Composable Geometric Motion Policies using Multi-Task Pullback Bundle Dynamical Systems

Andrew Bylard, Riccardo Bonalli, Marco Pavone

***Abstract*— Despite decades of work in fast reactive planning and control, challenges remain in developing reactive motion policies on non-Euclidean manifolds and enforcing constraints while avoiding undesirable potential function local minima. This work presents a principled method for designing and fusing desired robot task behaviors into a stable robot motion policy, leveraging the geometric structure of non-Euclidean manifolds, which are prevalent in robot configuration and task spaces. Our Pullback Bundle Dynamical Systems (PBDS) framework drives desired task behaviors and prioritizes tasks using separate position-dependent and position/velocity-dependent Riemannian metrics, respectively, thus simplifying individual task design and modular composition of tasks. For enforcing constraints, we provide a class of metric-based tasks, eliminating local minima by imposing non-conflicting potential functions only for goal region attraction. We also provide a geometric optimization problem for combining tasks inspired by Riemannian Motion Policies (RMPs) that reduces to a simple least-squares problem, and we show that our approach is geometrically well-defined. We demonstrate the PBDS framework on the sphere $\mathbb{S}^2$ and at 300-500 Hz on a manipulator arm, and we provide task design guidance and an open-source Julia library implementation. Overall, this work presents a fast, easy-to-use framework for generating motion policies without unwanted potential function local minima on general manifolds.**

I. Introduction

Fast reactive planning and control is often crucial in dynamic, uncertain environments, and related techniques have a long history [1]–[5]. However, key challenges remain in applying these techniques to general manifolds. Well-known non-Euclidean manifolds are ubiquitous in robotics, often representing part of the configuration manifold (e.g., $SO(3)$ for aerial robot attitude). They can also be essential for representing robotic task spaces (e.g., the sphere $\mathbb{S}^2$ for painting or moving a fingertip around an object surface, such as in Fig. 1). Further, constraints may restrict the robot to a free submanifold which is difficult to represent explicitly. For example, an obstacle in the workspace can produce an additional topological hole that geodesics or "default" unforced trajectories should avoid [6], [7].

Often such constraints are handled by constrained optimization or by repulsive potentials. However, constrained optimization can scale poorly with the number and complexity of nonconvex constraints, often becoming impractical for real-time MPC without careful initialization [8]–[10]. On the other hand, repulsive potentials can be difficult to design without producing incorrect behavior. For example, it is well known that artificial potential function (APF) techniques [1], [2] without assumptions about the robot and workspace geometry [11]–[13] can produce many incorrect local minima and unnatural behavior such as oscillations within the free submanifold (e.g. in narrowing passages or when a constraint is near the goal) [14], [15].

A. Bylard, R. Bonalli, and M. Pavone are with the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305. {bylard, rbonalli, pavone} @stanford.edu. This work was supported in part by NSF Cyber-Physical Systems (CPS) Grant No. 1931815 and by KACST. Thanks to Fransesco Bullo, Benoit Landry, Thomas Lew, and Jean-Jacques Slotine for their helpful discussions and insights.

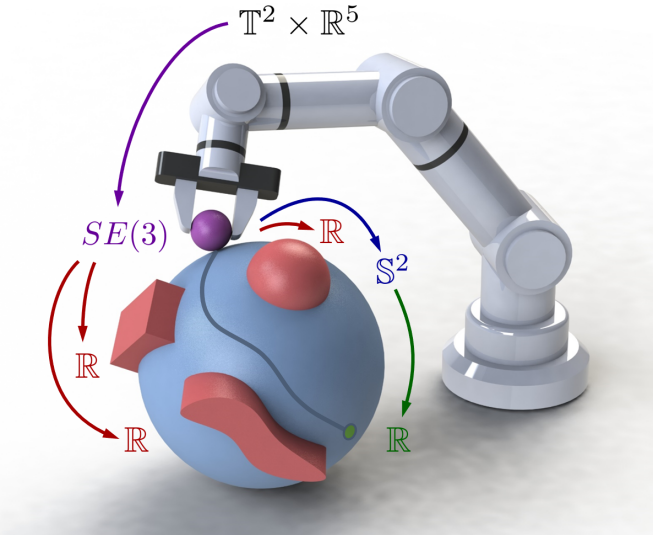


Fig. 1: Example tree of PBDS task mappings designed to move a ball along the surface of a sphere to a goal while avoiding obstacles. Depicted are manifolds representing: (black) joint configuration for a 7-DoF robot arm with two fully revolute joints, (purple) ball pose, (red) distances to obstacles, (blue) position on the sphere, and (green) distance to the goal. Note that damping can be defined on $\mathbb{T}^2 \times \mathbb{R}^5$, $SE(3)$, and/or $\mathbb{S}^2$ as desired.

An alternative coming from differential geometry is to encode constraints not with forces, but with *metrics*. For example, correctly designed Riemannian metrics [16] defined on the robot configuration manifold have been proposed to curve the manifold to prevent constraint violation not due to forces pushing the robot away, but due to the space stretching infinitely in the direction of constraints [17], [18]. Such a reliance on curvature rather than competing potential functions may also eliminate traps due to potential function local minima [19].

However, correctly designing such metrics directly on the configuration manifold can be difficult, motivating the use of *pullback metrics*, i.e. metrics are designed on task manifolds where constraints are naturally defined and then "pulled back" through the task map onto the configuration manifold. One recent work advocating this approach is RMPflow [17], which develops a tree of subtask manifolds with metrics and forces defined on the leaf manifolds which are pulled back and combined into a single acceleration policy on the configuration manifold. However, [17] uses potentials in addition to metrics to represent constraint-enforcing subtasks, increasing design complexity and leading to the same local minima issues as in other APF approaches.

Additionally, it is often useful to perform task prioritization dependent on robot velocity. For example, when a robot is moving away from an obstacle, the obstacle can often be safely ignored. To this end, [17] introduces Geometric Dynamical Systems (GDSs) weighted by Riemannian Motion Policies (RMPs) [20], which allows velocity-dependent met-

rics used both to specify subtask behavior (such as constraint avoidance) and to prioritize tasks. However, this approach presents two difficulties: First, it is very difficult to design task metrics which both specify desirable task behavior and prioritize tasks correctly. This has led to investigating learning the task metrics from demonstrations [21], but in any case, it is unclear what task behavior within a single task could benefit from velocity-dependent Riemannian metrics, as we discuss in Sec. IV-B.

Second, a GDS with velocity-dependent metrics does not meet the key property of *geometric consistency*, as we demonstrate in Sec. IV-A. Geometric consistency[1] (i.e. invariance to changes in coordinate representations) is an essential property of any differential geometric method which ensures that the quantities defined, in this case acceleration policies, are well-defined on the robot and task manifolds invoked, thus correctly capturing and leveraging the structure of these manifolds (for example, see Fig. 5).

To summarize, a large gap remains for producing a fast, easy-to-use, and geometrically-consistent approach for generating motion policies on general robot and task manifolds, achieving velocity-dependent task-weighting and leveraging metric-based enforcement of constraints to eliminate unwanted potential function local minima.

Statement of Contributions: To this end, we provide the following contributions:

1) We present the Pullback Bundle Dynamical Systems (PBDS) framework for combining multiple geometric task behaviors into a single robot motion policy, while maintaining geometric consistency and stability. In doing so, we provide a geometrically well-defined formulation of a weighting scheme inspired by RMPs, which reduces to a simple least-squares problem. We also remove the tension between single-task behavior and inter-task weighting by introducing separate velocity-dependent weighting pseudo-metrics for each task to handle task prioritization.

2) We apply the PBDS framework to tangent bundles to show limits to the practical use of velocity-dependent Riemannian metrics for task behavior design and to show why GDSs do not maintain geometric consistency.

3) We provide a class of constraint-enforcing tasks encoded solely via simple, analytical Riemannian metrics that stretch the space, rather than via traditional barrier function potentials, eliminating potential function local minima.

4) We demonstrate PBDS policy behavior in numerical experiments and at 300-500 Hz on a 7-DoF arm, and we provide a fast open-source Julia library called PBDS.jl.

Paper Organization: The paper is organized as follows. In Sec. II we recall relevant concepts from Riemannian geometry and establish some notation. Then Sec. III, builds a multi-task PBDS, extracts a motion policy, and provides stability results. Sec. IV unpacks key advantages over RMPflow's GDS/RMP framework. Finally, Sec. V provides implementation details and a robot arm demonstration.

II. Geometric Preliminaries

Here we recall some results in Riemannian geometry and establish notation that will be used throughout the paper. For a more detailed introduction, see [16], [22]. Let M be a smooth m-dimensional manifold, equipped with charts

[1]Geometrically-consistent objects are also known of as being "global" in differential geometry [22]. Thus we will also refer to such objects as being geometrically or globally well-defined.

$$\begin{array}{ccccc} TM & \xrightarrow{f^*df} & f^*TN & \xrightarrow{f^*\mathrm{Id}_{TN}} & TN \\ & \searrow^{\pi_M} & \downarrow{\scriptstyle f^*\pi_N} & & \downarrow{\scriptstyle \pi_N} \\ & & M & \xrightarrow{f} & N \end{array}$$

Fig. 2: Commutative diagram of the key manifolds used to form a pullback bundle dynamical system. Note that the pullback bundle f^*TN is an n-vector bundle over the robot configuration manifold M.

assigning coordinates to locally-Euclidean patches. M has a smooth tangent bundle TM containing tangent spaces T_pM at each point $p \in M$, and it has a cotangent bundle T^*M, which is a natural place to define forces [23]. We will also denote $(p, v) \in T_pM$ as v_p or v depending on desired emphasis of the base point p. Given a Riemannian metric g, the couple (M, g) is a Riemannian manifold. The metric provides a smoothly-varying inner product $g_p : T_pM \times T_pM \longrightarrow \mathbb{R}$ at each point and thus induces norms $||\cdot||_{g_p}$. The metric g also gives a "sharp" operator $\sharp : T^*M \longrightarrow TM$ and generalized gradient $\mathrm{grad} f : M \longrightarrow TM$, with $f \in C^\infty(M)$, which are useful for applying forces. A choice of connection ∇ in TM (e.g. the standard Levi-Civita connection) assigns a unique acceleration operator D_σ to each curve σ in M. Thus we can compute acceleration $D_\sigma\sigma' = \nabla_{\sigma'(t)}\sigma'$, where σ' is the velocity along σ (used interchangeably with $\dot{\sigma}$ notation, which can also indicate a time derivative). Then by choosing generalized forces $\mathcal{F} : TM \longrightarrow T^*M$, one can specify a dynamical system on M satisfying $D_\sigma\sigma'(t) = \mathcal{F}(\sigma(t))^\sharp$. This can be written in local coordinates as the second-order differential equations

$$\ddot{\sigma}^k(t) + \dot{\sigma}^i(t)\dot{\sigma}^j(t)\Gamma^k_{ij}(\sigma(t)) = g^{kj}(\sigma(t))\mathcal{F}_j(\sigma(t), \dot{\sigma}(t)), \quad (1)$$

where we make use of Einstein index notation and where Γ^k_{ij} are the Christoffel symbols associated with ∇. We will use this type of dynamical system to design desired behavior on task manifolds that corresponding robot motion policies should aim to replicate.

Note also that we will use bold symbols when convenient to denote local matrix or vector representations of objects. For example, given $\Xi^k \triangleq \dot{\sigma}^i\dot{\sigma}^j\Gamma^k_{ij}$, we can write (1) as

$$\ddot{\boldsymbol{\sigma}}(t) + \boldsymbol{\Xi}(\boldsymbol{\sigma}(t), \dot{\boldsymbol{\sigma}}(t)) = \boldsymbol{g}^{-1}(\boldsymbol{\sigma}(t))\boldsymbol{\mathcal{F}}(\boldsymbol{\sigma}(t), \dot{\boldsymbol{\sigma}}(t)).$$

We may also abbreviate such expressions, for example as

$$\ddot{\boldsymbol{\sigma}} + \boldsymbol{\Xi} = \boldsymbol{g}^{-1}\boldsymbol{\mathcal{F}}.$$

III. Pullback Bundle Dynamical Systems

Given k tasks, let M and N_i be smooth m- and n_i-dimensional robot configuration and task manifolds, respectively, where $\{f_i : M \longrightarrow N_i\}_{i=1,\dots,k}$ is a set of smooth task maps. These tasks could represent a variety of desired behaviors such as attraction to a point/region, velocity damping, or avoidance of a constraint. To illustrate, consider a running pedagogical example of a point robot on the surface of a ball, aiming to reach a goal while avoiding obstacles. A simple design for the relevant tasks is shown in Table I. As shown, the desired task behaviors can be encoded by designing on each task manifold N_i a potential function $\Phi_i : N_i \longrightarrow \mathbb{R}_+$, dissipative forces $\mathcal{F}_{D,i} : TN_i \longrightarrow T^*N_i$, and a Riemannian metric g_i (which we can refer to as the behavior metric), all of which are smooth. We also choose the standard Levi-Civita connection ∇_i [22].

