dot{\sigma}} \cap \mathcal{A}_{\text{safe}}$ is characterized by linear inequalities (Section IV), the steered problem (48) remains a convex QP and can be solved with the same infrastructure as the autonomous problem. Algorithm 1 summarizes the per-step procedure with the safety constraints written out.

Theorem V.1 (Autonomous behavior recovery). *If all control inputs $u_l = 0$, \bar{a} is a minimizer of (48).*

Algorithm 1 SafePBDS control step

Require: state $(\sigma, \dot{\sigma}) \in TM$; behavior tasks $\{(f_i, g_i, \Phi_i, \mathcal{F}_{D,i}, w_i)\}_{i=1}^K$; safety tasks indexed by \mathcal{J}_E (ECBF, Theorem IV.1) and \mathcal{J}_B (BCBF, Theorem IV.3); control tasks $\{(f_l, g_l, w_l, u_l)\}_{l=1}^L$.

Ensure: coordinate acceleration $\ddot{\sigma} \in T_\sigma M$.

1: $\mathcal{A}_{\text{safe}} \leftarrow T_\sigma M$

2: **for** each $j \in \mathcal{J}_E$ **do**

3: intersect $\mathcal{A}_{\text{safe}}$ with the half-space

$$\frac{\partial(h_{0,j} \circ f_j)}{\partial \sigma^k} a^k \geq -\frac{\partial^2(h_{0,j} \circ f_j)}{\partial \sigma^k \partial \sigma^\ell} \dot{\sigma}^k \dot{\sigma}^\ell - \kappa_{2,j} \dot{h}_{0,j} - \kappa_{1,j} h_{0,j}$$

4: **end for**

5: **for** each $j \in \mathcal{J}_B$ **do**

6: intersect $\mathcal{A}_{\text{safe}}$ with $\dot{h}_j(\sigma, \dot{\sigma}, a) \geq -\alpha_j(h_j(x_j, \dot{x}_j))$, where \dot{h}_j is given by (45)

7: **end for**

8: **QP 1** (autonomous safe acceleration):

$$\bar{a} \leftarrow \arg \min_{a \in \mathcal{A}_{\text{safe}}} \sum_{i=1}^K \frac{1}{2} \|Z_i(a) - S_i(\dot{\sigma})\|_{F_i^* w_i}^2$$

9: **QP 2** (steered acceleration): solve (48) for $\ddot{\sigma}$ given \bar{a} , with $a \in \mathcal{A}_{\text{safe}}$

10: **return** $\ddot{\sigma}$

Proof. Since Z_i is affine in a and the feasible set $\mathcal{D}_{\dot{\sigma}(t)} \cap \mathcal{A}_{\text{safe}}$ is characterized by linear constraints (Section IV), Equation (48) is a convex QP for which the KKT conditions are necessary and sufficient. Let $Aa \leq b$ denote the assembled linear constraints. Setting $u_l = 0$, the KKT stationarity condition of (48) is

$$\begin{aligned} \nabla_a \left[\sum_{i=1}^K \frac{1}{2} \|Z_i(a) - S_i\|_{F_i^* w_i}^2 \right. \\ \left. + \sum_{l=1}^L \frac{1}{2} \|Z_l(a) - Z_l(\bar{a})\|_{F_l^* w_l}^2 \right] + A^\top \lambda = 0, \end{aligned} \quad (49)$$

with $Aa \leq b$, $\lambda \geq 0$, and $\lambda^\top (Aa - b) = 0$.

By construction, $\bar{a} \in \mathcal{D}_{\dot{\sigma}(t)} \cap \mathcal{A}_{\text{safe}}$, so primal feasibility holds. Since Z_l is affine in a , the gradient of $\|Z_l(a) - Z_l(\bar{a})\|_{F_l^* w_l}^2$ vanishes at $a = \bar{a}$. The stationarity condition thus reduces to that of the autonomous problem (Definition IV.1), which \bar{a} satisfies with optimal autonomous problem dual variables $\bar{\lambda}$. As the feasible set is identical, all KKT conditions are satisfied and \bar{a} minimizes (48). \square

Remark 15. If one interprets \bar{a} as the resulting acceleration of a task manifold SMCS (Equation (1)), the second sum in (48) acts as an additional task manifold force.

Remark 16. \bar{a} is not necessarily a unique minimizer. In practice, since \bar{a} is used to warm start the numerical solver, it is typically selected when all $u_l = 0$.

Remark 17 (Desiderata). The three desiderata from the beginning of this section are satisfied by construction: (1) each control task map $f_l : M \rightarrow N_l$ may have $n_l \leq m$, so the total action dimension $\sum_{l=1}^L n_l$ can be less than m ; (2) if the control task maps collectively form a submersion (i.e., $\text{rank}[Jf_1^\top \cdots Jf_L^\top] = m$), all m degrees of freedom are controllable; and (3) Theorem V.1 guarantees that zero control input recovers the autonomous behavior.

Example 1 (continued). To expose a one-dimensional action space that selects among qualitatively distinct avoidance behaviors (e.g. routing the elbow on either side of a workspace obstacle), we use a control task map given by the coordinate projection onto joint 1,

$$f: M \rightarrow \mathbb{R}: \sigma \mapsto \sigma^1,$$

with identity behavior metric $g = 1$ and a positive scalar weighting pseudometric w . The action input $u \in \mathbb{R}$ injects an acceleration on joint 1; flipping the sign of u steers the arm into a different homotopy class of safe paths around the obstacle, while $u = 0$ recovers the autonomous behavior by Theorem V.1. This is demonstrated in Section VII-B.

VI. EVALUATION: SIMULATED \mathbb{S}^2 DOUBLE INTEGRATOR

Using a point robot on \mathbb{S}^2 , we first validate the theoretical properties of SafePBDS, in particular chart invariance, safety guarantees, multi-task composition, and effects of our action space design. All experiment configurations in this section are summarized in Table II. Additional details are available in Appendix B.

A. System Setup

a) System definition: Consider a point robot with unit mass traveling on the surface of a unit sphere. The configuration manifold of the robot is \mathbb{S}^2 , which has a natural embedding $\bar{\varphi}: \mathbb{S}^2 \hookrightarrow \mathbb{R}^3$. We consider the atlas $\{(U_N, \varphi_N), (U_S, \varphi_S)\}$ formed by the north and south pole stereographic projection charts. We can define corresponding maps $\bar{\varphi}_N: \mathbb{R}^2 \rightarrow \mathbb{R}^3$ and $\bar{\varphi}_S: \mathbb{R}^2 \rightarrow \mathbb{R}^3$ from chart coordinates (y_1, y_2) to embedding coordinates (x_1, x_2, x_3) .

b) Configuration Dynamics: Since PBDS does not account for the configuration manifold geometry (Remark 3), we treat the chart coordinates as \mathbb{R}^2 and integrate with a fourth-order Runge–Kutta scheme when simulating the system forward.

c) Task Dynamics: Following [6], we use the chart-to-embedding task map $f = \bar{\varphi}_\alpha: \mathbb{R}^2 \rightarrow \mathbb{R}^3$, with an attractor potential and dissipative force defined on the embedding \mathbb{R}^3 .

d) Safety constraint: We choose the safety task as staying at least arclength r from a point $x_o \in \mathbb{S}^2$. The safety function (Definition III.3) is $h_0(y) = \arccos(\bar{\varphi}_\alpha(y) \cdot x_o) - r$, with $\alpha \in \{N, S\}$ the active chart. The safe set is $\mathcal{C}_0 = \{y \in \mathbb{R}^2: h_0(y) \geq 0\}$. For BCBF, we set $\xi = 0$ and compare two metrics: the round metric $g_{ij} = \frac{4}{(1+\|y\|^2)^2} \delta_{ij}$, which is the pullback of the embedded \mathbb{S}^2 metric through the stereographic chart, and the flat metric δ_{ij} , which treats the chart coordinates as Euclidean.

B. Autonomous SafePBDS on \mathbb{S}^2

First, we validate the theoretical properties of ECBFs and BCBFs in SafePBDS without task manifold actions, shown in Figure 2(a–b).

a) Metric dependence and chart switching: As expected from Remark 7, ECBF is metric-independent. The two BCBF runs produce visibly different avoidance trajectories, since BCBF depends on the metric via $\text{grad } h_0$, $\|e_x\|_N$, and $\nabla_x \tilde{\xi}$; the BCBF also stays farther from the unsafe set boundary

TABLE II
RUN CONFIGURATIONS FOR THE \mathbb{S}^2 DOUBLE INTEGRATOR EXPERIMENTS.

| Run | Chart | Safety (chart, metric) | Action |
|--|-----------|------------------------|---------------------|
| <i>Autonomous: metric dependence and chart switching</i> | | | |
| (i) | N | ECBF (N) | – |
| (ii) | N | BCBF (N, round) | – |
| (iii) | N | BCBF (N, flat) | – |
| (iv) | Switching | ECBF (Switch) | – |
| (v) | Switching | BCBF (Switch, round) | – |
| <i>Autonomous: safety recovery</i> | | | |
| (vi) | N | ECBF (N) | – |
| (vii) | N | BCBF (N, round) | – |
| <i>Steered</i> | | | |
| (viii) | N | ECBF (N) | 0 (autonomous) |
| (ix) | N | ECBF (N) | $+u_\perp$ |
| (x) | N | ECBF (N) | $-u_\perp$ |
| (xi) | N | ECBF (N) | u_{unsafe} |
| (xii) | S | ECBF (S) | $-u_\perp$ |

due to the additional “safe velocity tracking error” penalty (second term in (45)). We verify chart switching by varying the configuration manifold chart between north and south. This is the safety-augmented analogue of the chart switching experiment in [6, Fig. 5]. Run (iv) matches the fixed-chart reference (i) on \mathbb{S}^2 , confirming that the SafePBDS produce consistent behavior across chart choices and can be deployed on atlases with multiple charts.

b) Safety recovery: To test recovery, we run both an ECBF and a BCBF trajectory starting from inside the unsafe set. Both formulations drive the system out of the unsafe region and converge to the goal. The ECBF constrains only $h_0 \geq 0$ and recovers along the most direct path. The BCBF’s safe velocity field deviation penalty causes its recovery to maintain a larger clearance from the obstacle boundary.

C. Steered SafePBDS on \mathbb{S}^2

Next, we demonstrate the properties of the task manifold actions (Section V). We use an ECBF safety constraint and an identity control task map $f_I = \text{id}: \mathbb{R}^2 \rightarrow \mathbb{R}^2$ on the active chart. We place the obstacle so the system faces two valid avoidance paths on each side of the obstacle. We run five trajectories (viii)–(xii), all shown in Figure 2(c):

- (viii) *Autonomous* ($u = 0$): the system breaks symmetry via a small positional offset and converges to one side.
- (ix) $+u_\perp$: a tangential action perpendicular to the start–goal geodesic with $\|u_\perp\| = 1$, steering the robot to the same side as the autonomous bias.
- (x) $-u_\perp$: flipping the sign steers the robot to the opposite side, resolving the topological ambiguity that the autonomous dynamics alone cannot.
- (xi) u_{unsafe} : a large action ($\|u\| = 10$) continuously pointing toward the obstacle center. SafePBDS prevents the constraint violation, and safety is maintained ($h_0(t) \geq 0$) despite the adversarial action.
- (xii) $-u_\perp$ on the south pole chart: we repeat (x) on the south pole chart to verify chart invariance under action inputs.