As mentioned, these components give dynamical systems on the task manifolds N_i, but we want to form a robot motion policy from a dynamical system on the configuration

Task	Task Map	Behavior Metric	Dissipative Forces	Potential	Weight pseudometric
Goal Attraction	$f_1 : \mathbb{S}^2 \to \mathbb{R} : p \mapsto \text{dist}(p, p_{\text{goal}})$	$\boldsymbol{g}_1 = 1$	$\boldsymbol{\mathcal{F}}_{D,1} = 0$	$\Phi_1 = \|\boldsymbol{x}\|_2^2$	$\boldsymbol{w}_i = \boldsymbol{I}_2$
Damping	$f_2 : \mathbb{S}^2 \to \mathbb{S}^2 : p \mapsto p$	$\bar{\boldsymbol{g}}_2 = \boldsymbol{I}_3$	$\bar{\boldsymbol{\mathcal{F}}}_{D,2} = -4\dot{\bar{\boldsymbol{x}}}$	$\Phi_2 = 0$	$\bar{\boldsymbol{w}}_2 = \boldsymbol{I}_6$
Obstacle Avoidance	$f_i : \mathbb{S}^2 \to \mathbb{R}_+ : p \mapsto \text{dist}(p, \mathcal{X}_{\text{obs}})$	$\boldsymbol{g}_i = \exp(1/(2\boldsymbol{x}^2))$	$\boldsymbol{\mathcal{F}}_{D,i} = 0$	$\Phi_i = 0$	See (8)

TABLE I: Simple task design summary for running example of a robot on a sphere surface navigating to a goal point while avoiding obstacles. Note that each obstacle has an avoidance tasks (hence the indices i). For the damping task, the overbars denote representations in ambient Euclidean space, which are easily pulled back through standard embedding maps to induce corresponding objects on $\mathbb{S}^2$ and $\mathbb{T}S^2$ (see Appx. A.II for details).

manifold M. One way to proceed is by using pullbacks, i.e. "pulling back" the necessary objects through the task maps f_i to operate over M. First, we recall the definition of the pullback bundle of TN, omitting task indices for simplicity:

$$f^*TN = \coprod_{p \in M} \pi_N^{-1}(f(p)), \tag{2}$$

where $\coprod$ is a disjoint union and π_N is the standard projection on TN. This is itself a well-defined vector bundle, and in particular is an $(m+n)$-dimensional manifold. The relationships between these manifolds are depicted in Fig 2, including a pullback differential $f^*df : TM \longrightarrow f^*TN : (p, v) \mapsto (p, \pi_2(df_p(v)))$, where π_2 denotes a projection onto the velocity component.

Next we define the pullback connection $f^*\nabla$, whose Christoffel symbols are given locally by

$$f^*\Gamma_{ij}^k(p) = \frac{\partial f^\ell}{\partial x^i}(p)\Gamma_{\ell j}^k(f(p))$$

for all $p \in M$. In Lemma C.1 we show this connection is globally well-defined and compatible with a pullback metric f^*g. This connection gives a corresponding pullback acceleration operator f^*D_γ for each curve γ in M. We can also denote the total acceleration given by the pullback forces by $f^*\mathcal{F}(\cdot)^\sharp : f^*TN \longrightarrow f^*TN : (p, v) \mapsto f^*\mathcal{F}_D(p, v)^\sharp - f^*\text{grad}\,\Phi(p)$, where we define the pullback dissipation forces $f^*\mathcal{F}_D$ and pullback gradient $f^*\text{grad}$ in Appx. C.I.

We are now equipped to construct a key building block for forming a full Pullback Bundle Dynamical System:

Definition III.1 (Local Pullback Bundle Dynamical System)**.** *Let $f : M \longrightarrow N$ be a smooth task map. Then for each $(p, v) \in M$, we can choose a curve $\alpha_{(p,v)} : (-\varepsilon, \varepsilon) \longrightarrow M$ resulting in $\gamma_{\alpha_{(p,v)}} : (-\varepsilon, \varepsilon) \longrightarrow f^*TN$ for some $\varepsilon > 0$ such that $(f, g, \Phi, \mathcal{F}_D, \alpha_{(p,v)})$ forms a local Pullback Bundle Dynamical System (PBDS) satisfying*

$$\textit{PBDS}_{\alpha_{(p,v)}} \begin{cases} f^*D_{\alpha_{(p,v)}}\gamma_{\alpha_{(p,v)}}(s) = f^*\mathcal{F}(\gamma_{\alpha_{(p,v)}}(s))^\sharp \\ \gamma_{\alpha_{(p,v)}}(0) = f^*df_{\alpha_{(p,v)}(0)}(\alpha'_{(p,v)}(0)), \\ \alpha'_{(p,v)}(0) = (p, v). \end{cases}$$

This local PBDS construction is local in the sense that in practice, we define a $\text{PBDS}_{\alpha_{(p,v)}}$ for each $(p, v) \in TM$ and only evaluate it at (p, v). However, it is geometrically well-defined and is required to ensure that we have well-defined dynamics at each point on the pullback bundle independent of the robot curve $\sigma : [0, \infty) \longrightarrow M$ that we ultimately follow. In particular, these dynamics must be well-posed at $t = 0$, (i.e. where f^*D_σ is not well-defined, requiring another curve $\alpha_{\sigma'(0)}$ to produce $f^*D_{\alpha_{\sigma'(0)}}$) and in cases where the curve $\alpha_{\sigma'(t)}$ defining a valid $\text{PBDS}_{\alpha_{\sigma'(t)}}$ for some $t \in [0, \infty)$ is not unique. For example, the latter may occur when $m > n$, which is often the case in practice.

From these local PBDS dynamics, we can extract the corresponding desired task pullback acceleration associated with each robot position and velocity:

$$S : TM \longrightarrow T(f^*TN) : (p, v) \mapsto \pi_{\text{VB}}(\gamma'_{\alpha_{(p,v)}}(0)), \tag{3}$$

where π_{VB} denotes the projection onto the vertical bundle. In Appx. C.III, we show that this map is geometrically well-defined. In particular, a curve $\alpha_{(p,v)}$ forming a valid local PBDS always exists, and the map S does not depend on the particular choices of curves $\alpha_{(p,v)}$. Given this fact, we will omit further mention of α and use the shorthand $\gamma'_{v_p}(0) \triangleq \gamma'_{\alpha_{(p,v)}}(0)$ for $(p, v) \in TM$.

Now for each individual task, we can form a map S_i as above to retrieve corresponding desired task pullback accelerations. However, our goal is to combine these tasks into a single robot motion policy. To do this, one approach is to define a robot acceleration policy using a geometrically well-defined optimization problem designed to strike a weighted balance between these pullback task acceleration policies.

This is not as straightforward as it may appear since the operative accelerations are each in different spaces and are thus difficult to compare (i.e. $\ddot{\sigma}$ is in TTM and $\gamma'_{\dot{\sigma}(t)}$ is in $T(f_i^*TN_i)$ for each task). To resolve this, we form a map relating robot accelerations on TTM to their resulting pullback task accelerations in $T(f_i^*TN_i)$:

$$\begin{aligned} &Z_i : TTM \longrightarrow T(f_i^*TN_i) \\ &\big((p, v), a\big) \mapsto \pi_{\text{VB}}\Big(d(f_i^*df_i)_{(p,v)}(a)\Big). \end{aligned} \tag{4}$$

Also, for each task we define a Riemannian pseudometric w_i on TN_i which will provide its weighting against other tasks, and we impose the following assumptions:

- $(A1)$ The pseudometrics w_i are at every point in TN_i either positive-definite or zero.
- $(A2)$ The Jacobian of the product map of the task maps associated with nonzero weights has rank m.

In practice, these weighting pseudometrics are often trivial to design (e.g., see our running example in Table I, where the attractor task weight is the Euclidean metric, and the damping task weight is induced by a Euclidean metric). However, they can also be used to switch tasks on and off, which is useful for enforcing constraints as shown in Sec. IV-C. Also, as in the running example, $(A1)$ and $(A2)$ are easy to satisfy. In particular, it is typical to use one or more damping tasks which together always provide dissipation along all degrees of freedom of the robot. Now to conclude, each w_i also has a pullback pseudometric $F_i^*w_i$ on $T(f_i^*TN_i)$ defined using the natural higher-order task map $F_i : TM \to TN_i : (p, v) \mapsto (p, (df_i)_p(v))$ (Appx. C.III).

We can now form the desired ODE on our robot manifold:

Definition III.2 (Multi-Task Pullback Bundle Dynamical System)**.** *Let $\{f_i : M \longrightarrow N_i\}_{i=1,\dots,k}$ be smooth task maps. 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 $\sigma : [0, \infty) \longrightarrow M$ satisfying*

$$\begin{cases} \ddot{\sigma}(t) = \underset{a \in \mathcal{D}_{\dot{\sigma}(t)}}{\arg\min} \ \sum_{i=1}^k \frac{1}{2}\|Z_i(a) - S_i(\dot{\sigma}(t))\|_{F_i^*w_i}^2 \\ \dot{\sigma}(0) = (p_0, v_0). \end{cases} \tag{5}$$

Here, $\mathcal{D}$ is the globally well-defined distribution of TTM such that the subspace $\mathcal{D}_{(p,v)} \subseteq T_{(p,v)}$ satisfies $a^v = v$ componentwise for each $a = ((p, v), (a^v, a^a)) \in \mathcal{D}_{(p,v)}$.

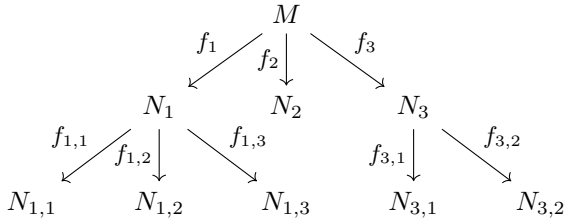


Fig. 3: Tree of manifolds and task maps in a multi-task PBDS policy. At the root is the robot configuration manifold, and task manifolds are at the leaves. This structure can be exploited to parallelize computation of contributions from various tasks and reuse computation from parent nodes.

Under assumptions $(A1)$-$(A2)$, as we show in Appx. C.III, the multi-task PBDS is geometrically well-defined and has a unique smooth solution. This gives us a dynamical system on M that provides our robot acceleration policy:

$$\ddot{\boldsymbol{\sigma}} = \left(\sum_{i=1}^{k} \boldsymbol{J}\boldsymbol{f}_i^\top \boldsymbol{w}_i^a \boldsymbol{J}\boldsymbol{f}_i\right)^\dagger \left(\sum_{i=1}^{k} \boldsymbol{J}\boldsymbol{f}_i^\top \boldsymbol{w}_i^a \boldsymbol{\mathcal{A}}_i\right), \tag{6}$$

$$\begin{aligned} \boldsymbol{\mathcal{A}}_i &= \boldsymbol{g}_i^{-1}(\boldsymbol{\mathcal{F}}_{D,i} - \nabla\boldsymbol{\Phi}_i) - (\dot{\boldsymbol{J}}\boldsymbol{f}_i - \boldsymbol{\Xi}_i)\dot{\boldsymbol{\sigma}} \\ (\Xi_i)_{kj} &= (Jf_i)_{\ell j}(\Gamma_i)^k_{\ell h}(Jf_i)_{hr}\dot{\sigma}^r \end{aligned} \tag{7}$$

where ∇ in this context is the Euclidean gradient operator, and where $\boldsymbol{w}_i^a \in \mathbb{R}^{n_i \times n_i}$ is the lower-right quadrant of the local matrix $\boldsymbol{w}_i \in \mathbb{R}^{2n_i \times 2n_i}$. In practice, this is the only quadrant necessary to design due to the vertical bundle projections in (3) and (4).

Next, we use LaSalle's invariance principle to demonstrate stability about a set of robot equilibrium states. This requires a Lyapunov function, which we build adapting known results in Lagrangian mechanics to the multi-task PBDS. In particular, we rely on one more key assumption:

$(A3)$ The combined dissipative forces corresponding to tasks having nonzero weights are strictly dissipative.

Like the others, $(A3)$ can be naturally satisfied (see the discussion of $(A2)$). Now consider the Lyapunov function candidate on the robot tangent bundle TM:

$$V(p,v) = \sum_{i=1}^{k} \frac{1}{2}\left\|(f_i^* df_i)_p(v)\right\|^2_{(f_i^* g_i)_p} + \Phi_i \circ f_i(p).$$

In Appx. C.III, we show that $\dot{V}(\sigma'(t)) < 0$ outside equilibrium states, leading to the following stability results:

Theorem III.1 (Global Stability of Multi-Task PBDS)**.** *Let $\{(f_i, g_i, \Phi_i, \mathcal{F}_{D,i}, w_i)\}_{i=1,\dots,k}$ be a multi-task PBDS where $\Phi_i \circ f_i$ are proper maps. Then given the Lyapunov function V above, the multi-task PBDS satisfies these:*

- *The sublevel sets of V are compact and positively invariant for* (5)*, so that every solution curve σ is defined in the whole interval $[0, +\infty)$.*
- *For every $\beta > 0$, every solution curve σ in M starting at $\dot{\sigma}(0) \in V^{-1}([0,\beta])$ converges in $V^{-1}([0,\beta])$ as $t \to +\infty$ to an equilibrium state $(p, 0) \in TM$ such that grad $\Phi_i(p) = 0$ if $w_i(p, 0) \neq 0$.*

IV. Comparison to Riemannian Motion Policies on Geometric Dynamical Systems

As mentioned, prior work proposed a framework called RMPflow which presented Geometric Dynamical Systems (GDS) for designing dynamical systems on task manifolds that could be "pulled back" onto the robot manifold to generate a robot motion policy [17]. Similar to this work, multiple GDSs could be combined through weighted optimization, in that case by using Riemannian Motion Policies (RMP) [20]. In this section, we will explain the key difficulties with the RMPflow approach (which we will refer to as GDS/RMP) and how they are resolved by our multi-task PBDS policies.