VII. EVALUATION: SIMULATED 7-DOF ROBOT ARM

We instantiate the running 7-DOF arm scenario (Example 1) in MuJoCo [46] to validate the pullback CBF for workspace

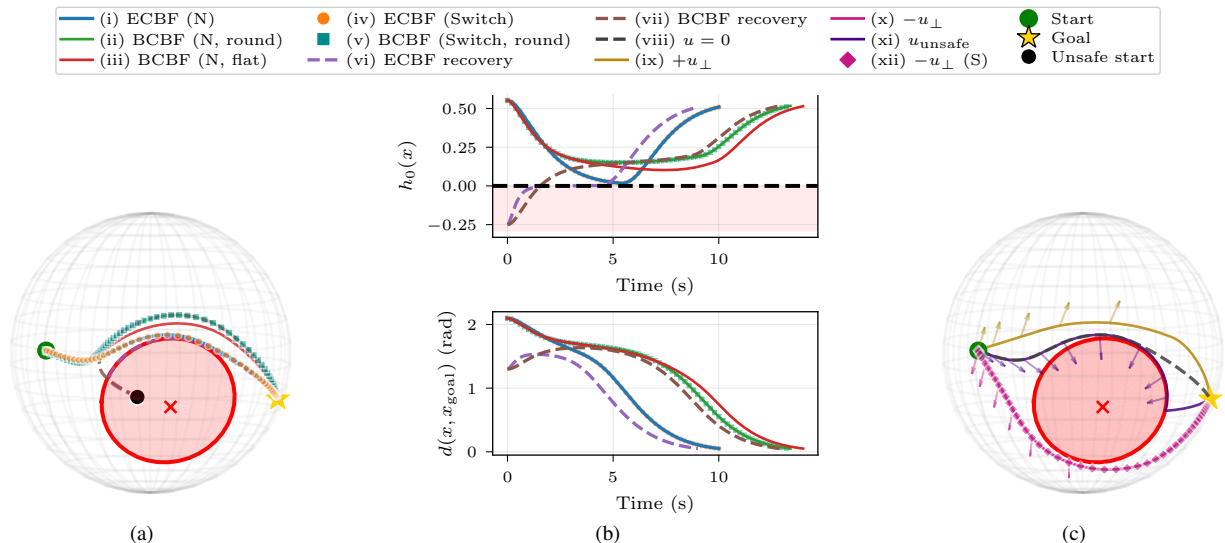


Fig. 2. Pullback CBF and action interface on \mathbb{S}^2 ; run indices match Table II and the legend. (a) Autonomous runs (i)–(vii) on the same scene; recovery runs (vi)–(vii) start inside the obstacle (\times marker). (b) $h_0(t)$ (top, shaded region is $h_0 < 0$) and geodesic distance to goal (bottom) for the same runs. (c) (viii)–(xii): tangential actions $\pm u_\perp$ select opposite homotopy classes, u_{unsafe} is clipped by the CBF, and (xii) repeats (x) on the south pole chart to verify chart invariance.

obstacle avoidance and the action space formulation for behavior modality selection. The arm starts at its home configuration; goals and obstacles vary per experiment as detailed below. Additional details are provided in Appendix C.

A. Autonomous SafePBDS on the 7-DOF Arm

We validate the pullback ECBF by comparing the full SafePBDS system against an ablation that removes the obstacle avoidance ECBFs while keeping joint limit ECBFs active. We test end effector tracking in two setups:

a) 6-DOF pose (SE(3)) tracking: A spherical obstacle (radius 0.10 m) is placed in between the start and goal positions. Figure 3(a) shows the full system (purple) deflecting around the obstacle, while the ablation (red, dashed) passes through it (penetration in orange); coordinate frames at equal arc length intervals show orientation converging to the goal.

b) Random orientation (SO(3)) tracking: We sample 50 scenarios that each track a goal orientation (a 30° – 150° rotation about a random axis) with a randomly placed spherical obstacle (radius 8 cm). Position tracking is forfeited due to the difficulty of randomly finding a reachable goal pose under the obstacle avoidance constraint. Scenarios with obstacle clearance below 5 cm are rejected and resampled. The full system is safe in all 50 runs ($\min h_{\text{obs}} = +0.010$), whereas the ablation violates the obstacle constraint in 11 of 50 runs ($\min h_{\text{obs}} = -0.080$); all runs reach the goal orientation.

B. Steered SafePBDS on the 7-DOF Arm

We validate the action interface (Section V) using end effector position (\mathbb{R}^3) tracking, which leaves four redundant degrees of freedom for the action input to exploit. When an obstacle lies in the robot’s workspace, the deterministic autonomous system defaults to one avoidance path, yet multiple safe paths exist. The action inputs allow an operator or high-level planner to select among these configurations with minimal input, instantiating the homotopy class selection application described in the running example (Example 1). We apply a brief constant scalar action on the first joint: $u = \pm 15$

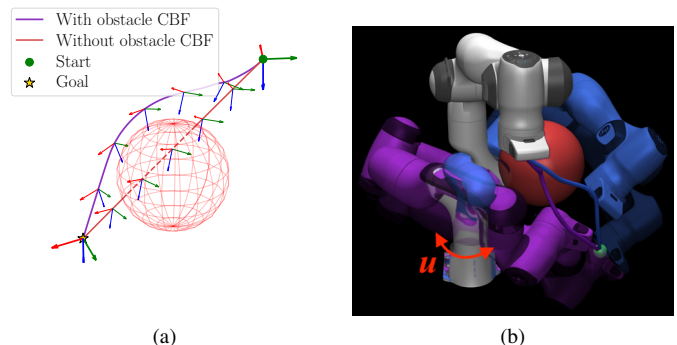


Fig. 3. 7-DOF arm: workspace safety and steered action. (a) Obstacle avoidance during 6-DOF pose tracking: the full system (purple) deflects around the obstacle, while the ablation without the obstacle CBF (red, dashed) passes through it (penetration in orange); coordinate frames at equal arc length intervals show orientation converging to the goal. (b) Behavior selection with action inputs and position-only (\mathbb{R}^3) tracking: flipping the sign of a single joint 1 action steers the forearm to opposite sides of the obstacle (blue vs. purple), while the end effector converges to the same position goal (green); the initial configuration (gray) and final silhouettes are shown for reference.

for the first 3.5 s, then $u = 0$. Flipping the sign of this scalar results in trajectories of distinct homotopy classes: the $+u$ arm (blue) passes on one side while the $-u$ arm (purple) passes on the other (Figure 3(b)). Both trajectories converge to the goal while preserving safety. This highlights a key practical advantage of the steered SafePBDS: qualitatively different behaviors can be selected using simple, transient inputs, while the autonomous dynamics and safety constraints handle the details of the motion.

VIII. EVALUATION: HARDWARE EXPERIMENTS

We validate SafePBDS on hardware with two dexterous manipulation tasks: autonomous grasping of diverse household objects (Section VIII-C) and in-hand reorientation via finger gaing (Section VIII-D).

A. Robot Setup

All hardware experiments are performed with a Franka Emika Panda 7-DOF arm and a Wonik Allegro 16-DOF



Fig. 4. Hardware setup for the dexterous manipulation experiments: a 7-DOF arm equipped with a 16-DOF dexterous hand, a camera for runtime perception, and household objects used for grasping and in-hand reorientation.

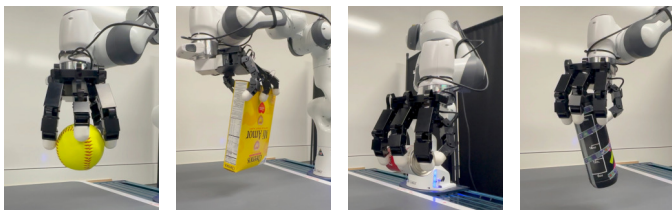


Fig. 5. 4-finger grasp examples across representative object categories. All four fingers form a force closure grasp under SafePBDS, with the per-finger fingertip-to-object distance and centroid centering action tasks driving the fingertips onto the object surface.

dexterous hand [1], for a combined 23-DOF system. The arm is controlled at 20 Hz via the Deoxys framework [60], using the joint impedance controller when PBDS is active, and the joint position controller for the pre-grasp approach. The hand is commanded over ZMQ using a mix of joint-level PD and Cartesian fingertip impedance modes. Runtime perception uses an Intel RealSense D435 camera with Segment Anything [17] for segmentation and FoundationPose [52] for 6-DOF pose tracking at 10 Hz, which updates an object pose in a MuJoCo [46] simulation that the PBDS controller runs against. Object meshes are obtained by scanning with KIRI Engine [16] on a LiDAR-equipped iPhone. Full control, perception, and modeling details are given in Appendix D.

B. Manifold Setup and Task Specification

The configuration manifold $M = \prod_{i=1}^m (\sigma_-^i, \sigma_+^i)$ is an open subset of \mathbb{R}^m equipped with the flat product metric. The combined arm-hand system used for dexterous grasping has $m = 23$; for in-hand reorientation (Section VIII-D) the arm is controlled separately and $m = 16$.

The grasping and finger-gaiting behaviors share the same overall structure: a joint space PBDS damping task, a family of pullback ECBF safety tasks, and task manifold action tasks driven by a higher-level planner. The safety tasks comprise (i) joint limit ECBFs; (ii) force closure ECBFs based on the l^* metric [20] for one four-finger-grasp variant and four per-finger-excluded variants; (iii) fingertip lift-distance ECBFs that keep each non-grasping fingertip away from the object surface; (iv) pairwise finger-finger distance ECBFs that prevent

fingertip collisions; (v) per-fingertip table avoidance ECBFs; and (vi) link-object distance ECBFs that keep non-fingertip links clear of the object (per-link CBFs for grasping; per-finger distal phalanx ECBFs for in-hand reorientation). The action tasks expose a per-finger controller on the 1D fingertip-to-object distance. For grasping, additional controllers on the 3D palm position and on the 3D average fingertip position. For in-hand reorientation, an additional action task drives the gaited fingertip's 3D position toward a planner-supplied target in the frame of the contact plane spanned by the three in-contact fingertips. In total, 84 ECBF tasks are instantiated for the grasping policy and 61 for the in-hand reorientation policy, a subset of which is active at each control step.

A finite-state machine modulates task weights to coordinate the grasp and gaiting phases. It selects the active force closure variant as fingers enter or leave contact, and toggles the per-finger lift-distance CBFs depending on whether each finger is stationary or in transit. The full enumeration of task maps, safety functions, and ECBF gains is given in Appendix D.2.

C. Dexterous Grasping

We apply SafePBDS to autonomous grasping of diverse household objects.

1) *Scene Setup and Test Protocol*: We select 20 household objects (Fig. 4) with mass ranging from 64 to 613 g and longest bounding box dimension ranging from roughly 6 to 29 cm. At the start of each trial, FoundationPose reports the object's 6-DOF pose, and a wrist pose sampler proposes top-down wrist pose candidates that are filtered for reachability, clearance, and aperture fit. The top-ranked candidate is executed in two stages: first, a Deoxys-controlled approach to the wrist pose (via a 15 cm waypoint above the pose followed by a direct descent); second, SafePBDS takes over the full 23-DOF arm-hand system and runs the PBDS QP at each control step. During the SafePBDS phase, two action tasks are composed: per-finger 1D fingertip-to-object distance controllers that drive each fingertip onto the object surface, and a 3D average-fingertip centering controller that aligns the four-fingertip centroid with the object's geometric center, for a total of $4 \times 1 + 3 = 7$ action dimensions. The force closure ECBF (Section VIII-B) is activated as the fingertips approach the object, so that the final squeeze converges to a certified force-closed grasp. A trial is recorded as a *success* if the object is successfully lifted with all four fingers in contact, a *partial success* if it is stably lifted with one finger not in contact, and a *failure* otherwise. The candidate filtering pipeline, action gain schedule, and full execution protocol are given in Appendix E.