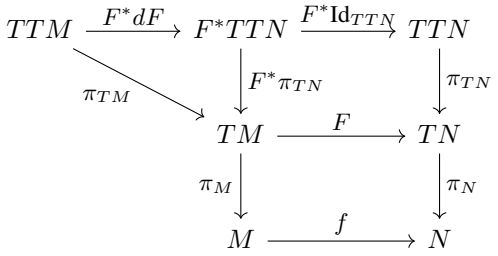


Fig. 4: Commutative diagram for a PBDS system applied to a task map between tangent bundles.

A. Geometric Consistency

The main shortcoming of the GDS framework is that it is in general not geometrically consistent. Geometric consistency is an essential property for any geometric method, ensuring its properties hold on non-Euclidean manifolds (i.e. outside a single locally-Euclidean patch) and are invariant to changes in coordinates. To see where geometric consistency is lost, we can derive the GDS framework as a modification of a PBDS applied to tangent bundles as follows.

For a given task map $f : M \longrightarrow N$, we define a higher-order task map $F : TM \longrightarrow TN$ as in Sec. III. Since tangent bundles also have a manifold structure, we can build a PBDS on F using the process in Sec. III, now operating on the higher-order manifolds shown in Fig. 4. However, now it is difficult to extract a robot acceleration policy corresponding to a curve on the robot manifold M because the PBDS curves σ have moved to TM. In particular, $\sigma(t) \in TM$ now contains both robot positions and velocities, so $\ddot{\sigma}(t)$ contains both acceleration and jerk. The GDS tackles this problem by projecting the geometric acceleration $F^* D_\sigma \gamma = a^i \partial_i^v + \kappa^i \partial_i^a$ onto the first half of the basis vectors ∂_i^v to remove components that would correspond to jerk. Unfortunately, this is a projection onto a horizontal bundle, and horizontal bundles are necessarily dependent on the choice of chart. In other words, it is impossible to define this projection globally.

In particular, a clear problem arises when considering a change of coordinates on the projected Riemannian task metric for the GDS. Consider a metric g on TN which can be represented in chart $\tilde{C}$ as $\tilde{g}^v_{ij} d\tilde{v}^i d\tilde{v}^j + \tilde{g}^a_{ij} d\tilde{a}^i d\tilde{a}^j$, so that $\tilde{\boldsymbol{g}} = \mathrm{blockdiag}(\tilde{\boldsymbol{g}}^v, \tilde{\boldsymbol{g}}^a)$. Let ϕ be the transition function from a new chart $\hat{C}$ to $\tilde{C}$. The Jacobian of ϕ has the form

$$\boldsymbol{J}\boldsymbol{\phi}(\hat{p},\hat{v}) = \begin{bmatrix} \boldsymbol{J}\boldsymbol{\phi}^v(\hat{p},\hat{v}) & 0 \\ \boldsymbol{J}\boldsymbol{\phi}^{av}(\hat{p},\hat{v}) & \boldsymbol{J}\boldsymbol{\phi}^a(\hat{p},\hat{v}) \end{bmatrix},$$

$$J\phi^v_{ij}(\hat{p},\hat{v}) = J\phi^a_{ij}(\hat{p},\hat{v}) = \frac{\partial \tilde{p}^j}{\partial \hat{p}^i}(\hat{p}),$$

$$J\phi^{av}_{ij}(\hat{p},\hat{v}) = \hat{v}^k \frac{\partial^2 \tilde{p}^j}{\partial \hat{p}^i \partial \hat{p}^k}(\hat{p}).$$

The metric coordinates in $\hat{C}$ can then be found by

$$\begin{aligned} \hat{\boldsymbol{g}} &= \boldsymbol{J}\boldsymbol{\phi}^\top \tilde{\boldsymbol{g}} \boldsymbol{J}\boldsymbol{\phi} \\ &= \begin{bmatrix} \boldsymbol{J}\boldsymbol{\phi}^{v\top}\tilde{\boldsymbol{g}}^v\boldsymbol{J}\boldsymbol{\phi}^v + \boldsymbol{J}\boldsymbol{\phi}^{av\top}\tilde{\boldsymbol{g}}^a\boldsymbol{J}\boldsymbol{\phi}^{av} & \boldsymbol{J}\boldsymbol{\phi}^{av\top}\tilde{\boldsymbol{g}}^a\boldsymbol{J}\boldsymbol{\phi}^a \\ \boldsymbol{J}\boldsymbol{\phi}^{a\top}\tilde{\boldsymbol{g}}^a\boldsymbol{J}\boldsymbol{\phi}^{av} & \boldsymbol{J}\boldsymbol{\phi}^{a\top}\tilde{\boldsymbol{g}}^a\boldsymbol{J}\boldsymbol{\phi}^a \end{bmatrix} \end{aligned}$$

so that $\hat{\boldsymbol{g}}^v = \boldsymbol{J}\boldsymbol{\phi}^{v\top}\tilde{\boldsymbol{g}}^v\boldsymbol{J}\boldsymbol{\phi}^v + \boldsymbol{J}\boldsymbol{\phi}^{av\top}\tilde{\boldsymbol{g}}^a\boldsymbol{J}\boldsymbol{\phi}^{av}$. Since $\tilde{\boldsymbol{g}}^a \neq \boldsymbol{0}$

by the properties of a metric, this shows that a projection onto g^v cannot be maintained through coordinate transformations. Since the projected geometric acceleration depends on g^v, the policy will change depending on the choice of coordinates and will thus exhibit unnatural behavior (e.g., undesirable geodesics) on non-Euclidean manifolds.

To make this concrete, consider our Table I example on a sphere, omitting obstacle avoidance. To see the effect of coordinate choice, we run both PBDS and GDS policies using different schemes for choosing coordinate charts on $\mathbb{S}^2$. The resulting trajectories can be see in Fig. 5a and b. Here, the suboptimal and inconsistent behavior of the GDS policy demonstrates the importance of geometric consistency. With such consistency, the policy can accurately capture and leverage the geometry of the manifold, as shown in the PBDS example. Without it, the policy instead introduces artifacts that disturb natural motion on the manifold and can lead to erratic behavior. In some cases it is possible for the GDS policy designer to engineer tasks within a given coordinate choice that suppresses such behavior, but for non-Euclidean metrics this will often be a struggle against the locally-defined curvature terms introduced by the GDS formalism.

B. Task Prioritization and Velocity-Dependent Metrics

The main motivation for building GDSs on tangent bundles is to define velocity-dependent metrics on task tangent bundles which can be used both to drive the GDS dynamics and enable velocity-dependent prioritization of tasks in the RMP framework. However, this introduces difficulty for the designer in forming task metrics which both give correct single-task behavior and weight each task correctly against all other tasks. Indeed, rather than allowing incremental design and modular combination of tasks, it can require jointly redesigning tasks for every new task combination.

To resolve this issue, the PBDS framework leverages two observations: (O1) There are few practical scenarios where a velocity-dependent Riemannian metric is useful for driving task dynamics, and (O2) there is no need to use a single metric for both task dynamics and task prioritization.

We can gain intuition for (O1) by considering a modified PBDS applied to tangent bundle tasks. For brevity, we outline the main modifications here and leave further details to Appx. D. Let $F_i : TM \longrightarrow TN_i$ be our task maps. To avoid projecting onto the horizontal bundle as in the GDS formulation, we instead wish to project onto the vertical bundle, an operation that is globally well-defined. However, to continue recovering an acceleration policy rather than a jerk policy, we consider permuted task maps $\hat{F}_i \triangleq \sigma_{TN_i} \circ F_i : TM \longrightarrow \mathbb{R}^{2d_i}$ defined locally as $\hat{F}_i(p, v) = ((d\bar{f}_i)_p(v), \bar{f}_i(p))$. This essentially swaps the position and velocity coordinates, an operation which is globally well-defined on parallelizable manifolds (e.g., Lie groups) having an embedding $\bar{\varphi}_i : N_i \longrightarrow \mathbb{R}^{d_i}$, giving $\bar{f}_i = \bar{\varphi}_i \circ f_i$. Then by considering curves in the corresponding pullback bundles satisfying a system analogous to (5) with the appropriate global projections applied, we arrive at the following ODE for a curve σ in configuration manifold M:

$$\ddot{\boldsymbol{\sigma}} = \left(\textstyle\sum_i \boldsymbol{\mathcal{C}}_i^\top \boldsymbol{w}_i^a \boldsymbol{\mathcal{C}}_i\right)^\dagger \left(\textstyle\sum_i \boldsymbol{\mathcal{C}}_i^\top \boldsymbol{w}_i^a \boldsymbol{\mathcal{A}}_i\right), \quad \boldsymbol{\mathcal{C}}_i = \boldsymbol{J}\bar{\boldsymbol{f}}_i + \boldsymbol{\Xi}_i^v$$

$$\boldsymbol{\mathcal{A}}_{\boldsymbol{i}} = (\boldsymbol{g}_i^{-1})^a(\boldsymbol{\mathcal{F}}_{D,i} - \nabla\boldsymbol{\Phi}_i) - (\dot{\boldsymbol{J}}\bar{\boldsymbol{f}}_i - \boldsymbol{\Xi}_i^a)\dot{\boldsymbol{\sigma}}$$

$$\Xi_{ij}^v = J\hat{F}_{kj}\Gamma_{k(h+m)}^{i+d} J\bar{f}_{h\ell}\dot{\sigma}^\ell, \; \Xi_{ij}^a = J\hat{F}_{k(j+d)}\Gamma_{k(h+m)}^{i+d} J\bar{f}_{h\ell}\dot{\sigma}^\ell.$$

It is now instructive to see the corresponding local policy equations for a set of uniformly weighted $f_i = \mathrm{Id}_{\mathbb{R}}$ tasks:

$$\ddot{\sigma} = \frac{\sum_i (1 + \Xi_i^a)((g_i^a)^{-1}(\mathcal{F}_{D,i} - \partial_{x_i}\Phi_i) - \Xi_i^v\dot{\sigma})}{\sum_i (1 + \Xi_i^a)^2}$$

$$\Xi_i^v = \tfrac{1}{2}(g_i^a)^{-1}\partial_{x_i} g_i^a \dot{\sigma}, \quad \Xi_i^a = \tfrac{1}{2}(g_i^a)^{-1}\partial_{v_i} g_i^a \dot{\sigma}.$$

This shows that a velocity-dependent metric modifies both the "force" of the task (its contribution to the numerator of the policy equation) and the total "mass" of the system (the denominator). However, while modifying a task force is desirable, modifying the total mass is not.

For example, consider using a velocity-dependent metric to enforce constraints. For position constraints, if a task metric increases as the robot approaches the constraint with velocity towards it, the total mass increases. This indeed makes it hard for the robot to accelerate further towards the constraint, but likewise it becomes hard for the robot to accelerate away. Similarly for velocity constraints, a barrier-type velocity metric can enforce velocity limits, but it then becomes difficult for other tasks to remove the high inertia of the robot near the velocity limit.

Thus rather than applying the PBDS framework to tasks mapping from the robot configuration tangent bundle TM, we instead nominally consider tasks mapping from the base robot manifold M (i.e., using only position-dependent metrics to design individual task behaviors), greatly simplifying the framework with little practical loss in expressiveness of tasks. Additionally, leveraging (O2), we lose no expressiveness in defining velocity-dependent prioritization of tasks by defining separate pseudometrics on the TN_i to perform the weighting. This also removes the difficulty of achieving desired task behavior and task prioritization with a single metric, thus resulting in a more modular framework.

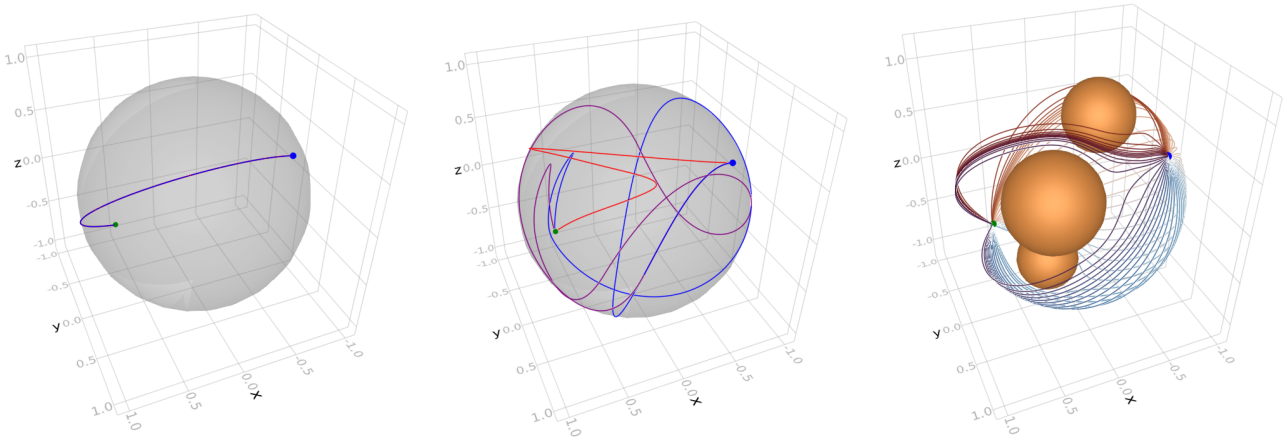
C. Metric-based Constraints

As mentioned, constraints imposed solely by Riemannian metrics rather than repulsive potentials can eliminate the spurious local minima in combined potential fields characteristic of many APF methods. However, this has not been considered in works employing the GDS/RMP framework [21], [24], [25]. The notion of constraint-enforcing Riemannian metrics was recently considered in [18]. However, rather than constructing such metrics explicitly, the approach numerically computes a distance field from the goal considering obstacles, interpreting this as a geodesic distance field whose gradient is then used to guide trajectory optimization. Such a technique is computationally intensive and may not be practical for complex, dynamic scenarios.

Instead, we propose a class of simple analytical Riemannian behavior metrics that provide tunable enforcement of constraints within the PBDS framework. To motivate the form of these metrics, consider again a set of $f_i : \mathbb{R} \longrightarrow \mathbb{R}$ tasks, this time using the PBDS motion policy of (6):

$$\ddot{\sigma} = \frac{\sum_i J f_i w_i^a((g_i)^{-1}(\mathcal{F}_{D,i} - \partial_{x_i}\Phi_i) - (\dot{J}f_i + \Xi_i)\dot{\sigma})}{\sum_i J f_i^2 w_i^a}.$$

For constraint tasks, we let f_i be a distance function to the constraint boundary, so that $x_i \in \mathbb{R}_+$ is the distance to the constraint and $\Xi_i = \frac{1}{2}(g_i)^{-1}\partial_{x_i} g_i \dot{\sigma}$. Since we add no repulsive potential and offload dissipative forces to other tasks, constraints are enforced through Ξ_i. In short, designing Ξ_i as a negative barrier function at the constraint boundary will slow negative velocities as desired. However, due to the requirements of g_i and the form of Ξ_i, a simple logarithmic



(a) PBDS geometric consistency test (b) GDS/RMP geometric consistency test (c) PBDS attractor and obstacle avoidance policy on $\mathbb{S}^2$

Fig. 5: (a) and (b) show a geometric consistency test using a point attractor on $\mathbb{S}^2$ and three schemes for choice of chart: (Red) Always using south pole stereographic projection, (Blue) Always using north pole stereographic projection. (Purple) Switching stereographic projections depending on the sphere hemisphere. The blue and green dots show the start and goal points, respectively. Note that using PBDS, all coordinate choices result in the same trajectory. (c) shows resulting trajectories from a PBDS attractor policy with obstacles added, where the curve colors represent different starting velocity directions.

or inverse barrier g_i is insufficient. In particular, $g_i = \log x_i$ is negative for $x_i < 1$, and $g_i = a/x_i^b$ always results in $\Xi_i = -b/x_i$ for all $a, b \in \mathbb{R}$, which has minimal tuning capability (i.e. a/x_i^b cannot decrease fast enough with decreasing x_i). There are many suitable options, but a good candidate is

$$g_i(x_i) = \exp(a/(bx_i^b))$$

for $a > 0$, $b > 1$, which results in a familiar and flexible barrier function $\Xi_i = -a/x_i^{b-1}$.

Additionally, we must weight constraint tasks such that they are only active when $\dot{x}_i < 0$, both to save computation and because such metric-based constraints can produce an undesired acceleration away from the constraint when $\dot{x}_i > 0$. This can be accomplished using a pseudometric

$$w_i^a(x_i, \dot{x}_i) = \begin{cases} 1 & \text{if } \dot{x}_i > 0 \text{ and } x_i < \beta, \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

where $\beta \in \mathbb{R}_+$ activates the constraint avoidance within some proximity. This works well numerically, but smooth approximations can be used to retain smoothness guarantees.

The effectiveness of this metric and pseudometric combination can be seen in our full Table I running example with results shown in Fig. 5c, where each obstacle has a task with a task map f_i giving the Euclidean distance to the obstacle in the ambient $\mathbb{R}^3$, metric $g_i = \exp(1/x_i^2)$, and weighting pseudometric w_i as defined in (8) with $\beta = \infty$.

V. Implementation and Arm Experiments

We implemented the PBDS framework and the GDS/RMP framework for comparison in a fast Julia package called PBDS.jl[2]. Despite the apparent complexity of its geometric formulation, the PBDS algorithm in implementation is quite simple and is summarized in Alg. 1. The main challenge in handling non-Euclidean manifolds within the PBDS framework is transitioning coordinate representations of the necessary objects between different coordinate charts and embeddings. However, PBDS.jl is designed to handle these transitions automatically. To increase performance in complex tasks, PBDS.jl also implements a computational tree inspired by [17] for cases where a task map tree structure such as in Fig. 3 allows significant reuse of computation. This requires careful segmenting and recombination of the main equations (6) and (7) (details in Appx. B).

[2]Available at `https://github.com/StanfordASL/PBDS.jl`

Algorithm 1: Multi-Task PBDS Policy

1: **Input:** Robot position and velocity $(\sigma, \dot{\sigma})$
2: **Output:** Robot acceleration command $\ddot{\sigma}$
3: **Data:** Task PBDSs $(f_i, g_i, \Phi_i, \mathcal{F}_{D,i})$, weights w_i
4: Compute $\mathcal{A}_i$ using (7) for each task
5: Compute combined acceleration policy $\ddot{\sigma}$ using (6)

For demonstration of PBDS on a complex robotic task, we considered a 7-DoF Franka Panda arm grasping a mug in a dynamic, cluttered environment (see video at bit.ly/pbds-vid). For mechanism modelling and visualization, we used packages from JuliaRobotics [26]. For the PBDS policy, we used attractor, obstacle avoidance, and joint-limit policies on Euclidean task manifolds and considered damping on $\mathbb{S}^2$ and $\mathbb{S}^1$ for gripper location around the mug and along the mug rim. Along with providing natural movement through obstacles, the fast Julia implementation allowed the robot to react quickly to changes in the environment by computing policy outputs at a rapid 300-500 Hz, a speed which could be accelerated further by exploiting the clear opportunities for parallelization offered by the PBDS framework.

VI. Conclusions

In conclusion, we have provided a fast, easy-to-use, and geometrically consistent framework for generating motion policies on non-Euclidean robot and task spaces. We have also shown how PBDS can leverage Riemannian metrics for simple constraint enforcement without generating undesirable potential function local minima. We highlight that although designing tasks on non-Euclidean manifolds can appear daunting, in practice the design is often trivial (Table I), and part of the strength of PBDS lies in its ability to seamlessly integrate different tasks that are designed in isolation. We also stress that a rich, diverse set of complex behaviors can be produced from combinations of a handful of common task design patterns. Future work for PBDS includes extensions to a broader class of constraints (e.g., velocity, acceleration, and control limits) and to tasks defined on discrete structures such as manifold triangle meshes.

REFERENCES

[1] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proc. IEEE Conf. on Robotics and Automation*, 1985.

[2] P. Khosla and R. Volpe, "Superquadric artificial potentials for obstacle avoidance and approach," in *Proc. IEEE Conf. on Robotics and Automation*, 1988.

[3] N. Hogan, "Impedance control: An approach to manipulation," in *American Control Conference*, 1984.

[4] S. M. Khansari-Zadeh and A. Billard, "A dynamical system approach to realtime obstacle avoidance," *Autonomous Robots*, vol. 32, no. 4, pp. 433–454, 2012.

[5] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical Movement Primitives: Learning attractor models for motor behaviors," *Neural Computation*, vol. 25, no. 2, pp. 328–373, 2013.

[6] N. Ratliff, M. Toussaint, and S. Schaal, "Understanding the geometry of workspace obstacles in motion optimization," in *Proc. IEEE Conf. on Robotics and Automation*, 2015.

[7] J. Agirrebeitia, R. Avilés, I. F. de Bustos, and G. Ajuria, "A new APF strategy for path planning in environments with obstacles," *Mechanism and Machine Theory*, vol. 40, no. 6, pp. 645–658, 2005.

[8] J. Schulman, J. Ho, A. Lee, I. Awwal, H. Bradlow, and P. Abbeel, "Finding locally optimal, collision-free trajectories with sequential convex optimization," in *Robotics: Science and Systems*, 2013.

[9] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the Atlas humanoid robot," *Autonomous Robots*, vol. 40, no. 3, pp. 429–455, 2015.

[10] W. Merkt, V. Ivan, and S. Vijaykumar, "Continuous-time collision avoidance for trajectory optimization in dynamic environments," in *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, 2019.

[11] E. Rimon and D. E. Koditschek, "Exact robot navigation wsing artificial potentia functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.

[12] J.-O. Kim and P. K. Khosla, "Real-time obstacle avoidance using harmonic potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 338–349, 1992.

[13] L. Huber, A. Billard, and J.-J. Slotine, "Avoidance of convex and concave obstacles with convergence ensured through contraction," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1462–1469, 2019.

[14] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Proc. IEEE Conf. on Robotics and Automation*, 1991.

[15] D. H. Kim, "Escaping route method for a trap situation in local path planning," *Int. Journal of Control, Automation and Systems*, vol. 7, no. 3, pp. 495–500, 2009.

[16] J. M. Lee, *Introduction to Riemannian Manifolds*, 2nd ed. Springer, 2018.

[17] C.-A. Cheng, M. Mukadam, J. Issac, S. Birchfield, D. Fox, B. Boots, and N. Ratliff, "RMPflow: A computational graph for automatic motion policy generation," in *Workshop on Algorithmic Foundations of Robotics*, 2018.

[18] J. Mainprice, N. Ratliff, M. Toussaint, and S. Schaal. (2020) An interior point method solving motion planning problems with narrow passages. Available at https://arxiv.org/abs/2007.04842.

[19] N. D. Ratliff, K. Van Wyk, M. Xie, A. Li, and M. A. Rana. (2020) Optimization fabrics. Available at https://arxiv.org/abs/2008.02399.

[20] N. D. Ratliff, J. Issac, D. Kappler, S. Birchfield, and D. Fox. (2018) Riemannian Motion Policies. Available at https://arxiv.org/abs/1601.04037.

[21] M. A. Rana, A. Li, H. Ravichandar, M. Mukadam, S. Chernova, D. Fox, B. Boots, and N. Ratliff, "Learning reactive motion policies in multiple task spaces from human demonstrations," in *Conf. on Robot Learning*, 2020.

[22] J. M. Lee, *Introduction to Smooth Manifolds*, 2nd ed. Springer, 2013.

[23] F. Bullo and A. D. Lewis, *Geometric Control of Mechanical Systems*. Springer-Verlag, 2004.

[24] A. Li, C.-A. Cheng, B. Boots, and M. Egerstedt, "Stable, concurrent controller composition for multi-objective robotic tasks," in *Proc. IEEE Conf. on Decision and Control*, 2019, Available at https://arxiv.org/abs/1903.12605.

[25] M. Mukadam, C.-A. Cheng, D. Fox, B. Boots, and N. Ratliff, "Riemannian Motion Policy fusion through learnable Lyapunov function reshaping," in *Conf. on Robot Learning*, 2020.

[26] T. Koolen and R. Deits, "Julia for robotics: Simulation and real-time control in a high-level programming language," in *Proc. IEEE Conf. on Robotics and Automation*, 2019.

[27] M. W. Hirsch, *Differential Topology*. Springer, 1976.

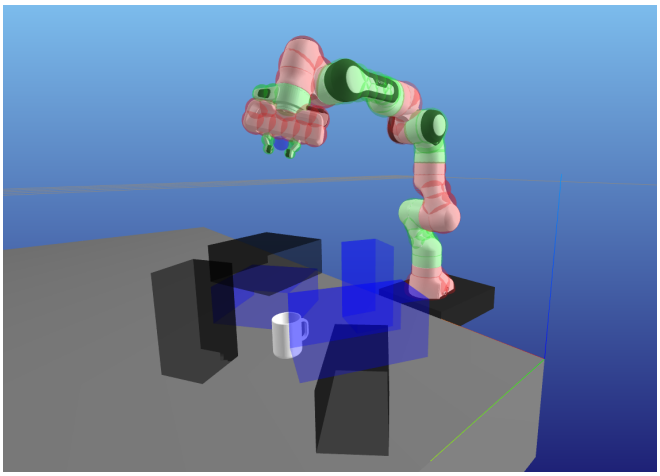


Fig. 6: Example experimental scenario for Franka Panda arm reaching through clutter to grasp a mug. The blue and black boxes are obstacles, and the green and red spheres on the arm represent the collisions for different robot links.

APPENDIX A
FURTHER EXPERIMENTAL DETAILS

In this section, we provide further details of the design of the PBDS policy driving our robot arm grasping experiments, including details of task design on $\mathbb{S}^1$ and $\mathbb{S}^2$, which may serve as a useful reference for practitioners.