To showcase the flexibility of our framework, we additionally evaluate grasping with one of the four fingers excluded. The active force closure ECBF is switched to the per-finger-excluded variant, so that l^* is computed over the three in-contact fingers, while the excluded finger is held clear of the object by retargeting its fingertip-to-object distance action to a fixed clearance and enabling its fingertip lift-distance ECBF. No other policy modification is required.

2) *Results*: Figure 7 summarizes the 4-finger grasping results (per-object trial counts in Table A2 of Appendix E).

Figure 5 shows representative 4-finger executions, and Figure 6 shows the 3-finger variants for each excluded finger.

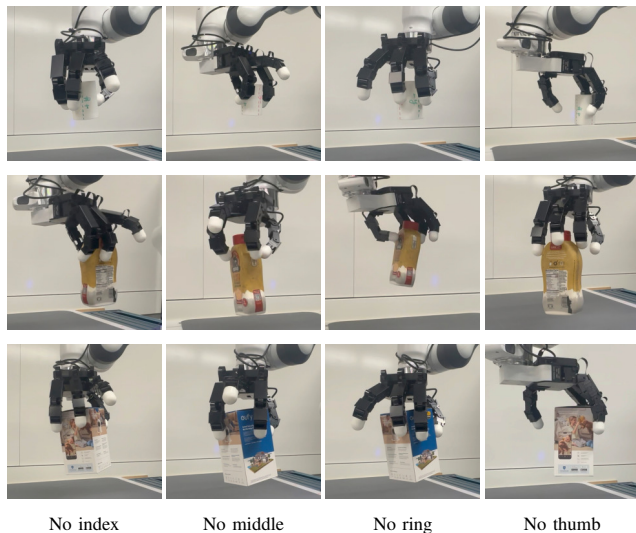


Fig. 6. 3-finger grasping variants for the three test objects. Each image is labeled with the excluded finger, which is held clear of the object while the remaining three fingers form a force closure grasp under SafePBDS. Across the three objects and four exclusions, tested at three table locations each (36 trials), SafePBDS achieves 34/36 (94.4%); both failures occur on the wide object (bottom row) with the thumb excluded, where its width forces the remaining index–ring pinch to span near the limits of the hand’s reachable workspace. Per-object counts are listed in Appendix Table A3.

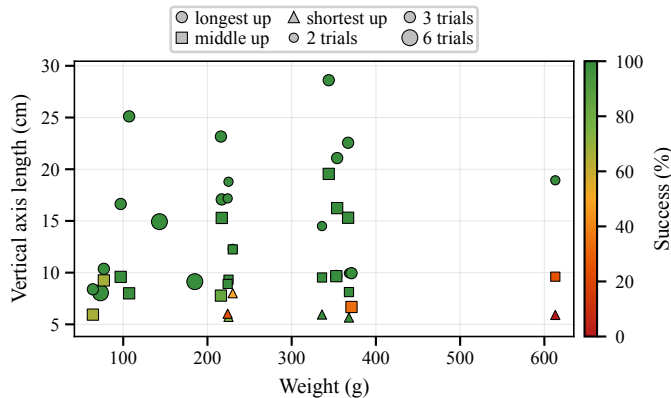


Fig. 7. Per-(object, pose) grasp outcomes plotted against object weight and the vertical bounding-box length at the tested pose. Marker color encodes the per-group success score (success = 1, partial = 0.5, failure = 0, averaged over the 2–6 trials in each group, shown on a 0–100% scale); shape encodes which bounding-box axis is oriented upward (longest, middle, or shortest); and size is proportional to trial count.

SafePBDS achieves an overall success rate of 111/120 (92.5%) on the 20 household objects, with 15 of 20 attaining full 6/6 success. Figure 7 plots each (object, pose) trial group against object weight and the vertical bounding-box length at that pose. Partial successes and failures concentrate among objects with small vertical extents (roughly ≤ 10 cm) and high weight. Low vertical extent is less forgiving of pose estimation errors and leaves less table clearance, while high weight requires grasp forces that approach the limits of the low-level impedance controller and the hand’s hardware capabilities.

The 3-finger ablation (Figure 6) succeeds in 34/36 trials (94.4%); thanks to the autonomous PBDS behavior, a high-level decision policy can reliably choose which finger to exclude despite the complexity of dexterous precision grasping.

D. Palm-Down In-Hand Reorientation (IHR)

We apply SafePBDS to palm-down IHR, where the goal is to rotate a grasped object about the palm normal axis with the palm normal facing down. This is significantly more challenging than the more common “IHR with palm facing up” in literature (e.g. [28, 14, 33, 57]); with the palm facing down, the object must be securely grasped throughout the process. To tackle the combinatorial complexity of finger gaiting sequences, we leverage offline planning in simulation to discover feasible reorientation trajectories, which are then deployed on hardware.

1) *Simulation Pre-Planning*: The main challenge in IHR is choosing which finger to move and where to move it. We address this with a depth-first tree search over finger-object contact states, where each node has all four fingers in contact and each edge corresponds to relocating one finger. The search tree is rooted at an initial grasp obtained by reusing the top-down grasping pipeline of Section VIII-C on a 6 cm diameter bottle. A priority queue at each depth orders nodes by the cumulative z -axis yaw rotation achieved along the path from the root, while a balance constraint caps the disparity in per-finger move counts so that no single finger is overused.

For each movable finger at a node, 12 candidate extensions are generated by a predetermined grid of steps along the object surface. Fingers are not allowed to move consecutively and the max gaiting step count difference across fingers is 1. This leaves at most 3 movable fingers and a maximum branching factor of 36. To prevent the search from getting stuck with a never-movable finger, a cyclic sequence of “next movable” fingers is assigned so that at each tree depth, the “next movable” finger must be movable after the candidate extension. Each candidate is forward-simulated under SafePBDS and a four-phase (LIFTING, TRAVERSING, DROPPING, ADJUSTING) primitive, and an accepted candidate becomes a tree edge. Throughout the primitive, the three-finger grasp is maintained by the corresponding force closure ECBF and pairwise fingertip spacing ECBFs (the object itself may still move). In TRAVERSING and DROPPING, the force closure ECBF excluding the next movable finger is additionally activated. A candidate is accepted when, on reaching ADJUSTING, the moving finger is aligned with its target contact, the hand is at rest, all four fingers are back in contact, the force closure ECBF excluding the next movable finger is non-negative, and the object tilt from the rotation axis is below 10° . Loss of contact at any in-contact finger during the rollout aborts execution and rejects the candidate. A simplified illustration is shown in Figure 1. More details are given in Appendix F. Figure 8 shows the resulting search trees.

2) *Scene Setup and Test Protocol*: A human operator places the object in between the finger cage, and the fingers grasp the object using the initial grasping configuration in the motion plan. The joint angles from the planned finger gaiting sequence are then played back open loop. To evaluate robustness of our plans, we test each plan under three conditions: (i) with the arm held static and no added payload, (ii) with the arm held static and up to 98 grams of weight incrementally added into the bottle, and (iii) with the arm swinging and 70 grams of loose weight inside the bottle.

3) *Results*: Figure 9 illustrates examples of the task execution. Using the same open loop motion plan across all three conditions, SafePBDS achieves over 360° rotation in both directions. We attribute this robustness to the fact that our IHR plan is certifiably in force closure. This demonstrates that SafePBDS offloads the burden on a higher-level planner (in this case the tree search) and allows a simple strategy to achieve highly constrained and difficult manipulation tasks.

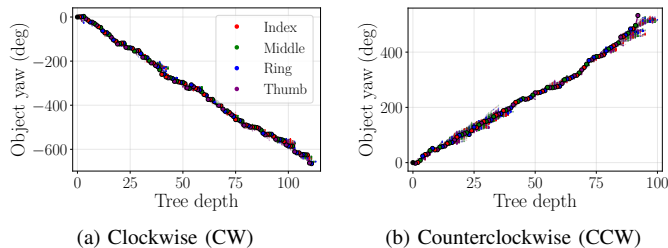
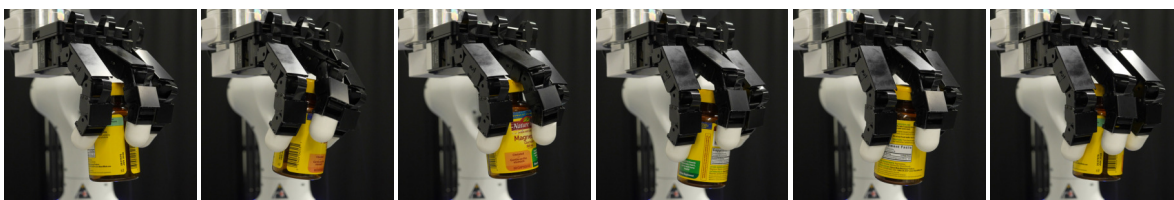


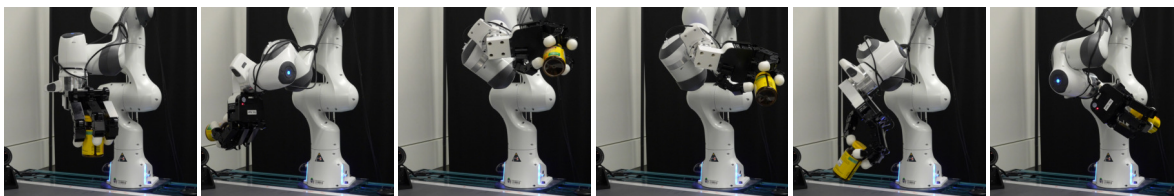
Fig. 8. Offline IHR motion plan trees for (a) clockwise and (b) counterclockwise target rotations. Each node represents a finger gaiting state, and the highlighted path from the root achieves the largest cumulative yaw.

IX. DISCUSSION

This paper presents SafePBDS (Safe Pullback Bundle Dynamical Systems), a geometric motion generation framework that extends PBDS [6] with task manifold safety guarantees and an action interface. At each control step, two convex QPs are solved sequentially: the first defines the autonomous safe acceleration, and the second injects the action input around it. The resulting configuration space acceleration is certifiably safe and recovers the autonomous task dynamics when the action input vanishes. We validate its theoretical properties in simulation on an S^2 double integrator and a 7-DOF Franka arm. On a 23-DOF Franka–Allegro hardware system, SafePBDS attains 92.5% grasp success across 20 household objects, and 94.4% for 3-finger grasps through simple action changes enabled by the action interface. Finally, SafePBDS enables the first model-based robust palm-down in-hand reorientation, producing over 360° rotation in both directions under varying object weight and arm motion.



(a) Clockwise rotation, static arm



(b) Counterclockwise rotation, swinging arm with weighted object

Fig. 9. In-hand reorientation on hardware. Snapshots are arranged temporally from left to right. (a) Clockwise rotation with the arm held static and empty bottle. (b) Counterclockwise rotation with the arm swinging and 2.5 oz loose weight in the bottle, demonstrating robustness under load and motion.