Since all seven joints of the Franka Panda arm have joint limits, the robot configuration manifold was $M = \mathbb{R}^7$ (note that for revolute joints without joint limits, the circle $\mathbb{S}^1$ should be used instead). For collision avoidance, we modelled a conservative collision hull around each link using a set of spheres, as shown in Fig. 6.

I. Task Set for Grasping a Mug

The full list of tasks for this PBDS policy and their corresponding task spaces is as follows.

Damping:

- $\mathbb{R}$: Robot arm joint damping
- $\mathbb{S}^2 \times \mathbb{R}$: Distance-toggled damping of gripper around mug
- $\mathbb{S}^1 \times \mathbb{R}$: Distance-toggled damping of gripper along mug rim

Constraints:

- $\mathbb{R}_+$: Joint limits (one for each joint)
- $\mathbb{R}_+$: Obstacle avoidance (one for each pair of obstacle and link collision hull sphere)
- $\mathbb{R}_+$: Self-collision avoidance (one for each pair of link collision hull spheres having the potential to collide)

Attractors:

- $\mathbb{R}^2$: Distance-toggled mug attractor
- $\mathbb{R}^2$: Distance-toggled gripper axis alignment attractor
- $\mathbb{R}^2$: Distance-toggled grasping angle alignment attractor
- $\mathbb{R}^2$: Distance-toggled grasp completion attractor

Note that although we have multiple potentials, one for each attractor task, they are easily designed to be always non-conflicting (i.e. the inner product of their resulting desired robot accelerations is never negative), and thus they produce no undesirable local minima.

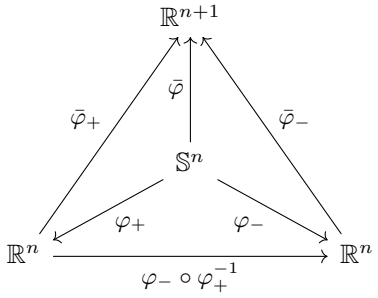


Fig. 7: Commutative diagram for the two stereographic charts of $\mathbb{S}^2$ and its embedding in Euclidean space. (Here we omit the subsets $U_+, U_- \subset \mathbb{S}^2$ which are the domains of the respective chart maps φ_+, φ_-).

II. Tasks on $\mathbb{S}^n$

For designing tasks on $\mathbb{S}^n$ for some n, it is often natural to consider $\mathbb{S}^2$ as being embedded in an ambient Euclidean space and design the necessary components $(g, \mathcal{F}_D, \Phi, w)$ there, as we also do in the example in Table I. These components in the ambient space then induce corresponding components on $\mathbb{S}^n$. Thus, it is important to be able to compute the induced versions of these components. The same reasoning applies to other manifolds which have a natural embedding in Euclidean space, but here we will consider $\mathbb{S}^n$ as it is a good example used in both the grasping and Table I scenarios.

The n-sphere $\mathbb{S}^n$ can be naturally embedded as the unit sphere in $\mathbb{R}^{n+1}$, using an embedding $\bar{\varphi} : \mathbb{S}^n \longrightarrow \mathbb{R}^{n+1}$. We can consider two stereographic projection charts on $\mathbb{S}^n$ with chart maps $\varphi_+ : U_+ \subset \mathbb{S}^n \longrightarrow \mathbb{R}^n$ and $\varphi_- : U_- \subset \mathbb{S}^n \longrightarrow \mathbb{R}^n$ representing the south and north pole charts, respectively. Then we can define corresponding maps $\bar{\varphi}_+$ and $\bar{\varphi}_-$ from chart coordinates to embedded coordinates as follows:

$$\bar{\varphi}_+^i(x) = \bar{\varphi}_-^i(x) = \frac{2x^i}{a}, \; a = \sum_{i=1}^{n} 1 + (x^i)^2$$
$$\bar{\varphi}_+^{n+1}(x) = -\bar{\varphi}_-^{n+1}(x) = \frac{2-a}{a}.$$

There is also a chart transition map between the chart coordinate representations defined by $\varphi_+ \circ \varphi_-^{-1}(x) = \varphi_- \circ \varphi_+^{-1}(x) = x/\|x\|_2^2$. These maps and the associated spaces are summarized in Fig. 7. We can construct similar maps for $T\mathbb{S}^2$ embedded in $\mathbb{R}^{2n+2}$.

Now given a behavior metric $\bar{g}$, dissipative forces $\bar{\mathcal{F}}_D$, and potential function Φ defined on $\mathbb{R}^{n+1}$ and weighting pseudometric w defined on defined on $\mathbb{R}^{2n+2}$, the pullbacks of these objects through $\bar{\varphi}_+$ and $\bar{\varphi}_-$ give their coordinate representations in the south and north pole charts, respectively. In particular, we can compute the relevant components in the south pole chart by:

$$\boldsymbol{g}_+ = \boldsymbol{J}\bar{\boldsymbol{\varphi}}_+^\top \bar{\boldsymbol{g}} \boldsymbol{J}\bar{\boldsymbol{\varphi}}_+, \quad \boldsymbol{\mathcal{F}}_{D+} = \boldsymbol{J}\bar{\boldsymbol{\varphi}}_+^\top \bar{\boldsymbol{\mathcal{F}}}_D$$
$$\boldsymbol{\Phi}_+ = \bar{\boldsymbol{\Phi}} \circ \bar{\boldsymbol{\varphi}}_+, \quad \boldsymbol{w}_+^a = \boldsymbol{J}\bar{\boldsymbol{\varphi}}_+^\top \bar{\boldsymbol{w}}^a \boldsymbol{J}\bar{\boldsymbol{\varphi}}_+.$$

Components in the north pole chart representation can be computed similarly.

III. Distance-Toggled Tasks

In some scenarios it is useful to switch tasks on or off based on some distance using the weighting pseudometrics. We have already seen an example of this in Sec. IV-C for constraint tasks, but as in the grasping scenario, it can also be useful for attractor and damping tasks.

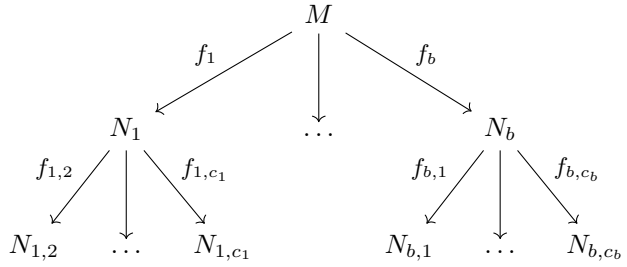


Fig. 8: Task map tree for an example multi-task PBDS.

To do this, one can simply form a new task map which is the product of the original task map and the distance function used for toggling. Thus, the new task manifold becomes the product of the original task manifold and $\mathbb{R}$, i.e. $\mathbb{R}$ becomes $\mathbb{R}^2$ and $\mathbb{S}$ becomes $\mathbb{S}^2 \times \mathbb{R}$. (Note that in PBDS.jl, we have implemented product task maps to conveniently handle heterogeneous manifolds like $\mathbb{S}^2 \times \mathbb{R}$). Given, one can form weights w for the task which are a function of the new task variable to toggle the task as desired.

APPENDIX B
COMPUTATIONAL TREE FOR MULTI-TASK PBDS

Here we give a derivation and details for implementation of the PBDS computational tree for reusing computation in multi-level trees of task maps. Consider a set of task maps forming a task map tree such as in Fig. 3. In such a case, multiple task maps are composite maps which share components (e.g., $f_{1,1} \circ f_1$ and $f_{1,2} \circ f_1$ share f_1). Thus in cases where applying component maps and computing Jacobians and Jacobian derivatives is expensive, it is useful to exploit this tree structure to reuse computation while computing the multi-task PBDS acceleration policy output.

Thus we propose quantities denoted K, A, B, $\mathcal{F}$, and ξ which can be numerically pulled up the tree from the leaves and combined into equation (6). Noting that each leaf corresponds to a different task, we initialize these quantities for each leaf as follows.

Leaf task manifold nodes:

$$\boldsymbol{K} = \boldsymbol{B} = \boldsymbol{J}\boldsymbol{f}^\top \boldsymbol{w}^a \boldsymbol{J}\boldsymbol{f}, \quad \boldsymbol{A} = \boldsymbol{J}\boldsymbol{f}^\top \boldsymbol{w}^a \dot{\boldsymbol{J}}\boldsymbol{f}$$
$$\boldsymbol{\mathcal{F}} = \boldsymbol{J}\boldsymbol{f}^\top \boldsymbol{w}^a \boldsymbol{g}^{-1}(\boldsymbol{\mathcal{F}}_D - \nabla \boldsymbol{\Phi})$$
$$\xi_{sr}^q = Jf_{\tau q} w_{\tau\eta}^a Jf_{\alpha s} \Gamma_{\alpha\beta}^\eta Jf_{\beta r},$$

where f is the map to the task manifold from the immediate parent, and we locally compute the task metric g, Christoffel symbols Γ, generalized force $\mathcal{F}_D$, potential gradient $\nabla\Phi$, and block $\boldsymbol{w}^a$ of the weighting pseudometric w as usual. Next, for parent nodes representing intermediate manifolds in the composite task maps, we combine the child quantities as follows.

Intermediate manifold nodes:

$$\boldsymbol{K} = \boldsymbol{J}\boldsymbol{f}^\top \left(\sum \boldsymbol{K}_c\right) \boldsymbol{J}\boldsymbol{f}, \quad \boldsymbol{B} = \boldsymbol{J}\boldsymbol{f}^\top \left(\sum \boldsymbol{B}_c\right) \boldsymbol{J}\boldsymbol{f}$$
$$\boldsymbol{A} = \boldsymbol{J}\boldsymbol{f}^\top \left(\left(\sum \boldsymbol{A}_c\right) \boldsymbol{J}\boldsymbol{f} + \left(\sum \boldsymbol{B}_c\right) \dot{\boldsymbol{J}}\boldsymbol{f}\right)$$
$$\boldsymbol{\mathcal{F}} = \boldsymbol{J}\boldsymbol{f}^\top \sum \boldsymbol{\mathcal{F}}_c, \quad \xi_{h\ell}^k = Jf_{qk} Jf_{sh} \left(\sum \xi_c\right)_{sr}^q Jf_{r\ell},$$

where the c subscripts indicating quantities from the child nodes, summations are over all child nodes, and f again denotes the map to the node from the immediate parent. Finally at the root node containing the robot configuration manifold, we combine as follows.

Root robot manifold node:

$$\boldsymbol{K} = \sum \boldsymbol{K}_c, \quad \boldsymbol{A} = \sum \boldsymbol{A}_c, \quad \boldsymbol{\mathcal{F}} = \sum \boldsymbol{\mathcal{F}}_c$$

$$\xi_{k\ell} = \Big(\sum \xi_c\Big)^k_{h\ell} \dot{\sigma}^h.$$

We can now compute the multi-task PBDS acceleration policy output using the root node quantities as

$$\ddot{\boldsymbol{\sigma}} = \boldsymbol{K}^\dagger(\boldsymbol{\mathcal{F}} - (\boldsymbol{A} + \boldsymbol{\xi})\dot{\boldsymbol{\sigma}}). \tag{9}$$

We now demonstrate that this strategy of splitting and reusing computation correctly reconstructs (6).

Lemma B.1 (Computational Tree)**.** *Given a multi-task PBDS and a corresponding tree of task maps, computing K, A, $\mathcal{F}$, and ξ at the root node gives*

$$\boldsymbol{K}^\dagger(\boldsymbol{\mathcal{F}} - (\boldsymbol{A} + \boldsymbol{\xi})\dot{\boldsymbol{\sigma}}) = \left(\sum_i \boldsymbol{Jf}_i^\top \boldsymbol{w}_i^a \boldsymbol{Jf}_i\right)^\dagger \left(\sum_i \boldsymbol{Jf}_i^\top \boldsymbol{w}_i^a \boldsymbol{\mathcal{A}}_i\right)$$

Proof. Consider the two-level task tree in Fig. 8, having b intermediate manifold nodes and c_i children for the ith intermediate node, for a total of $\sum_{i=1}^b c_i$ leaf nodes and corresponding task maps. For such a multi-task PBDS, we can expand (6) as

$$\ddot{\boldsymbol{\sigma}} = \left(\sum_{i=1}^b \sum_{j=1}^{c_i} \boldsymbol{Jf}_i^\top \boldsymbol{Jf}_{i,j}^\top \boldsymbol{w}_{i,j}^a \boldsymbol{Jf}_{i,j} \boldsymbol{Jf}_i\right)^\dagger \left(\sum_{i=1}^b \sum_{j=1}^{c_i} \boldsymbol{Jf}_i^\top \boldsymbol{Jf}_{i,j}^\top \boldsymbol{w}_{i,j}^a \left(\boldsymbol{g}_{i,j}^{-1}\Big(\boldsymbol{\mathcal{F}}_{D,i,j} - \nabla\Phi_{i,j}\Big) - \left(\frac{d}{dt}\Big(\boldsymbol{Jf}_{i,j}\boldsymbol{Jf}_i\Big) + \boldsymbol{\Xi}_{i,j}\right)\dot{\boldsymbol{\sigma}}\right)\right),$$

where we use the i subscript to denote quantities associated to the intermediate nodes and use the i,j subscript for those associated to the leaf nodes and their corresponding tasks.

Now unpacking each of the terms in (9) at the root robot manifold node, we have

$$\boldsymbol{K} = \sum_{i=1}^b \boldsymbol{K}_i = \sum_{i=1}^b \boldsymbol{Jf}_i^\top \left(\sum_{j=1}^{c_i} \boldsymbol{K}_{i,j}\right) \boldsymbol{Jf}_i = \sum_{i=1}^b \sum_{j=1}^{c_i} \boldsymbol{Jf}_i^\top \boldsymbol{Jf}_{i,j}^\top \boldsymbol{w}_{i,j}^a \boldsymbol{Jf}_{i,j} \boldsymbol{Jf}_i,$$

$$\begin{aligned}\boldsymbol{A} &= \sum_{i=1}^b \boldsymbol{A}_i \\ &= \sum_{i=1}^b \boldsymbol{Jf}_i^\top \left(\left(\sum_{j=1}^{c_i} \boldsymbol{A}_{i,j}\right)\dot{\boldsymbol{Jf}}_i + \left(\sum_{j=1}^{c_i} \boldsymbol{B}_{i,j}\right)\boldsymbol{Jf}_i\right) \\ &= \sum_{i=1}^b \sum_{j=1}^{c_i} \boldsymbol{Jf}_i^\top \left(\boldsymbol{Jf}_{i,j}^\top \boldsymbol{w}_{i,j}^a \boldsymbol{Jf}_{i,j}\dot{\boldsymbol{Jf}}_i + \boldsymbol{Jf}_i^\top \boldsymbol{w}_{i,j}^a \dot{\boldsymbol{Jf}}_{i,j}\boldsymbol{Jf}_i\right) \\ &= \sum_{i=1}^b \sum_{j=1}^{c_i} \boldsymbol{Jf}_i^\top \boldsymbol{Jf}_{i,j}^\top \boldsymbol{w}_{i,j}^a \frac{d}{dt}\Big(\boldsymbol{Jf}_{i,j}\boldsymbol{Jf}_i\Big),\end{aligned}$$

$$\begin{aligned}\boldsymbol{\mathcal{F}} &= \sum_{i=1}^b \boldsymbol{\mathcal{F}}_i = \sum_{i=1}^b \boldsymbol{Jf}_i^\top \left(\sum_{j=1}^{c_i} \mathcal{F}_{i,j}\right) \\ &= \sum_{i=1}^b \sum_{j=1}^{c_i} \boldsymbol{Jf}_i^\top \boldsymbol{Jf}_{i,j}^\top \boldsymbol{w}_{i,j}^a \boldsymbol{g}_{i,j}^{-1}\big(\boldsymbol{\mathcal{F}}_{D,i,j} - \nabla\Phi_{i,j}\big),\end{aligned}$$

$$\begin{aligned}\xi_{k\ell} &= \left(\sum_{i=1}^b \xi_i\right)^k_{h\ell} \dot{\sigma}^h \\ &= \sum_{i=1}^b (Jf_i)_{qk}(Jf_i)_{sh}\left(\sum_{j=1}^{c_i}\xi_{i,j}\right)^q_{sr}(Jf_i)_{r\ell}\dot{\sigma}^h \\ &= \sum_{i=1}^b \sum_{j=1}^{c_i} (Jf_i)_{qk}(Jf_{i,j})_{\tau q}(w_{ij}^a)_{\tau\eta}(Jf_i)_{sh}(Jf_{i,j})_{\alpha s} \\ &\quad (\Gamma_{i,j})^\eta_{\alpha\beta}(Jf_{i,j})_{\beta r}(Jf_i)_{r\ell}\dot{\sigma}^h \\ &= \sum_{i=1}^b \sum_{j=1}^{c_i} \big(\boldsymbol{Jf}_i^\top \boldsymbol{Jf}_{i,j}^\top \boldsymbol{w}_{i,j}^a\big)_{k\eta}(Jf_{i,j}Jf_i)_{\alpha h} \\ &\quad (\Gamma_{i,j})^\eta_{\alpha\beta}(Jf_{i,j}Jf_i)_{\beta\ell}\dot{\sigma}^h \\ &= \sum_{i=1}^b \sum_{j=1}^{c_i} \big(\boldsymbol{Jf}_i^\top \boldsymbol{Jf}_{i,j}^\top \boldsymbol{w}_{i,j}^a\big)_{k\eta} J(f_{i,j}\circ f_i)_{\alpha h} \\ &\quad (\Gamma_{i,j})^\eta_{\alpha\beta} J(f_{i,j}\circ f_i)_{\beta\ell}\dot{\sigma}^h \\ &= \sum_{i=1}^b \sum_{j=1}^{c_i} \big(\boldsymbol{Jf}_i^\top \boldsymbol{Jf}_{i,j}^\top \boldsymbol{w}_{i,j}^a\big)_{k\eta}(\Xi_{i,j})_{\eta h}\dot{\sigma}^h \\ &= \sum_{i=1}^b \sum_{j=1}^{c_i} \boldsymbol{Jf}_i^\top \boldsymbol{Jf}_{i,j}^\top \boldsymbol{w}_{i,j}^a \boldsymbol{\Xi}_{i,j}\dot{\boldsymbol{\sigma}}.\end{aligned}$$

Combining these in (9), we see that we indeed recover the correct robot acceleration policy from (6). □

APPENDIX C
MATHEMATICAL DETAILS FOR PULLBACK BUNDLE DYNAMICAL SYSTEMS

In this section, we give a detailed derivation of Pullback Bundle Dynamical Systems and extraction of multi-task policies, showing that each of the key objects are geometrically well-defined.

Let M and N be smooth m- and n-dimensional robot configuration and task manifolds, having canonical projections π_M and π_N, respectively. Let $f : M \longrightarrow N$ be a smooth task map and let ∇ be the Levi-Civita connection in TN corresponding to metric g on N with Christoffel symbols

$$\Gamma^k_{ij} = \frac{1}{2}g^{kh}(\partial_i g_{jh} + \partial_j g_{ih} - \partial_h g_{ij}).$$

We form a pullback bundle f^*TN as defined in (2), which is a smooth n-vector bundle over M and itself an $(m+n)$-dimensional manifold.

I. Constructions on the Pullback Bundle

Note that since f^*TN is not a tangent bundle, we must reconstruct all of the standard objects and operators normally used for defining mechanical systems on tangent bundles. We first denote the pullback bundle projection as $f^*\pi_N$, and for convenience we define a pullback differential

$$f^*df : TM \longrightarrow f^*TN : (p, v) \mapsto (p, \pi_2(df_p(v)))$$

and the pullback of the identity map on TN

$$f^*\mathrm{Id}_{TN} : f^*TN \longrightarrow TN : (p, v) \mapsto (f(p), v),$$

both of which are smooth and globally well-defined.

We now construct a global pullback connection on f^*TN:

Lemma C.1 (Pullback Connection)**.** *For every $p \in M$, choose Christoffel symbols defined locally as*

$$f^*\Gamma^k_{ij}(p) = \frac{\partial f^\ell}{\partial x^i}(p)\Gamma^k_{\ell j}(f(p)).$$

These functions are globally extendable and form a global connection (which we call the pullback connection),

$$f^*\nabla : \Gamma(TM) \times \Gamma(f^*TN) \longrightarrow \Gamma(f^*TN).$$

Proof. The chosen Christoffel symbols are locally smooth. To show they are global, we must check whether they satisfy the chart transition formula for Christoffel symbols on vector bundles.

Consider two charts at $p \in M$ denoted by C_M and $\tilde{C}_M$ and two charts at $f(p) \in N$ denoted by C_N and $\tilde{C}_N$. Then TM and TN have corresponding local frames $(\frac{\partial}{\partial x^i})$, $(\frac{\partial}{\partial \tilde{x}^i})$ and $(\frac{\partial}{\partial y^i})$, $(\frac{\partial}{\partial \tilde{y}^i})$ respectively. Suppose we also have a vector bundle E over N (note that in our case we have $E = TN$). Given local frames (E_i) and $(\tilde{E}_i)$ for E,e there exists a smooth, non-singular matrix of functions (A_{ij}) such that locally $\tilde{E}_i = A_{ij}E_i$.

Now given a connection $\nabla : \Gamma(TN) \times \Gamma(E) \longrightarrow \Gamma(E)$, the related Christoffel symbols can be denoted by Γ^k_{ij} and $\tilde{\Gamma}^k_{ij}$ The chart transition formula for such Christoffel symbols is

$$\tilde{\Gamma}^k_{ij} = A^{pk}\frac{\partial y^q}{\partial \tilde{y}^i}\left(A_{jr}\Gamma^p_{qr} + \frac{\partial A_{jp}}{\partial y^q}\right).$$

We must now show a similar formula holds for $f^*\Gamma^k_{ij}$. Given the local frames for E above, we can construct local frames for the pullback bundle f^*E as $(f^*E_i) \triangleq (E_i \circ f)$ and $(f^*\tilde{E}_i) \triangleq (\tilde{E}_i \circ f)$. Then for $f^*\tilde{E}_i = B_{ij}f^*E_j$, it is easily seen that $B_{ij} = A_{ij} \circ f$. We can now compute

$$\begin{aligned}
f^*\tilde{\Gamma}^k_{ij} &= \frac{\partial \tilde{f}^\ell}{\partial \tilde{x}^i}(p)\tilde{\Gamma}^k_{\ell j}(f(p)) \\
&= \frac{\partial y^q}{\partial \tilde{y}^\ell}(f(p))\frac{\partial \tilde{f}^\ell}{\partial \tilde{x}^i}(p)A^{pk}(f(p)) \\
&\quad \left(A_{jr}(f(p))\Gamma^p_{qr}(f(p)) + \frac{\partial A_{jp}}{\partial y^q}(f(p))\right) \\
&= \frac{\partial x^\ell}{\partial \tilde{x}^i}(p)\frac{\partial f^q}{\partial x^\ell}(p)B^{pk}(p) \\
&\quad \left(B_{jr}(p)\Gamma^p_{qr}(f(p)) + \frac{\partial A_{jp}}{\partial y^q}(f(p))\right) \\
&= \frac{\partial x^\ell}{\partial \tilde{x}^i}(p)B^{pk}(p) \\
&\quad \left(B_{jr}(p)\frac{\partial f^q}{\partial x^\ell}(p)\Gamma^p_{qr}(f(p)) + \frac{\partial A_{jp}}{\partial y^q}(f(p))\frac{\partial f^q}{\partial x^\ell}(p)\right) \\
&= \frac{\partial x^\ell}{\partial \tilde{x}^i}(p)B^{pk}(p)\left(B_{jr}(p)f^*\Gamma^p_{\ell r}(p) + \frac{\partial B_{jp}}{\partial x^\ell}(p)\right)
\end{aligned}$$

We therefore find that the Christoffel symbols $f^*\Gamma^k_{ij}$ can be extended globally and the result follows. □

This naturally leads to a geometric acceleration operator for curves with velocity in f^*TN. Let $\sigma : I \longrightarrow M$ be a smooth curve, and denote the space of pullback bundle vector fields over σ

$$f^*\mathfrak{X}(\sigma) \triangleq \{V : I \longrightarrow f^*TN \mid V \text{ is smooth}, \\ V(t) \in f^*E_{\sigma(t)}, \forall t \in I\}.$$

As a classical result, for every curve σ in M, the pullback connection $f^*\nabla$ defines a unique acceleration operator

$$f^*D_\sigma : f^*\mathfrak{X}(\sigma) \longrightarrow f^*\mathfrak{X}(\sigma),$$

which locally takes the following form:

$$f^*D_\sigma V(t) = \left(\dot{V}^k(t) + \dot{\sigma}^i(t)V^j(t)f^*\Gamma^k_{ij}(\sigma(t))\right)f^*\partial_k|_{\sigma(t)}, \tag{10}$$

where $(f^*\partial_i)$ is the frame for f^*TN defined by $f^*\partial_i(p) = (p, \pi_2(\partial_i(f(p))))$.

Now we construct a pullback metric on f^*TN. Note that Lyapunov-type results for classical mechanical systems rely on the fact that the Levi-Civita connection is *compatible* with the related Riemannian metric. Indeed, this allows one to differentiate the norm of some energy, obtaining such variations as a function of the evolutionary equation of the associated mechanical system. Thus, it is critical to reproduce such a property in our framework to achieve stability.

Lemma C.2. *Let g be a Riemannian metric on N which is compatible with connection ∇. Then the pullback metric*

$$f^*g : f^*TN \times f^*TN \longrightarrow \mathbb{R} \\ ((p, v_1), (p, v_2)) \mapsto g_{f(p)}(v_1, v_2)$$

is compatible with the pullback connection $f^*\nabla$.