A. Limitations

Like other vector-field policies, SafePBDS solves a local motion generation problem: over long horizons, the system may become trapped in local minima, and escaping them must be handled by the higher-level policy that provides the action input. In addition, SafePBDS is kinematic and therefore requires a lower-level tracker to realize the commanded accelerations, or an impedance controller when compliant contact forces are needed. Finally, the pullback derivations assume surjective submersion task maps and a fully or over-actuated robot; underactuated systems are therefore outside the scope of the present framework.

B. Future work

On the application side, an important direction is to combine the action interface enabled by SafePBDS with more advanced decision-making policies, such as reinforcement learning or vision-language-action models. On the theoretical side, relaxing the Assumption 1 will allow SafePBDS to handle a broader class of objectives, including tasks with singularities or nonsmooth task maps. In addition, incorporating robot dynamics and the contact forces required for manipulation, rather than relying on an acceleration tracker or impedance controller, will allow SafePBDS to reason directly about interaction forces alongside geometric safety.

REFERENCES

- [1] *Allegro Hand Linux Project*. https://github.com/simlabrobotics/allegro_hand_linux_v4.
- [2] Aaron D. Ames et al. “Control Barrier Functions: Theory and Applications”. In: *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 3420–3431.
- [3] Aditya Bhatt et al. “Surprisingly Robust In-Hand Manipulation: An Empirical Study”. In: *Proceedings of Robotics: Science and Systems (RSS)*. 2021. DOI: 10.15607/RSS.2021.XVII.089.
- [4] Jeannette Bohg et al. “Data-Driven Grasp Synthesis—A Survey”. In: *IEEE Transactions on Robotics* 30.2 (2014), pp. 289–309.

- [5] Francesco Bullo and Andrew D. Lewis. *Geometric Control of Mechanical Systems*. Vol. 49. Texts in Applied Mathematics. New York-Heidelberg-Berlin: Springer Verlag, 2004. ISBN: 0-387-22195-6.
- [6] Andrew Bylard, Riccardo Bonalli, and Marco Pavone. “Composable Geometric Motion Policies using Multi-Task Pullback Bundle Dynamical Systems”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 7464–7470.
- [7] Tao Chen, Jie Xu, and Pulkit Agrawal. “A System for General In-Hand Object Re-Orientation”. In: *Proceedings of the 5th Conference on Robot Learning (CoRL)*. Vol. 164. Proceedings of Machine Learning Research. PMLR, 2022, pp. 297–307.
- [8] Tao Chen et al. “Visual Dexterity: In-Hand Reorientation of Novel and Complex Object Shapes”. In: *Science Robotics* 8.84 (2023), eadc9244.
- [9] Ching-An Cheng et al. “RMPflow: A Geometric Framework for Generation of Multitask Motion Policies”. In: *IEEE Transactions on Automation Science and Engineering* 18.3 (2021), pp. 968–987.
- [10] Cheng Chi et al. “Diffusion Policy: Visuomotor Policy Learning via Action Diffusion”. In: *The International Journal of Robotics Research* 44.10-11 (2025), pp. 1684–1704.
- [11] James Eells and Luc Lemaire. “A Report on Harmonic Maps”. In: *Bulletin of the London Mathematical Society* 10.1 (1978), pp. 1–68.
- [12] Thomas Feix et al. “The GRASP Taxonomy of Human Grasp Types”. In: *IEEE Transactions on Human-Machine Systems* 46.1 (2016), pp. 66–77.
- [13] Gemini Robotics Team, Google DeepMind. *Gemini Robotics 1.5: Pushing the Frontier of Generalist Robots with Advanced Embodied Reasoning, Thinking, and Motion Transfer*. 2025. arXiv: 2510.03342 [cs.LG].
- [14] Ankur Handa et al. “DeXtreme: Transfer of Agile In-Hand Manipulation from Simulation to Reality”. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. 2023, pp. 5977–5984.
- [15] Oussama Khatib. “A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation”. In: *IEEE Journal on Robotics and Automation* 3.1 (1987), pp. 43–53.
- [16] KIRI Innovations. *KIRI Engine: 3D Scanner App*. <https://www.kiriengine.app/>. Photogrammetry and LiDAR-based 3D scanning application. 2024.
- [17] Alexander Kirillov et al. “Segment Anything”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2023, pp. 3992–4003.
- [18] John M. Lee. *Introduction to Smooth Manifolds*. 2nd ed. Vol. 218. Graduate Texts in Mathematics. Springer, 2012.
- [19] Hrishit Leen et al. *Deep Learning for Dexterous Robot Grasping*. Survey, Georgia Institute of Technology. 2025.
- [20] Albert H. Li et al. “FRoGGeR: Fast Robust Grasp Generation via the Min-Weight Metric”. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2023, pp. 6809–6816.
- [21] Jie Liu, Hwa Jen Yap, and Anis Salwa Mohd Khairuddin. “Review on Motion Planning of Robotic Manipulator in Dynamic Environments”. In: *Journal of Sensors* 2024 (2024), p. 5969512.
- [22] Tengyu Liu et al. “Synthesizing Diverse and Physically Stable Grasps With Arbitrary Hand Structures Using Differentiable Force Closure Estimator”. In: *IEEE Robotics and Automation Letters* 7.1 (2022), pp. 470–477.
- [23] Tyler Ga Wei Lum et al. “DextrAH-G: Pixels-to-Action Dexterous Arm-Hand Grasping with Geometric Fabrics”. In: *Proceedings of The 8th Conference on Robot Learning (CoRL)*. Vol. 270. Proceedings of Machine Learning Research. PMLR, 2025, pp. 3182–3211.
- [24] Andrew S. Morgan et al. “Complex In-Hand Manipulation via Compliance-Enabled Finger Gaiting and Multi-Modal Planning”. In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 4821–4828.
- [25] Richard M. Murray, Zexiang Li, and S. Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.
- [26] Rhys Newbury et al. “Deep Learning Approaches to Grasp Synthesis: A Review”. In: *IEEE Transactions on Robotics* 39.5 (2023), pp. 3994–4015.
- [27] Quan Nguyen and Koushil Sreenath. “Exponential Control Barrier Functions for enforcing high relative-degree safety-critical constraints”. In: *2016 American Control Conference (ACC)*. IEEE. 2016, pp. 322–328.
- [28] OpenAI et al. “Learning Dexterous In-Hand Manipulation”. In: *The International Journal of Robotics Research* 39.1 (2020), pp. 3–20.
- [29] Tao Pang et al. “Global Planning for Contact-Rich Manipulation via Local Smoothing of Quasi-Dynamic Contact Models”. In: *IEEE Transactions on Robotics* 39.6 (2023), pp. 4691–4711.
- [30] Physical Intelligence. $\pi_{0.7}$: *A Steerable Generalist Robotic Foundation Model with Emergent Capabilities*. 2026. arXiv: 2604.15483 [cs.LG].
- [31] Johannes Pitz et al. “Learning a Shape-Conditioned Agent for Purely Tactile In-Hand Manipulation of Various Objects”. In: *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2024, pp. 13112–13119.
- [32] Michael Posa, Cecilia Cantu, and Russ Tedrake. “A direct method for trajectory optimization of rigid bodies through contact”. In: *The International Journal of Robotics Research* 33.1 (2014), pp. 69–81.
- [33] Haozhi Qi et al. “General In-Hand Object Rotation with Vision and Touch”. In: *Proceedings of The 7th Conference on Robot Learning (CoRL)*. Vol. 229. Proceedings of Machine Learning Research. PMLR, 2023, pp. 2549–2564.
- [34] Nathan D Ratliff et al. “Riemannian motion policies”. In: *arXiv preprint arXiv:1801.02854* (2018).
- [35] Nathan D. Ratliff et al. “Optimization Fabrics”. In: *arXiv preprint arXiv:2008.02399* (2020).

- [36] Massimiliano de Sa, Pio Ong, and Aaron D. Ames. “From Bundles to Backstepping: Geometric Control Barrier Functions for Safety-Critical Control on Manifolds”. In: *arXiv preprint arXiv:2510.20202* (2025).
- [37] Luis Sentis and Oussama Khatib. “Synthesis of Whole-Body Behaviors through Hierarchical Control of Behavioral Primitives”. In: *International Journal of Humanoid Robotics* 2.4 (2005), pp. 505–518.
- [38] Wenceslao Shaw Cortez et al. “Control Barrier Functions for Mechanical Systems: Theory and Application to Robotic Grasping”. In: *IEEE Transactions on Control Systems Technology* 29.2 (2021), pp. 530–545.
- [39] Yuki Shirai et al. “Robust pivoting: Exploiting frictional stability using bilevel optimization”. In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 992–998.
- [40] Bruno Siciliano and Jean-Jacques E. Slotine. “A General Framework for Managing Multiple Tasks in Highly Redundant Robotic Systems”. In: *Proceedings of the 5th International Conference on Advanced Robotics (ICAR)*. 1991, pp. 1211–1216.
- [41] Leon Sievers, Johannes Pitz, and Berthold Bäuml. “Learning Purely Tactile In-Hand Manipulation with a Torque-Controlled Hand”. In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 2745–2751.
- [42] H. J. Terry Suh et al. “Dexterous Contact-Rich Manipulation via the Contact Trust Region”. In: *The International Journal of Robotics Research* (2026). OnlineFirst. DOI: 10.1177/02783649251398875.
- [43] Markku Suomalainen, Yiannis Karayiannidis, and Ville Kyrki. “A survey of robot manipulation in contact”. In: *Robotics and Autonomous Systems* 156 (2022), p. 104224.
- [44] Chen Tang et al. “Deep Reinforcement Learning for Robotics: A Survey of Real-World Successes”. In: *Annual Review of Control, Robotics, and Autonomous Systems* 8 (2025), pp. 153–188.
- [45] Andrew J. Taylor et al. “Safe Backstepping with Control Barrier Functions”. In: *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE. 2022, pp. 5775–5782.
- [46] Emanuel Todorov, Tom Erez, and Yuval Tassa. “MuJoCo: A physics engine for model-based control”. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2012, pp. 5026–5033.
- [47] TRI LBM Team et al. “A Careful Examination of Large Behavior Models for Multitask Dexterous Manipulation”. In: (2025). arXiv: 2507.05331 [cs.LG].
- [48] Toshiaki Tsuji et al. “A Survey on Imitation Learning for Contact-Rich Tasks in Robotics”. In: *The International Journal of Robotics Research* (2026). OnlineFirst. DOI: 10.1177/02783649261417694.
- [49] Dylan Turpin et al. “Fast-Grasp’D: Dexterous Multi-finger Grasp Generation Through Differentiable Simulation”. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2023, pp. 8082–8089.
- [50] Karl Van Wyk et al. “Geometric Fabrics: Generalizing Classical Mechanics to Capture the Physics of Behavior”. In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 3202–3209.
- [51] Abraham Itzhak Weinberg et al. “Survey of Learning-Based Approaches for Robotic In-Hand Manipulation”. In: *Frontiers in Robotics and AI* 11 (2024), p. 1455431.
- [52] Bowen Wen et al. “FoundationPose: Unified 6D Pose Estimation and Tracking of Novel Objects”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024, pp. 17868–17879.
- [53] Albert Wu, Michelle Guo, and C. Karen Liu. “Learning Diverse and Physically Feasible Dexterous Grasps with Generative Model and Bilevel Optimization”. In: *Proceedings of The 6th Conference on Robot Learning (CoRL)*. Vol. 205. Proceedings of Machine Learning Research. PMLR, 2023, pp. 1938–1948.
- [54] Guofan Wu and Koushil Sreenath. “Safety-critical and constrained geometric control synthesis using control Lyapunov and control Barrier functions for systems evolving on manifolds”. In: *2015 American Control Conference (ACC)*. IEEE. 2015, pp. 2038–2044.
- [55] Wei Xiao and Calin Belta. “High-Order Control Barrier Functions”. In: *IEEE Transactions on Automatic Control* 67.7 (2022), pp. 3655–3662.
- [56] Max Yang et al. “AnyRotate: Gravity-Invariant In-Hand Object Rotation with Sim-to-Real Touch”. In: *Proceedings of The 8th Conference on Robot Learning (CoRL)*. Vol. 270. Proceedings of Machine Learning Research. PMLR, 2025, pp. 4727–4747.
- [57] Zhao-Heng Yin et al. “Rotating without Seeing: Towards In-hand Dexterity through Touch”. In: *Proceedings of Robotics: Science and Systems (RSS)*. 2023. DOI: 10.15607/RSS.2023.XIX.036.
- [58] Tony Z. Zhao et al. “Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware”. In: *Proceedings of Robotics: Science and Systems (RSS)*. 2023. DOI: 10.15607/RSS.2023.XIX.016.
- [59] Zhigen Zhao et al. “A Survey of Optimization-Based Task and Motion Planning: From Classical to Learning Approaches”. In: *IEEE/ASME Transactions on Mechatronics* (2024). Early Access. DOI: 10.1109/TMECH.2024.3452509.
- [60] Yifeng Zhu et al. “VIOLA: Imitation Learning for Vision-Based Manipulation with Object Proposal Priors”. In: *Proceedings of The 6th Conference on Robot Learning (CoRL)*. Vol. 205. Proceedings of Machine Learning Research. PMLR, 2023, pp. 1199–1210.
- [61] Simon Zimmermann et al. “A multi-level optimization framework for simultaneous grasping and motion planning”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 2966–2972.

APPENDIX A
PROOF OF LEMMA IV.2

Lemma IV.2. *In the task manifold PBDS setup, the BCBF candidate h given in (25) has time derivative*

$$\begin{aligned} \dot{h} &= \langle \text{grad } h_0, \dot{x} \rangle_N + \varepsilon \langle e_x, \nabla_{\dot{x}} \tilde{\xi} \rangle_N \\ &\quad - \varepsilon \langle e_x, df(M\nabla_{\dot{\sigma}} \dot{\sigma}) + (\nabla df)(\dot{\sigma}, \dot{\sigma}) \rangle_N. \end{aligned}$$

Proof. Write $e^\alpha = \dot{x}^\alpha - \tilde{\xi}^\alpha$ for the error components. The coordinate expression for the differential of h at $(x, \dot{x}) \in TN$, acting on $(a_x, a_{\dot{x}}) \in T_{(x, \dot{x})}(TN)$, is

$$\begin{aligned} dh \cdot (a_x, a_{\dot{x}}) &= \frac{\partial h_0}{\partial x^\gamma} a_x^\gamma \\ &\quad + \varepsilon g_{\alpha\beta} e^\beta \left(\frac{\partial \tilde{\xi}^\alpha}{\partial x^\gamma} a_x^\gamma - a_{\dot{x}}^\alpha \right) - \frac{\varepsilon}{2} \frac{\partial g_{\alpha\beta}}{\partial x^\gamma} e^\alpha e^\beta a_x^\gamma. \quad (50) \end{aligned}$$

Step 1: Substitute the dynamics.: Along a trajectory, $a_x^\gamma = \dot{x}^\gamma$ and $a_{\dot{x}}^\alpha = \ddot{x}^\alpha$. From (31), the coordinate acceleration on N is $\ddot{x}^\alpha = ({}^N\nabla_{\dot{x}} \dot{x})^\alpha - N\Gamma_{\gamma\delta}^\alpha \dot{x}^\gamma \dot{x}^\delta$. Using the geometric kinematics (35), we substitute

$$\ddot{x}^\alpha = (df(M\nabla_{\dot{\sigma}} \dot{\sigma}))^\alpha + ((\nabla df)(\dot{\sigma}, \dot{\sigma}))^\alpha - N\Gamma_{\gamma\delta}^\alpha \dot{x}^\gamma \dot{x}^\delta.$$

The parenthesized term in (50) becomes

$$\begin{aligned} \frac{\partial \tilde{\xi}^\alpha}{\partial x^\gamma} \dot{x}^\gamma - \ddot{x}^\alpha &= \frac{\partial \tilde{\xi}^\alpha}{\partial x^\gamma} \dot{x}^\gamma + N\Gamma_{\gamma\delta}^\alpha \dot{x}^\gamma \dot{x}^\delta \\ &\quad - (df(M\nabla_{\dot{\sigma}} \dot{\sigma}))^\alpha - ((\nabla df)(\dot{\sigma}, \dot{\sigma}))^\alpha, \quad (51) \end{aligned}$$

so (50) yields

$$\begin{aligned} \dot{h} &= \frac{\partial h_0}{\partial x^\gamma} \dot{x}^\gamma + \varepsilon g_{\alpha\beta} e^\beta \left(\frac{\partial \tilde{\xi}^\alpha}{\partial x^\gamma} \dot{x}^\gamma + N\Gamma_{\gamma\delta}^\alpha \dot{x}^\gamma \dot{x}^\delta \right. \\ &\quad \left. - (df(M\nabla_{\dot{\sigma}} \dot{\sigma}))^\alpha - ((\nabla df)(\dot{\sigma}, \dot{\sigma}))^\alpha \right) \\ &\quad - \frac{\varepsilon}{2} \frac{\partial g_{\alpha\beta}}{\partial x^\gamma} e^\alpha e^\beta \dot{x}^\gamma. \quad (52) \end{aligned}$$

Step 2: Decompose $\dot{x} = e + \tilde{\xi}$.: Split the second index of $N\Gamma_{\gamma\delta}^\alpha \dot{x}^\gamma \dot{x}^\delta$ using $\dot{x}^\delta = e^\delta + \tilde{\xi}^\delta$:

$$N\Gamma_{\gamma\delta}^\alpha \dot{x}^\gamma \dot{x}^\delta = N\Gamma_{\gamma\delta}^\alpha \dot{x}^\gamma \tilde{\xi}^\delta + N\Gamma_{\gamma\delta}^\alpha \dot{x}^\gamma e^\delta.$$

The first two terms inside the parentheses of (52) combine with $N\Gamma_{\gamma\delta}^\alpha \dot{x}^\gamma \tilde{\xi}^\delta$ to form the covariant derivative of $\tilde{\xi}$ in the direction \dot{x} :

$$(\nabla_{\dot{x}} \tilde{\xi})^\alpha = \frac{\partial \tilde{\xi}^\alpha}{\partial x^\gamma} \dot{x}^\gamma + N\Gamma_{\gamma\delta}^\alpha \dot{x}^\gamma \tilde{\xi}^\delta.$$

This leaves a remainder $N\Gamma_{\gamma\delta}^\alpha \dot{x}^\gamma e^\delta$, so (52) becomes

$$\begin{aligned} \dot{h} &= \frac{\partial h_0}{\partial x^\gamma} \dot{x}^\gamma + \varepsilon g_{\alpha\beta} e^\beta \left((\nabla_{\dot{x}} \tilde{\xi})^\alpha + N\Gamma_{\gamma\delta}^\alpha \dot{x}^\gamma e^\delta \right. \\ &\quad \left. - (df(M\nabla_{\dot{\sigma}} \dot{\sigma}))^\alpha - ((\nabla df)(\dot{\sigma}, \dot{\sigma}))^\alpha \right) \\ &\quad - \frac{\varepsilon}{2} \frac{\partial g_{\alpha\beta}}{\partial x^\gamma} e^\alpha e^\beta \dot{x}^\gamma. \quad (53) \end{aligned}$$

Step 3: Cancellation via metric compatibility.: It remains to show that the two terms involving $e^\alpha e^\beta \dot{x}^\gamma$ in (53) cancel. By metric compatibility [4, Prop. 5.5(c)],

$$\frac{\partial g_{\alpha\beta}}{\partial x^\gamma} = g_{\delta\beta} N\Gamma_{\gamma\alpha}^\delta + g_{\alpha\delta} N\Gamma_{\gamma\beta}^\delta.$$

Contracting against $e^\alpha e^\beta \dot{x}^\gamma$ and using the symmetry of $e^\alpha e^\beta$ in $\alpha \leftrightarrow \beta$,

$$\frac{\partial g_{\alpha\beta}}{\partial x^\gamma} e^\alpha e^\beta \dot{x}^\gamma = 2 g_{\delta\beta} N\Gamma_{\gamma\alpha}^\delta e^\alpha e^\beta \dot{x}^\gamma,$$

so the ∂g term evaluates to

$$-\frac{\varepsilon}{2} \frac{\partial g_{\alpha\beta}}{\partial x^\gamma} e^\alpha e^\beta \dot{x}^\gamma = -\varepsilon g_{\delta\beta} N\Gamma_{\gamma\alpha}^\delta e^\alpha e^\beta \dot{x}^\gamma.$$

Meanwhile, relabeling $\alpha \leftrightarrow \delta$ in the remainder term from (53) gives

$$\varepsilon g_{\alpha\beta} N\Gamma_{\gamma\delta}^\alpha \dot{x}^\gamma e^\delta e^\beta = \varepsilon g_{\delta\beta} N\Gamma_{\gamma\alpha}^\delta \dot{x}^\gamma e^\alpha e^\beta.$$

These two expressions are equal and opposite, so they cancel.

Step 4: Collect.: After cancellation, (53) reduces to

$$\begin{aligned} \dot{h} &= \frac{\partial h_0}{\partial x^\gamma} \dot{x}^\gamma + \varepsilon g_{\alpha\beta} e^\beta \left((\nabla_{\dot{x}} \tilde{\xi})^\alpha \right. \\ &\quad \left. - (df(M\nabla_{\dot{\sigma}} \dot{\sigma}))^\alpha - ((\nabla df)(\dot{\sigma}, \dot{\sigma}))^\alpha \right), \end{aligned}$$

which, upon identifying $\frac{\partial h_0}{\partial x^\gamma} \dot{x}^\gamma = \langle \text{grad } h_0, \dot{x} \rangle_N$ and $g_{\alpha\beta} e^\beta (\cdot)^\alpha = \langle e_x, \cdot \rangle_N$, is (45). \square

APPENDIX B
EVALUATION: SIMULATED \mathbb{S}^2 DOUBLE INTEGRATOR

Using a point robot on \mathbb{S}^2 , we first validate the theoretical properties of SafePBDS, in particular chart invariance, safety guarantees, multi-task composition, and effects of our action space design.