Proof. Since g and ∇ are compatible, we know they satisfy the compatibility condition

$$\frac{d}{dt}g_{\rho(t)}(X(t), X(t)) = 2g_{\rho(t)}(D_\rho X(t), X(t)) \tag{11}$$

for any smooth vector field X along any smooth curve ρ in N. To show the same for f^*g and $f^*\nabla$, let $\sigma : I \longrightarrow M$ be a smooth curve and let $V \in f^*\mathfrak{X}(\sigma)$. Then $\rho : I \longrightarrow N : t \mapsto f(\sigma(t))$ and $W : I \longrightarrow TN : t \mapsto f^*\mathrm{Id}_{TN}(V(t))$ are well-defined and smooth. Now using (11) we obtain

$$\begin{aligned}
&\frac{d}{dt}f^*g_{\sigma(t)}(V(t), V(t)) \\
&\quad = \frac{d}{dt}g_{f(\sigma(t))}\left(f^*\mathrm{Id}_{TN}(V(t)), f^*\mathrm{Id}_{TN}(V(t))\right) \\
&\quad = \frac{d}{dt}g_{\rho(t)}\left(W(t), W(t)\right) \\
&\quad = 2g_{\rho(t)}\left(D_\rho W(t), W(t)\right).
\end{aligned} \tag{12}$$

Applying (10) adapted for D_σ gives

$$\begin{aligned}
D_\rho W(t) &= \left(\dot{W}^k(t) + \dot{\rho}^\ell(t)W^j(t)\Gamma^k_{\ell j}(\rho(t))\right)\partial_k|_{\rho(t)} \\
&= \left(\dot{W}^k(t) + \frac{\partial f^\ell}{\partial x^i}(\sigma(t))\dot{\sigma}^i(t)W^j(t)\Gamma^k_{\ell j}(\rho(t))\right)\partial_k|_{\rho(t)} \\
&= \left(\dot{W}^k(t) + \dot{\sigma}^i(t)W^j(t)f^*\Gamma^k_{ij}(\sigma(t))\right)\partial_k|_{\rho(t)} \\
&= f^*\mathrm{Id}_{TN}\left(\left(\dot{V}^k(t) + \dot{\sigma}^i(t)V^j(t)f^*\Gamma^k_{ij}(\sigma(t))\right)\partial_k|_{\sigma(t)}\right) \\
&= f^*\mathrm{Id}_{TN}(f^*D_\sigma V(t)).
\end{aligned}$$

Thus continuing (12) finally provides

$$\begin{aligned}\frac{d}{dt}&f^*g_{\sigma(t)}(V(t),V(t))\\&=2g_{\rho(t)}\left(f^*\mathrm{Id}_{TN}(f^*D_\sigma V(t)),f^*\mathrm{Id}_{TN}(V(t))\right)\\&=2f^*g_{\sigma(t)}(f^*D_\sigma V(t),V(t)).\end{aligned}$$

□

Next we develop the sharp operator on f^*TN, useful for handling forces. First we define the dual pullback bundle

$$f^*T^*N=\coprod_{p\in M}(\pi_N^*)^{-1}(f(p)),$$

where π_N^* is the canonical projection for the cotangent bundle T^*N. We can now define the flat operator

$$\flat:f^*TN\longrightarrow f^*T^*N:(p,v)\mapsto f^*g_p(v,\cdot),$$

which is easily seen to be a smooth bundle isomorphism. The sharp operator is naturally defined to be the inverse of the flat operator, i.e.,

$$\sharp:f^*T^*N\longrightarrow f^*TN:(p,\omega)\mapsto(p,\omega)^\flat.$$

In particular, locally we have $(\omega(p)^\sharp)^i=g^{ij}(f(p))\omega_j(p)$.

Now consider dissipative forces $\mathcal{F}_D:TN\longrightarrow T^*N$, which we define satisfying the dissipative property $\langle\mathcal{F}_D(p,v),v\rangle\leq 0$ for all $(p,v)\in TN$, where $\langle\cdot,\cdot\rangle$ is the natural pairing between a covector and a vector. For convenience, we associate to $\mathcal{F}_D$ corresponding pullback dissipative forces as

$$f^*\mathcal{F}_D:f^*TN\longrightarrow f^*T^*N:(p,v)\mapsto(p,\pi_2(\mathcal{F}_D(f(p),v)),$$

such that we have $f^*\mathcal{F}_D(\cdot)^\sharp:f^*TN\longrightarrow f^*TN$. From this definition, it follows that we have $\langle f^*\mathcal{F}_D(p,v),v\rangle\leq 0$ for all $(p,v)\in f^*TN$ so that these pullback forces inherit the dissipative property. Note that the dissipative forces $\mathcal{F}_D$ can also be time-dependent and similar results to those shown in this work will hold, but we keep them time-independent for simplicity.

The last object we must construct on f^*TN is a gradient operator for handling potentials. Given any smooth function $\Phi:N\longrightarrow\mathbb{R}$, we locally define the smooth map

$$B_\Phi:M\longrightarrow f^*T^*N:p\mapsto\frac{\partial\Phi}{\partial y^i}(f(p))f^*dy^i|_p,$$

where (f^*dy^i) is the coframe for f^*T^*N defined by $f^*dy^i(p)=(p,\pi_2(dy^i(f(p))))$. This map is intentionally designed such that for every $(p,v)\in TM$ we have

$$\begin{aligned}B_\Phi(p)(f^*df_p(v))&=\frac{\partial\Phi}{\partial y^i}(f(p))f^*dy^i|_p(f^*df_p(v)))\\&=\frac{\partial\Phi}{\partial y^i}(f(p))f^*dy^i|_p\left(v^j\frac{\partial f^k}{\partial x^j}(p)\left.\frac{\partial}{\partial y^k}\right|_{f(p)}\right)\\&=v^j\frac{\partial f^i}{\partial x^j}(p)\frac{\partial\Phi}{\partial y^i}(f(p))=v^j\frac{\partial(\Phi\circ f)}{\partial y^i}(p)\\&=d(\Phi\circ f)(p,v)\end{aligned}\tag{13}$$

which is crucial for our stability analysis. Now we can define the pullback gradient operator such that

$$f^*\mathrm{grad}\,\Phi:M\longrightarrow f^*TN:p\mapsto B_\Phi(p)^\sharp.$$

To denote the total acceleration contributed by the pullback forces (using the sharp operator to indicate converting from forces to accelerations), we can also define the shorthand

$$\begin{aligned}f^*\mathcal{F}(\cdot)^\sharp:f^*TN&\longrightarrow f^*TN\\(p,v)&\mapsto f^*\mathcal{F}_D(p,v)^\sharp-f^*\mathrm{grad}\,\Phi(p)\end{aligned}$$

II. Local Pullback Bundle Dynamical Systems

We finally come to the definition of an important building block for Pullback Bundle Dynamical Systems (PBDS) called a local PBDS.

Definition C.1 (Local Pullback Bundle Dynamical System)**.** *Let $f:M\longrightarrow N$ be a smooth task map for a Riemannian task manifold (N,g), let $\Phi:N\longrightarrow\mathbb{R}_+$ be a smooth potential function, and let $\mathcal{F}_D:TN\longrightarrow T^*N$ be dissipative forces. Then for each $(p,v)\in M$, we can choose a curve $\alpha_{(p,v)}:(-\varepsilon,\varepsilon)\longrightarrow M$ resulting in $\gamma_{\alpha_{(p,v)}}:(-\varepsilon,\varepsilon)\longrightarrow f^*TN$ for some $\varepsilon>0$ such that $(f,g,\Phi,\mathcal{F}_D,\alpha_{(p,v)})$ forms a local Pullback Bundle Dynamical System (PBDS) satisfying*

$$\textit{PBDS}_{\alpha_{(p,v)}}\begin{cases}f^*D_{\alpha_{(p,v)}}\gamma_{\alpha_{(p,v)}}(s)=f^*\mathcal{F}(\gamma_{\alpha_{(p,v)}}(s))^\sharp\\\gamma_{\alpha_{(p,v)}}(0)=f^*df_{\alpha_{(p,v)}(0)}(\alpha'_{(p,v)}(0)),\\\alpha'_{(p,v)}(0)=(p,v).\end{cases}$$

The reason for this local PBDS definition is to ensure that PBDS is well-posed at $t=0$ (i.e. where a curve $\alpha_{\sigma'(0)}$ is required to define $f^*D_{\alpha_{\sigma'(0)}}$) and in cases where the curve $\alpha_{\sigma'(t)}$ defining a valid $\text{PBDS}_{\alpha_{\sigma'(t)}}$ for some $t\in[0,\infty)$ is not unique. For example, the latter may occur when $m>n$, which is often the case in practice.

Now we must show that a local PBDS exists at each point in TM. In particular, we must show that for any $(p,v)\in TM$, there exists a curve $\alpha_{(p,v)}$ satisfying $\text{PBDS}_{\alpha_{(p,v)}}$. If (U,φ) is a local chart for M centered at p let $\alpha_{(p,v)}(s)\triangleq\varphi^{-1}(s\boldsymbol{v})$, using local coordinates $\boldsymbol{v}\in\mathbb{R}^m$. This curve is well-defined and smooth for small times around zero, so the desired curve $\alpha_{(p,v)}$ always exist, meaning the corresponding local PBDS always exists.

To begin the process of extracting a robot motion policy, we can define a map giving the desired pullback task accelerations from local PBDSs corresponding to each point in the robot configuration tangent bundle TM:

$$G:TM\longrightarrow T(f^*TN):(p,v)\mapsto\dot\gamma_{\alpha_{(p,v)}}(0).\tag{14}$$

For this map to be well-defined, we must show that it does not depend on the choice of curves $\alpha_{(p,v)}$. Let $\alpha_{(p,v)}$ and $\beta_{(p,v)}$ be two curves in M satisfying $\text{PBDS}_{\alpha_{(p,v)}}$ and $\text{PBDS}_{\beta_{(p,v)}}$, respectively. These result in corresponding curves $\gamma_{\alpha_{(p,v)}}$ and $\gamma_{\beta_{(p,v)}}$ in f^*TN as provided in the local PBDS definition, which exist as a consequence of standard Cauchy-Lipschitz arguments. In particular, since every quantity defining a local PBDS is smooth, these curves are at least C^1, allowing us to compute $\dot\gamma_{\alpha_{(p,v)}}$ and $\dot\gamma_{\beta_{(p,v)}}$ pointwise. Thus using (10) we can compute

$$\begin{aligned}\dot\gamma^k_{\alpha_{(p,v)}}(0)&=(f^*\mathcal{F}(\gamma_{\alpha_{(p,v)}}(0))^\sharp)^k\\&\quad-\dot\alpha^i_{(p,v)}(0)\gamma^j_{\alpha_{(p,v)}}(0)f^*\Gamma^k_{ij}(\alpha_{(p,v)}(0))\\&=(f^*\mathcal{F}(f^*df(\dot\alpha_{(p,v)}(0)))^\sharp)^k\\&\quad-\dot\alpha^i_{(p,v)}(0)(f^*df(\dot\alpha_{(p,v)}(0)))^jf^*\Gamma^k_{ij}(\alpha_{(p,v)}(0))\\&=(f^*\mathcal{F}(\gamma_{\beta_{(p,v)}}(0))^\sharp)^k\\&\quad-\dot\beta^i_{(p,v)}(0)\gamma^j_{\beta_{(p,v)}}(0)f^*\Gamma^k_{ij}(\beta_{(p,v)}(0))\\&=\dot\gamma^k_{\beta_{(p,v)}}(0).\end{aligned}$$

Therefore we have $\dot{\gamma}_{\alpha_{(p,v)}}(0) = \dot{\gamma}_{\beta_{(p,v)}}(0)$ by the chart invariance of the derivative operator. This shows that G is well-defined and, in particular, smooth.

Given this result, we will adopt the shorthand $\gamma'_{v_p}(0) \triangleq \gamma'_{\alpha_{(p,v)}}(0)$ for $(p,v) \in TM$, knowing that suitable choices of $\alpha_{(p,v)}$ exist.

III. Multi-Task Pullback Bundle Dynamical Systems

Next, we continue with a detailed derivation for the extraction of multi-task PBDS policies. For k tasks, consider task maps $\{f_i : M \to N_i\}_{i=1,\dots,k}$. We equip every n_i-dimensional task manifold N_i with a Riemannian metric g_i, a potential function Φ_i, and dissipative forces $\mathcal{F}_{D,i}$.

We can associate to every robot position and velocity combination $v_p \in TM$ a smooth section $\gamma_{v_p,i} : [0,1] \longrightarrow f^*TN_i$ satisfying the single-task dynamics of the PBDS specified by $(f_i, g_i, \Phi_i, \mathcal{F}_{D,i})$. Extending the construction of (14) to an operator G_i for each task, we collect the vectors $\gamma'_{v_p,i}(0) = ((p,v),(\dot{\gamma}^v_{v_p,i}(0), \dot{\gamma}^a_{v_p,i}(0)))$ into a globally well-defined and smooth operator

$$\begin{aligned} S_i : TM &\longrightarrow T\big(f_i^*TN_i\big) \\ (p,v) &\mapsto \pi_{\text{VB}}(G_i(p,v)) = \pi_{\text{VB}}(\dot{\gamma}_{v_p,i}(0)) \\ &\qquad = ((f_i^*df_i)_p(v), (0, \dot{\gamma}^a_{v_p,i}(0))), \end{aligned}$$

where π_{VB} denotes the globally well-defined projection onto the vertical bundle. Recall that the vertical bundle of a bundle is the subbundle formed by the kernel of the differential of its standard projection, in this case $\ker(\pi_{f_i^*TN_i})$. In particular, locally we have $\pi_{\text{VB}}(b^i\partial_i^v + a^i\partial_i^a) = a^i\partial_i^a$ for $T(f_i^*TN_i)$.