1 System Setup

a) System definition.: Consider a point robot with unit mass traveling on the surface of a unit sphere. The configuration manifold of the robot is \mathbb{S}^2 , which has a natural embedding $\bar{\varphi} : \mathbb{S}^2 \hookrightarrow \mathbb{R}^3$. We consider the atlas $\{(U_N, \varphi_N), (U_S, \varphi_S)\}$ formed by the north and south pole stereographic projection charts, where:

- $U_N = \mathbb{S}^2 \setminus \{(0, 0, 1)\}$
- $\varphi_N : U_N \rightarrow \mathbb{R}^2$
- $U_S = \mathbb{S}^2 \setminus \{(0, 0, -1)\}$
- $\varphi_S : U_S \rightarrow \mathbb{R}^2$

We can define corresponding maps $\bar{\varphi}_N : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ and $\bar{\varphi}_S : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ from chart coordinates (y_1, y_2) to embedding coordinates (x_1, x_2, x_3) :

$$\bar{\varphi}_N(y_1, y_2) = \left(\frac{2y_1}{\|y\|^2 + 1}, \frac{2y_2}{\|y\|^2 + 1}, \frac{\|y\|^2 - 1}{\|y\|^2 + 1} \right), \quad (54)$$

where $\|y\|^2 = y_1^2 + y_2^2$. The south pole chart $\varphi_S : U_S \rightarrow \mathbb{R}^2$ projects from $S = (0, 0, -1)$:

$$\varphi_S(x_1, x_2, x_3) = \left(\frac{x_1}{1 + x_3}, \frac{x_2}{1 + x_3} \right), \quad (55)$$

with inverse

$$\bar{\varphi}_S(y_1, y_2) = \left(\frac{2y_1}{\|y\|^2 + 1}, \frac{2y_2}{\|y\|^2 + 1}, \frac{1 - \|y\|^2}{\|y\|^2 + 1} \right). \quad (56)$$

The metric in both charts is given by $g_{ij} = \frac{4}{(1 + \|y\|^2)^2} \delta_{ij}$, with Christoffel symbols

$$\Gamma_{ij}^k = \frac{-2}{1 + \|y\|^2} (y_i \delta_{jk} + y_j \delta_{ik} - y_k \delta_{ij}). \quad (57)$$

b) *Configuration Dynamics*: We compare two configuration manifold dynamics, both implemented with fourth-order Runge–Kutta integration schemes:

- 1) *Geometric*: $\ddot{y}^k + \Gamma_{ij}^k \dot{y}^i \dot{y}^j = F^k$, using the Christoffel symbols of the round metric.
- 2) *Flat*: $\ddot{y}^k = F^k$, ignoring the manifold geometry and treating chart coordinates as \mathbb{R}^2 with $\Gamma = 0$ and $g = \delta_{ij}$.

c) *Task Dynamics*: Beyond the PBDS with embedding task used throughout the main text, the chart invariance experiment in this appendix uses two additional task dynamics variants. We consider three classes of dynamics on \mathbb{S}^2 in total:

- 1) *Direct SMCS* (appendix only): a geodesic attractor potential $\Phi(y) = \frac{1}{2} d_{\mathbb{S}^2}(y, y_{\text{goal}})^2$ with Riemannian dissipation $\mathcal{F}_D = -b g \dot{y}$ defines a single SMCS directly on the configuration manifold (no PBDS).
- 2) *Trivial PBDS* (appendix only): we define the task map $f = \bar{\varphi}_\alpha: \mathbb{R}^2 \rightarrow \mathbb{R}^3$ (chart-to-embedding) with *zero* potential and *zero* dissipative force on the embedding. The resulting PBDS task SMCS has no task manifold forces; however, the PBDS closed-form (11) still produces a nonzero coordinate acceleration $\ddot{\sigma}$ because the Hessian of the task map injects the Christoffel-symbol correction. This isolates the geometric contribution of the task map from any task manifold forcing.
- 3) *PBDS with embedding task* (used in main text): following [1], we define the task map $f = \bar{\varphi}_\alpha: \mathbb{R}^2 \rightarrow \mathbb{R}^3$ (chart-to-embedding) and design an attractor potential and dissipative force on the embedding \mathbb{R}^3 . The PBDS closed-form (11) yields a coordinate acceleration $\ddot{\sigma}$.

d) *Safety constraint*: The robot must stay arclength r from a point $x_o \in \mathbb{S}^2$. Since arclength on the unit sphere equals the angle between embedding vectors, the safety function (Definition III.3) is

$$h_0(x) = \arccos(x \cdot x_o) - r, \quad x, x_o \in \mathbb{S}^2 \subset \mathbb{R}^3, \quad (58)$$

or, in chart coordinates, $h_0(y) = \arccos(\bar{\varphi}_\alpha(y) \cdot x_o) - r$ with $\alpha \in \{N, S\}$ the active chart. The safe set is $\mathcal{C}_0 = \{x \in \mathbb{S}^2 : h_0(x) \geq 0\}$. We enforce $h_0 \geq 0$ via the ECBF and BCBF formulations of Section IV; for BCBF, we set $\xi = 0$ and compare two metrics on the chart: the round metric $g_{ij} = \frac{4}{(1+\|y\|^2)^2} \delta_{ij}$ and the flat metric δ_{ij} .

e) *Shared experimental conditions*: All experiments start from a point on the equator of \mathbb{S}^2 (in the overlap of both charts, away from both poles). The chart invariance experiment (Section B.2) uses nonzero initial velocity so that the Christoffel terms are nonzero from the start. The pullback CBF and action interface experiments (Sections B.3–B.4) share the same scene: an obstacle of radius $r = 0.5$ rad centered on the geodesic midpoint between start and goal, with zero initial velocity.

2 Chart Invariance of Geometric vs. Flat Configuration Dynamics

We first demonstrate that on a curved configuration manifold, ignoring the geometry produces chart-dependent and incorrect dynamics, but that the PBDS task map alone recovers the correct geometric behavior even with zero task manifold forces.

TABLE A1
CHART INVARIANCE RUN CONFIGURATIONS: FREE MOTION ($F = 0$), NONZERO INITIAL VELOCITY. N/S: NORTH/SOUTH POLE CHART.

| Run | Config dynamics | Task dynamics |
|-----|-----------------|---------------|
| (a) | Geometric (N) | Direct SMCS |
| (b) | Flat (N) | Direct SMCS |
| (c) | Geometric (S) | Direct SMCS |
| (d) | Flat (S) | Direct SMCS |
| (e) | Flat (N) | Trivial PBDS |
| (f) | Flat (S) | Trivial PBDS |

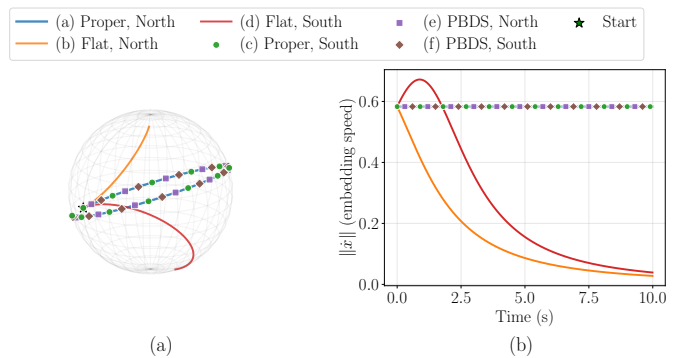


Fig. A1. Free motion on \mathbb{S}^2 . (a) Geometric (a,c) follow great circles and overlap; flat (b,d) diverge from the geodesic and from each other. Trivial PBDS (e,f) with zero task manifold forces and flat integration overlay the geometric trajectories, showing that the task map’s Hessian alone injects the Christoffel correction. Overlapping curves are drawn with staggered marker overlays so each is individually visible. (b) Geometric and trivial PBDS maintain constant embedding speed (geodesic property); flat does not.

The run configurations for the chart invariance experiments are listed in Table A1.

As shown in Figure A1, the geometric trajectories (a) and (c) follow great circles and agree on \mathbb{S}^2 up to numerical precision. The flat trajectories (b) and (d), which ignore the Christoffel symbols, diverge from each other and from the geodesic. Runs (e) and (f) use flat dynamics with the trivial PBDS: zero potential and zero dissipation on the embedding, so the only source of coordinate acceleration is the Hessian of the chart-to-embedding task map. Despite having flat configuration dynamics, the PBDS task dynamics provide the correct geometric correction, and both (e) and (f) match the geometric runs (a) and (c). (e) and (f) demonstrate that the task dynamics governs the PBDS behaviors, with no regard to the configuration manifold dynamics.

3 Autonomous SafePBDS on \mathbb{S}^2

We validate Autonomous SafePBDS (without task manifold actions) on the \mathbb{S}^2 double integrator to test for metric dependence, chart switching, and safety recovery. All experiments use the obstacle avoidance safety function (58).

a) *Metric dependence and chart switching*: The ECBF constraint involves only coordinate partial derivatives of h_0 and the coordinate acceleration, with no metric tensors or Christoffel symbols (Remark 7), making it metric-independent. The two BCBF runs produce visibly different avoidance trajectories, since the BCBF depends on the metric via $\text{grad } h_0$, $\|e_x\|_N$, and $\nabla_x \xi$; the BCBF also stays farther from the unsafe set boundary $h_0 = 0$ due to the required “safe velocity field.”

Additionally, we verify chart switching by varying the chart used for the configuration space integration. This is the

safety-augmented analogue of the chart switching experiment in [1, Fig. 5]. The task SMCS on \mathbb{R}^3 is unchanged. Run (iv) matches the fixed-chart reference (i) on \mathbb{S}^2 , confirming that the PBDS QP solution and safety constraint produce consistent embedding space behavior across chart transitions and that SafePBDS can be deployed on atlases with multiple charts.

b) *Recovery from outside the safe set.*: To test recovery, we run both an ECBF and a BCBF trajectory starting from inside the unsafe set. Both formulations drive the system out of the unsafe region and converge to the goal. However, the ECBF constrains only $h_0 \geq 0$ and recovers along the most direct path, while the BCBF’s backstepping barrier $h = h_0 - \frac{\epsilon}{2} \|\dot{x} - \tilde{\xi}\|_N^2$ additionally penalizes velocity that deviates from the safe velocity field $\tilde{\xi}$. This velocity-dependent margin causes the BCBF recovery to maintain a larger clearance from the obstacle boundary once $h_0 \geq 0$ is reached, consistent with the greater conservatism of the BCBF observed in the safe-start runs.

4 Steered SafePBDS on \mathbb{S}^2

We demonstrate the action interface (Section V) on the \mathbb{S}^2 double integrator. We use PBDS embedding dynamics with an ECBF safety constraint and an identity control task map $f_l = \text{id}: \mathbb{R}^2 \rightarrow \mathbb{R}^2$ on the active chart, so the action $u \in \mathbb{R}^2$ is a residual chart-space acceleration. All actions are defined geometrically in the embedding space and mapped to chart coordinates via the Jacobian pseudoinverse at each timestep, making them chart-invariant. We place the obstacle between the start and the goal, so that the autonomous system ($u = 0$) faces two valid avoidance paths on each side of the obstacle. We run five trajectories (viii)–(xii) (Table II), all shown in Figure 2(c):

- (viii) *Autonomous* ($u = 0$, gray): the system breaks symmetry via a small positional offset and converges to one side of the obstacle.
- (ix) $+u_\perp$ (blue): a tangential action perpendicular to the start–goal geodesic with $\|u_\perp\| = 1$, steering the robot to the same side as the autonomous bias.
- (x) $-u_\perp$ (orange): flipping the sign steers the robot to the opposite side, resolving the topological ambiguity that the autonomous dynamics alone cannot.
- (xi) u_{unsafe} (green): a large action ($\|u\| = 10$) continuously pointing toward the obstacle center. SafePBDS prevents the constraint violation, so the actual acceleration deviates from the command near the obstacle boundary. Safety is maintained ($h_0(t) \geq 0$) despite the adversarial action.
- (xii) $-u_\perp$ on the south pole chart (dashed purple): to verify chart invariance, we repeat (x) on the south pole chart. The overlay with the north pole chart run confirms that the steered QP solution is independent of chart choice.

APPENDIX C

EVALUATION: SIMULATED 7-DOF ROBOT ARM

1 System Setup

Consider a 7-DOF Franka Emika Panda arm with joint limits, simulated in MuJoCo [6]. In joint coordinates we

write $\sigma = (\sigma^1, \dots, \sigma^7)$, so the configuration manifold $M = \prod_{i=1}^7 (\sigma_-^i, \sigma_+^i)$ is an open subset of \mathbb{R}^7 , equipped with the flat product metric so that ${}^M\Gamma = 0$ in joint coordinates. The behavior comprises three PBDS tasks (items 1–3), two pullback CBF safety tasks (items 4–5), and a action task (item 6):

- 1) An *orientation attractor* task with scalar task map $f: M \rightarrow \mathbb{R}$, $f(\sigma) = \|\tilde{q}_{ee}(\sigma) - q_{\text{goal}}\|_2$, the chord distance in the \mathbb{R}^4 quaternion embedding between the goal and the end effector unit quaternion, with $\tilde{q}_{ee} = \text{sign}(q_{ee}^\top q_{\text{goal}}) q_{ee}$ resolving the antipodal $\pm q$ ambiguity in the universal double cover $\mathbb{S}^3 \rightarrow \text{SO}(3)$. The task uses the quadratic potential $\Phi(d) = d^2$ and the Euclidean metric on the scalar output; a companion damping task applies linear dissipation with identity map and Euclidean metric on the \mathbb{R}^4 quaternion.
- 2) A *position attractor* task (when active) with task map $f: M \rightarrow \mathbb{R}^3$, with a quadratic potential attracting toward a goal position, with the Euclidean metric; a companion damping task applies linear dissipation with identity map and Euclidean metric on \mathbb{R}^3 .
- 3) A *joint space damping* task with $f = \text{id}: M \rightarrow \mathbb{R}^7$.
- 4) *Joint limit ECBF* tasks with task maps $f_k: M \rightarrow \mathbb{R}$, $k = 1, \dots, 7$, given by coordinate projections, each with a safety function $h_{0,k}$ enforced via the pullback ECBF (Section IV).
- 5) *Workspace obstacle ECBF* tasks (one per collision body) with per-body safety functions $h_{0,j}(\sigma) = d_{\text{geom}}(g_j, g_{\text{obs}})$, where d_{geom} is the signed distance between a robot link collision body g_j and the obstacle sphere g_{obs} , computed via MuJoCo’s `mj_geomDistance`.
- 6) A *steered action* with scalar control task map $f: M \rightarrow \mathbb{R}$: $\sigma \mapsto \sigma^1$, projecting onto joint 1, so the action $u \in \mathbb{R}$ is a residual joint 1 acceleration.

All experiments use the same shared scenario unless otherwise noted: the arm starts at its home configuration and must reach a goal end effector orientation obtained from the home orientation by an XYZ Euler rotation of $(45^\circ, 60^\circ, 90^\circ)$, with a spherical obstacle (radius 0.10 m) placed in the sweep path of links 4–6. Each run proceeds until the system converges (task error and velocity below tolerance) or reaches a maximum duration of 15 s (20 s for the run with action input).

APPENDIX D

HARDWARE SETUP AND TASK SPECIFICATION DETAILS

This appendix expands on the hardware setup (Section VIII-A) and the manifold/task specification (Section VIII-B) used for the dexterous grasping and in-hand reorientation experiments.

1 Control and Perception

The Franka Panda arm is controlled at 20 Hz via the Deoxys framework [8]: the pre-grasp approach uses the Deoxys joint position controller, and during PBDS execution the streamed joint targets are tracked by the Deoxys joint impedance controller with a min-jerk position interpolator. The Allegro hand

is commanded over ZMQ with two control modes: joint-level PD control for free-space positioning (e.g., opening to a pre-grasp configuration) and Cartesian fingertip impedance control with gravity compensation for compliant grasping, with force closure regulated upstream by the PBDS controller.

For the grasping task, runtime perception is provided by an Intel RealSense D435 camera. Object segmentation uses the Segment Anything Model (SAM) [3], and 6-DOF pose tracking is provided by FoundationPose [7], streamed at 10 Hz to the controller via a Redis pub/sub interface with exponential smoothing to reduce pose jitter. During grasp execution, streaming perception continuously updates the object pose in the MuJoCo [6] simulation at each control step, enabling the PBDS controller to react to object motion.

For object modeling, we scan each object using KIRI Engine [2] with a LiDAR-equipped iPhone 14 Pro.

2 Task Specification

The grasping and finger-gaiting behaviors share the same overall structure but differ in a small number of policy-specific CBFs and in several gain choices, noted inline below.

- 1) A *joint space damping* task with task map $f_D = \text{id}: M \rightarrow \mathbb{R}^m$, equipped with a dissipative force $\mathcal{F}_D = -B\dot{\sigma}$ and the flat metric. For grasping, the damping matrix is diagonal with separate entries, $B_{\text{arm}} = 20I_7$ and $B_{\text{hand}} = 0.1I_{16}$. For in-hand reorientation, the arm is fixed and $B_{\text{hand}} = 2.0I_{16}$.
- 2) *Joint limit ECBF* tasks with task maps $f_k: M \rightarrow \mathbb{R}$, $k = 1, \dots, m$, given by the coordinate projections $f_k(\sigma) = \sigma^k$, each encoding the upper and lower joint limits. For grasping, the arm joints use softened gains $(\kappa_1, \kappa_2) = (2, 5)$ (their larger ranges and tighter acceleration limits make (10, 50) QP-infeasible), while the hand joints use (10, 50); for in-hand reorientation, all 16 hand joints use (10, 50).
- 3) *Force closure ECBF* tasks with task map $f_{fc,S}: M \rightarrow \mathbb{R}$ that computes the scaled l^* metric [5] over a subset $S \subseteq \{1, 2, 3, 4\}$ of fingertip contacts:

$$h_{0,S}(\sigma) = l_S^*(\sigma) \cdot |S| \cdot n_b - l_{\min}^*$$

where $n_b = 3$ is the number of basis wrenches per contact under the hard-contact friction model with coefficient $\mu = 0.3$, $l_{\min}^* = 0.01$, and $(\kappa_1, \kappa_2) = (2, 4)$. Both the full-grasp variant $S = \{1, 2, 3, 4\}$ and the four per-finger-excluded variants $S = \{1, 2, 3, 4\} \setminus \{i\}$ are instantiated; the FSM enables whichever is consistent with the currently contacting fingers.

- 4) *Fingertip lift-distance ECBF* tasks with task maps $f_{\text{lift},i}: M \rightarrow \mathbb{R}$, $i = 1, \dots, 4$, given by the signed distance of fingertip i to the object surface. The safety function is $h_{0,i} = d_i - d_{\text{lift}}$ with $d_{\text{lift}} = 10$ mm, keeping each stationary fingertip above the surface during the transit of another finger. Gains are $(\kappa_1, \kappa_2) = (12, 7)$ for grasping and (10, 10) for in-hand.
- 5) *Finger-finger distance ECBF* tasks with task maps $f_{ij}: M \rightarrow \mathbb{R}$, for each pair (i, j) of distinct fingers, given by the Euclidean distance d_{ij} between fingertip centers. Two barriers are defined per pair: a quadratic *spacing*

barrier $h_0 = -(d_{ij} - d_{\text{ref},ij})^2 + \epsilon^2$ with $\epsilon = 1.5$ mm that keeps each pair within a tolerance of its reference spacing $d_{\text{ref},ij}$, and a linear *minimum-separation* barrier $h_0 = d_{ij} - d_{\min}$ with $d_{\min} = 40$ mm that prevents fingertip collisions. Gains are $(\kappa_1, \kappa_2) = (12, 7)$ for grasping and (10, 10) for in-hand.

- 6) *Table avoidance ECBF* tasks with task maps $f_{\text{tab},i}: M \rightarrow \mathbb{R}$, $i = 1, \dots, 4$, given by the world frame height of each fingertip. The safety function is $h_{0,i} = z_i - z_{\text{table}} - p$ with $p = 10$ mm fingertip padding, enforced with $(\kappa_1, \kappa_2) = (10, 10)$.
- 7) *Link-object ECBF* tasks (grasping only) penalizing interference between non-fingertip hand geometry and the object, comprising: a per-finger *fingertip penetration* CBF with $h_{0,i} = d_i + 3$ mm (up to 3 mm tip ingress tolerated) and gains (12, 7); a *palm* CBF with $h_0 = d_{\text{palm}} - 5$ mm and gains (12, 7); per-finger *proximal phalanx* CBFs with $h_{0,i} = d_{\text{prox},i}$ and gains (4, 5); and per-finger *medial phalanx* CBFs with $h_{0,i} = d_{\text{med},i} + 5$ mm and gains (12, 7). The sign of the additive constant indicates how much ingress is permitted before the CBF activates.
- 8) *Distal link-object ECBF* tasks (in-hand only) with task maps $f_{\text{dist},i}: M \rightarrow \mathbb{R}$, $i = 1, \dots, 4$, given by the signed distance between the distal phalanx of finger i and the object surface, with $h_{0,i} = d_{\text{dist},i} + 20$ mm (allowing up to 20 mm ingress) and soft gains $(\kappa_1, \kappa_2) = (1, 2)$. This replaces the per-link CBFs of item 7, which are deactivated for the finger-gaiting policy.
- 9) *Fingertip to object distance actions*. For each fingertip we expose a control task map $f_{\text{lift},i}$ carrying a PD controller on the scheduled fingertip-to-object distance target. The grasping policy additionally exposes an *average-fingertip-position* action task (combined arm-hand only) that aligns the centroid of the four fingertips with the object's geometric center, and a *palm position* action task on the \mathbb{R}^3 end effector position. When all action inputs are zero, the system reduces to the autonomous PBDS defined by items 1–8.

For in-hand reorientation, the gaited fingertip's position is controlled in the frame of the trifinger plane spanned by the three in-contact fingertips. For each excluded finger i , we define a 10-dimensional *trifinger plane manifold* $P_{\setminus i}$, whose coordinates are the centroid (3D) and orientation quaternion (4D) of the plane spanned by the three non-lifted fingertips, together with the position of finger i relative to that plane (3D). The gaited-fingertip action task drives the last 3 coordinates of $P_{\setminus i}$ toward a planner-supplied target via PD control.

Task weights are modulated dynamically by a finite-state machine (FSM) that coordinates the grasp and gaiting phases (approach, contact, lift, traverse, drop). The FSM selects the active force closure variant $f_{fc,S}$ as fingers enter or leave contact, and toggles $f_{\text{lift},i}$ depending on whether finger i is stationary or in transit.

APPENDIX E

DEXTEROUS GRASPING TRIAL PROTOCOL

This appendix expands on the hardware grasping experiments in Section VIII-C. Each 4-finger grasping trial proceeds

as follows:

- 1) The stable rest pose of each object is determined and permuted across 3 table locations.
- 2) FoundationPose [7] estimates the object’s 6-DOF pose via the Redis streaming interface.
- 3) Candidate wrist poses are generated by sampling 18 yaw angles (every 20°) around the vertical axis. For each yaw, a top-down pose (palm facing downward) is computed above the object’s bounding-box center. Three standoff distances (0, 7.5, 15 mm) along the approach direction are evaluated, yielding up to 54 candidates per object.
- 4) Each candidate is filtered through a five-stage pipeline: (i) *object width check*: the object extent along the thumb-to-index aperture axis must not exceed 1.5 times the open-hand span; (ii) *arm inverse kinematics (IK)*: a MuJoCo-based IK solver verifies reachability with a 2 cm position tolerance, 5° rotation tolerance, and a 5% joint limit margin; (iii) *palm-object collision*: the palm axis-aligned bounding box is checked against the object with a 1 cm clearance margin; (iv) *hand-object collision*: full hand-object contact detection via MuJoCo rejects poses with penetration; and (v) *aperture verification*: the object center must lie between the thumb and opposing fingers, and in front of the palm. Valid candidates are ranked by a quality metric that favors tight width fit and small standoff distance, and the top-ranked candidate is selected for hardware execution.
- 5) Using the latest perception update, candidate generation and ranking are repeated at the start of each hardware attempt, so the chosen pose reflects the current object estimate.
- 6) On hardware, the arm first moves to its home configuration and the hand opens to a pre-grasp pose (using a two-phase open: fingertip impedance control for coarse motion, then joint PD for fine positioning). The arm then executes a two-stage approach: first moving to a waypoint 15 cm above the grasp pose, then descending to the final wrist pose via joint-position control through Deoxys [8].
- 7) The simulation and hardware joint configurations are synchronized, after which SafePBDS takes over the full 23-DOF arm-hand system in MuJoCo [6]. Two action tasks are composed at each control step: (i) per-finger *fingertip-to-object distance* tasks that PD-servo each fingertip toward the object surface with a clipped acceleration command, and (ii) an *average-fingertip centering* task on the wrist that drives the centroid of the four fingertips toward the object’s mesh geometric center, using adaptive gains that taper as the fingertips close in and that freeze the wrist once all four fingertips are within 1 cm of the surface. The force closure ECBF (Appendix D.2) is enabled once all active fingertips are within 5 cm of the object, so that the final squeeze converges to a certified force-closed grasp. The action tasks and ECBFs are resolved by the PBDS QP into a joint-acceleration command that is integrated through inverse dynamics, with the arm portion streamed back to Deoxys and the hand portion streamed to the Allegro.

Object pose is continuously updated from the streaming perception pipeline; once the hand is near the object, visual occlusions become severe, so we hold the last reliable pose observation for the remainder of the grasp execution. A trial is recorded as a *success* if the object is stably lifted with all four fingers in contact, a *partial success* if the object is stably lifted but one finger is not in contact with the object surface, and a *failure* otherwise.

a) *3-finger variant.*: For the 3-finger ablation, the active force closure CBF is switched to the per-finger-excluded variant $f_{fc,S\setminus\{i\}}$ (Appendix D.2, item 3), so that l^* is computed over the three in-contact fingers. The excluded finger is held clear of the object by (i) retargeting its fingertip-to-object distance action to a fixed 50 mm clearance under the same PD law, and (ii) enabling its fingertip lift-distance CBF (Appendix D.2, item 4) to enforce the 10 mm minimum fingertip-to-object clearance. Besides changing the scalar action target and the active force closure variant, no other policy modification is made.

TABLE A2
4-FINGER GRASPING RESULTS ACROSS 20 HOUSEHOLD OBJECTS. EACH OBJECT IS TESTED OVER 6 TRIALS (2 PER TABLE LOCATION).

| Object | Succ. | Object | Succ. |
|----------------|------------------------|-------------|----------------|
| Ball | 6 | Cylinder | 5 |
| Bottle | 6 | Flapjack | 6 [†] |
| Camera (large) | 6 | Meat | 6 |
| Camera (small) | 6 | Mustard | 5 [†] |
| Cereal | 6 | Onion | 5 |
| Cracker | 6 | Prism | 6 |
| Chicken | 6 | Ramen | 5 |
| Chips | 6 [†] | Soup | 4 |
| Cocoa | 6 | Wafer | 6 |
| Cube | 6 | YCB Mustard | 3 [†] |
| Overall | 111/120 (92.5%) | | |

[†]Includes 1 partial success (finger not in contact with object).

TABLE A3
3-FINGER GRASPING RESULTS WITH ONE FINGER EXCLUDED, TESTED ACROSS 3 TABLE LOCATIONS PER CONFIGURATION. FOLDED INTO FIGURE 6 IN THE MAIN TEXT; REPRODUCED HERE FOR THE PER-OBJECT BREAKDOWN.

| Object | Thumb | Index | Middle | Ring | Total |
|----------------|----------|----------|----------|----------|-------------------|
| Cylinder | 3 | 3 | 3 | 3 | 12 |
| Flapjack | 3 | 3 | 3 | 3 | 12 |
| Camera (large) | 1 | 3 | 3 | 3 | 10 |
| Total | 7 | 9 | 9 | 9 | 34 (94.4%) |

APPENDIX F

IN-HAND REORIENTATION PLANNER AND STATE MACHINE

This appendix expands on the offline simulation planner and the lift-and-drop state machine described in Section VIII-D.

1 Depth-First Search Planner

Each node in the search tree represents a state with all 4 fingers in contact with the object, and each edge represents moving one finger to a different contact location. The search is rooted at an initial grasp computed by reusing the grasping pipeline of Section VIII-C: we apply the 4-finger grasp computation to a 60 mm-diameter, 85 mm-tall cylinder, with the hand reoriented so that the palm faces downward over the

object. The resulting hand configuration is opened slightly so that each fingertip is 5 mm from the object surface.

Among nodes at the same depth, the priority queue orders by cumulative z -axis rotation progress,

$$\Delta\psi_{\text{total}} = \sum_{k=1}^{N_s} \Delta\psi_k, \quad (59)$$

where $\Delta\psi_k$ is the incremental yaw change at step k along the path from the root; this accumulation properly handles rotations exceeding $\pm 180^\circ$. To prevent over-reliance on a single finger, we additionally constrain the move counts across fingers to remain balanced,

$$\max_{i,j} |c_i - c_j| \leq \Delta c_{\text{max}}, \quad (60)$$

where c_i is the number of times finger i has been moved; empirically $\Delta c_{\text{max}} = 1$ yields the best rotation progress. The search terminates once a maximum node count is reached, returning the path to the node with highest rotation progress.

To prevent the search from getting stuck with a never-movable finger, the planner maintains a cyclic sequence of “next movable” fingers, requiring that at each tree depth, the “next movable” finger be movable in any accepted candidate’s resulting state. The cycle is fixed by hand side and target rotation direction: for the right hand and counterclockwise rotation we use INDEX→MIDDLE→RING→THUMB; for clockwise rotation we use RING→MIDDLE→INDEX→THUMB. At a node of depth d , the next movable finger f_d^{next} is $\text{order}[(s_0 + d) \bmod 4]$, where s_0 is the index in `order` of the first movable finger at the root. The specific cyclic order is not load-bearing; any cycle that visits all four fingers serves the same purpose, namely keeping every finger periodically responsible for gating acceptance.

At each node, the planner identifies which fingers may be lifted without violating force closure, and additionally excludes the finger moved at the parent node so that no finger is moved on two consecutive steps. For each remaining movable finger i , candidate actions are generated by sampling:

- 1) Step sizes of $\delta \in \{10, 20, 30, 40\}$ mm on the world XY plane along the object surface, and
- 2) Aperture angles $\phi \in \{30^\circ, 50^\circ, 70^\circ, 90^\circ\}$ controlling the finger’s approach direction in the plane perpendicular to the object axis.

2 Lift-and-Drop State Machine

Each candidate action is forward-simulated using `SafePBDS` and evaluated by a four-phase state machine that governs a single lift-and-drop primitive: `LIFTING`, `TRAVERSING`, `DROPPING`, and `ADJUSTING`. In `LIFTING`, the moving finger is pulled clear of the object; once the fingertip has cleared by a small lift margin, the machine advances to `TRAVERSING`, during which the finger is steered toward the target contact location. Upon aligning with the target in the plane orthogonal to the approach direction, the machine enters `DROPPING`, which closes the finger onto the object until re-contact is detected, and finally `ADJUSTING`, which waits for the hand to come to rest. Throughout all four phases, the three in-contact fingers are held in place by the `PBDS` controller through the force closure `ECBF` on the

three-finger support set, which keeps them force-closed with the object, and pairwise fingertip spacing `ECBFs` over pairs of in-contact fingers preserve their relative geometry. During `TRAVERSING` and `DROPPING`, the force closure `ECBF` excluding the next movable finger f_d^{next} (defined above) is additionally pre-enabled, so that the candidate’s resulting state retains movability for that finger.

State transitions are triggered as follows:

- `LIFTING` → `TRAVERSING`: the moving finger’s fingertip-to-object distance exceeds 0.007 m, which is 70% of the nominal 0.01 m lift clearance.
- `TRAVERSING` → `DROPPING`: the component of the fingertip tracking error orthogonal to the approach direction falls below 0.01 m.
- `DROPPING` → `ADJUSTING`: the re-contacting fingertip-to-object distance drops below the contact threshold of 0.002 m.

A transition is accepted only if the machine reaches `ADJUSTING` and, in that state, simultaneously satisfies all of the following: the moving finger is aligned with its target contact (orthogonal tracking error below 0.01 m), the hand has come to rest (joint-velocity norm below 0.2 rad/s), all four fingers are in contact with the object (each finger satisfying both the fingertip and fingertip-center object distances below 0.002 m), and the force closure `ECBF` excluding f_d^{next} remains non-negative. We additionally require the object tilt at this resulting state to stay below $\theta_{\text{tilt}}^{\text{max}} = 10^\circ$, evaluated from the object orientation relative to the search root. Any loss of contact at an in-contact finger during the rollout aborts execution and rejects the transition; failure to meet the `ADJUSTING` success conditions, or a tilt exceeding $\theta_{\text{tilt}}^{\text{max}}$ at the resulting state, likewise rejects the transition.

APPENDIX REFERENCES

- [1] Andrew Bylard, Riccardo Bonalli, and Marco Pavone. “Composable Geometric Motion Policies using Multi-Task Pullback Bundle Dynamical Systems”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 7464–7470.
- [2] KIRI Innovations. *KIRI Engine: 3D Scanner App*. <https://www.kiriengine.app/>. Photogrammetry and LiDAR-based 3D scanning application. 2024.
- [3] Alexander Kirillov et al. “Segment Anything”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2023, pp. 3992–4003.
- [4] John M. Lee. *Introduction to Riemannian Manifolds*. 2nd ed. Vol. 176. Graduate Texts in Mathematics. Springer, 2018.
- [5] Albert H. Li et al. “FRoGGeR: Fast Robust Grasp Generation via the Min-Weight Metric”. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2023, pp. 6809–6816.
- [6] Emanuel Todorov, Tom Erez, and Yuval Tassa. “MuJoCo: A physics engine for model-based control”. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2012, pp. 5026–5033.

- [7] Bowen Wen et al. “FoundationPose: Unified 6D Pose Estimation and Tracking of Novel Objects”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024, pp. 17868–17879.
- [8] Yifeng Zhu et al. “VIOLA: Imitation Learning for Vision-Based Manipulation with Object Proposal Priors”. In: *Proceedings of The 6th Conference on Robot Learning (CoRL)*. Vol. 205. Proceedings of Machine Learning Research. PMLR, 2023, pp. 1199–1210.