Next, we form a map relating robot accelerations on TTM to their resulting task accelerations in $T(f^*TN_i)$:

$$\begin{aligned} Z_i : TTM &\longrightarrow T(f_i^*TN_i) \\ \big((p,v),a\big) &\mapsto \pi_{\text{VB}}\Big(d(f_i^*df_i)_{(p,v)}(a)\Big). \end{aligned}$$

Since the pullback differential $f_i^*df_i$ is a smooth map between smooth manifolds M and $f_i^*TN_i$, the differential of the pullback differential $d(f_i^*df_i)$ is globally well-defined. Thus Z_i is globally well-defined and smooth.

We also assign a weighting pseudometric w_i to each task tangent bundle TN_i. Given the higher-order task maps $F_i : TM \longrightarrow TN_i : (p,v) \mapsto (f_i(p), (df_i)_p(v))$, we can define globally well-defined pullback pseudometrics

$$\begin{aligned} F_i^*w_i : T(f_i^*TN_i) \times T(f_i^*TN_i) &\longrightarrow \mathbb{R} \\ ((p,v),a_1),((p,v),a_2) &\mapsto (w_i)_{F_i(p,v)}(a_1,a_2). \end{aligned}$$

Now we define a function which will choose a robot acceleration for each robot position and velocity in TM and will thus drive our PBDS dynamics:

$$\begin{aligned} \zeta : TM &\longrightarrow \mathcal{D} \subset TTM \\ (p,v) &\mapsto \operatorname*{arg\,min}_{a \in \mathcal{D}_{(p,v)}} \sum_{i=1}^{k} \frac{1}{2} \|Z_i(a) - S_i(p,v)\|^2_{F_i^*w_i}, \end{aligned} \tag{15}$$

where $\mathcal{D}$ is the globally well-defined distribution of TTM such that the subspace $\mathcal{D}_{(p,v)} \subseteq T_{(p,v)}TM$ satisfies $a^v = v$ componentwise for each $a = ((p,v),(a^v,a^a)) \in \mathcal{D}_{(p,v)}$. Because all the involved quantities are globally defined, as soon as (15) is well-defined (i.e., the minimization problem has a unique solution for every $(p,v) \in TM$) it is automatically globally defined.

To investigate conditions that guarantee existence and uniqueness of solutions to this minimization problem for every $(p,v) \in TM$, we derive solutions to (15) locally. In doing so, we also prove that (15) is smooth. First, in order to unpack $Z_i(a)$, we note that the differential $d(f_i^*df_i)$ locally takes the form

$$\boldsymbol{J}(\boldsymbol{f}_i^*\boldsymbol{df}_i)(p,v) = \begin{bmatrix} \boldsymbol{J}\boldsymbol{f}_i(p) & 0 \\ \dot{\boldsymbol{J}}\boldsymbol{f}_i(p,v) & \boldsymbol{J}\boldsymbol{f}_i(p) \end{bmatrix},$$

where for $(p,v) \in TM$ we define $\dot{Jf}_i$ locally as

$$(\dot{Jf}_i)_{kj}(p,v) = v^\ell \frac{\partial^2 f_i^j}{\partial x^\ell \partial x^k}(p).$$

Now for $a \in \mathcal{D}_{(p,v)}$ we have

$$\begin{aligned} Z_i(a) &= \pi_{\text{VB}}\left(d(f_i^*df_i)_{(p,v)}(a)\right) \\ &= \pi_{\text{VB}}\big(((f_i^*df_i)_p(v), ((Jf_i)_{jk}(p)v^k, \\ &\qquad (\dot{Jf}_i)_{jk}(p,v)v^k + (Jf_i)_{jk}(p)(a^a)^k))\big) \\ &= ((f_i^*df_i)_p(v), (0, (\dot{Jf}_i)_{jk}(p,v)v^k + (Jf_i)_{jk}(p)(a^a)^k). \end{aligned}$$

Thus for $a \in \mathcal{D}_{(p,v)}$ we can locally compute

$$\begin{aligned} &\|Z_i(a) - S_i(p,v)\|^2_{F_i^*w_i} \\ &\quad = \|\boldsymbol{J}\boldsymbol{f}_i(p)\boldsymbol{a}^a + \dot{\boldsymbol{J}}\boldsymbol{f}_i(p,v)\boldsymbol{v} - \dot{\boldsymbol{\gamma}}^a_{v_p,i}(0)\|^2_{\boldsymbol{w}_i^a(F_i(p,v))}. \end{aligned} \tag{16}$$

using the notation $\|\boldsymbol{x}\|^2_{\boldsymbol{B}} = \boldsymbol{x}^\top \boldsymbol{B}\boldsymbol{x}$. Additionally for $\dot{\gamma}^a_{v_p,i}(0)$ we can locally compute

$$\begin{aligned} (\dot{\gamma}^a_{v_p,i})^k(0) &= (f_i^*\mathcal{F}_i(\gamma_{v_p,i}(0))^\sharp)^k - v^\ell(t)\gamma^j_{v_p,i}(0)f_i^*(\Gamma_i)^k_{\ell j}(p) \\ &= f_i^*\mathcal{F}_{D,i}(\gamma_{v_p,i}(0))^\sharp - f_i^*\operatorname{grad}\Phi_i(p) \\ &\quad - v^\ell(t)\gamma^j_{v_p,i}(0)f_i^*(\Gamma_i)^k_{\ell j}(p) \\ &= g_i^{kj}(f_i(p))\left(\mathcal{F}^j_{D,i}((df_i)_p(v)) - \frac{\partial\Phi_i}{\partial y_i^j}(f_i(p))\right) \\ &\quad - v^\ell(t)(Jf_i)_{jh}(p)v^h(t)(Jf_i)_{r\ell}(p)(\Gamma_i)^k_{rj}(f_i(p)). \end{aligned}$$

Thus, we can compute solutions to (15) locally by

$$\begin{aligned} \boldsymbol{\zeta}(p,v) = &\left(\sum_{i=1}^{k} \boldsymbol{J}\boldsymbol{f}_i(p)^\top \boldsymbol{w}_i^a(F_i(p,v))\boldsymbol{J}\boldsymbol{f}_i(p)\right)^\dagger \\ &\left(\sum_{i=1}^{k} \boldsymbol{J}\boldsymbol{f}_i(p)^\top \boldsymbol{w}_i^a(F_i(p,v))\boldsymbol{\mathcal{A}}_i(p,v)\right), \end{aligned}$$

where $\boldsymbol{\mathcal{A}}_i(p,v) = \dot{\boldsymbol{J}}\boldsymbol{f}_i(p,v)\boldsymbol{v}(t) - \dot{\boldsymbol{\gamma}}^a_{v_p,i}(0)$. We can now see that that for ζ to be smooth and always have a unique solution, the sum within the pseudoinverse must be full rank. To achieve this, we make some key assumptions about the joint design of the task maps f_i and weighting pseudometrics w_i which will also be important for the stability results. First for convenience, we define a function giving the indices of tasks having nonzero weights:

$$\begin{aligned} \mathcal{I} : TM &\longrightarrow \mathcal{P}(\{1,\dots,k\}) \\ (p,v) &\mapsto \{i \in \{1,\dots,k\} \mid w_i((df_i)_p(v)) \neq 0\}. \end{aligned} \tag{17}$$

This allows us to build a family of product task maps $f_{(p,v)} \triangleq \prod_{i \in \mathcal{I}(p,v)} f_i$ giving for each $(p,v) \in TM$ the product map of the task maps associated to nonzero weights.

Now we make the following assumptions:

- $(A1)$ The pseudometrics w_i are at every point in TN_i either positive-definite or zero.
- $(A2)$ The differential $(df_{(p,v)})_p$ is always of rank m.

Note that $(A2)$ implies that $m \leq \sum_{i=1}^k n_i$. Now for all

$(p,v) \in TM$ we can compute

$$\begin{aligned}
&\sum_{i=1}^{k} \boldsymbol{J}\boldsymbol{f}_i(p)^\top \boldsymbol{w}_i^a(F_i(p,v))\boldsymbol{J}\boldsymbol{f}_i(p) \\
&= \sum_{i\in\mathcal{I}(p,v)} \boldsymbol{J}\boldsymbol{f}_i(p)^\top \boldsymbol{w}_i^a(F_i(p,v))\boldsymbol{J}\boldsymbol{f}_i(p) \\
&= \boldsymbol{J}\boldsymbol{f}_{(p,v)}^\top(p)\boldsymbol{w}_{(p,v)}^a(F_{(p,v)}(p,v))\boldsymbol{J}\boldsymbol{f}_{(p,v)}(p),
\end{aligned}$$

where $F_{(p,v)} \triangleq \prod_{i\in\mathcal{I}(p,v)} F_i$ and $\boldsymbol{w}_{(p,v)}^a(F_{(p,v)}(p,v)) \triangleq \text{blockdiag}(\{\boldsymbol{w}_i^a(F_i(p,v))\}_{i\in\mathcal{I}(p,v)})$. Given $(A1)$ and $(A2)$, the final local matrix is positive definite, so the original sum is indeed full rank, giving us unique solutions and smoothness of ζ. Thankfully, these assumptions are easily satisfied in practice as explained in Sec III.

However, it is useful to show that in general, with probability one the local matrix $\boldsymbol{J}\boldsymbol{f}_{(p,v)}(p)$ has full rank at every point $p \in M$ if the size of $\mathcal{I}(p,v)$ is large enough (in particular, if $\text{card}(\mathcal{I}(p,v)) \geq 2m$ for every $(p,v) \in TM$). This may be inferred as a consequence of the Whitney immersion theorem. However, by leveraging transversality theory we can directly prove that, i.e., for a sufficiently high number of tasks, the required full-rank properties are satisfied pointwise with probability one.

To provide a more precise statement, we proceed by steps. Our objective is achieved by selecting at least $2m$ tasks, if we can shown that the complementary of the set

$$\begin{aligned}
\mathcal{S} \triangleq \Big\{&(f_1,\dots,f_{2m}) \in C^\infty(M;N_1)\times \\
&\cdots \times C^\infty(M;N_{2m}) :\ \text{rank}(f^{2m}) < m\Big\}
\end{aligned}$$

is dense with respect to the Whitney topology (see, e.g., [27]). We prove that the complementary of the set $\mathcal{S}$ is dense in $C^\infty(M;N_1)\times\cdots\times C^\infty(M;N_{2m})$ with respect to the Whitney topology by the jet transversality theorem (see, e.g., [27]). For this, we denote

$$\theta(f_1,\dots,f_{2m}) \triangleq \left(\tilde{\theta}, \theta_0^{f_1},\dots,\theta_0^{f_{2m}},\theta_1^{f_1},\dots,\theta_1^{f_{2m}}\right)$$

the jet of order one of $f_1,\dots,f_{2m}$. Let us define the following smooth mapping in local jet coordinates

$$\rho(\theta) \triangleq \left(\theta_1^{f_1},\dots,\theta_1^{f_{2m}}\right).$$

It is clear that ρ is a local summersion. It is also clear that locally we have

$$\mathcal{S} = \bigcup_{1\leq i\leq m-1} \rho^{-1}(L(2m,m;i))$$

where $L(2m,m;i)$ denotes the Stiefel manifold of linear mappings from $\mathbb{R}^m$ to $\mathbb{R}^{2m}$ of rank $1 \leq i \leq m$. Therefore, in terms of codimension, we have

$$\begin{aligned}
\text{codim}(\mathcal{S}) &\geq \text{codim}\big(\rho^{-1}(L(2m,m;m-1))\big) \\
&= \text{codim}\big(L(2m,m;m-1)\big) = m+1.
\end{aligned}$$

Then, the conclusion comes from a direct application of the jet transversality theorem (see, e.g., [27]).

Now using the above constructions, we finally give the definition of a multi-task PBDS:

Definition C.2 (Multi-Task Pullback Bundle Dynamical System)**.** *Let $\{f_i : M \longrightarrow N_i\}_{i=1,\dots,k}$ be smooth task maps for a Riemannian task manifolds (N,g), with corresponding smooth potential functions $\Phi_i : N_i \longrightarrow \mathbb{R}_+$, dissipative forces $\mathcal{F}_{D,i} : TN_i \longrightarrow T^*N_i$, and weighting pseudometrics 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 $\sigma : [0,\infty) \longrightarrow M$ satisfying*

$$\begin{cases}
\ddot{\sigma}(t) = \zeta(\dot{\sigma}(t)) = \\
\qquad \underset{a\in\mathcal{D}_{\dot{\sigma}(t)}}{\arg\min}\ \sum_{i=1}^k \frac{1}{2}\|Z_i(a) - S_i(\dot{\sigma}(t))\|^2_{F_i^* w_i} \\
\dot{\sigma}(0) = (p_0,v_0).
\end{cases} \tag{18}$$

Assuming $(A1)$ and $(A2)$ to ensure smoothness of ζ and the existence of unique solutions to (15), we see that we have existence, uniqueness, and smoothness of solution curves σ for a multi-task PBDS.

IV. Proof of Multi-Task PBDS Stability

This section proves the stability results summarized in the main body of the paper. First, we make another key assumption related to the design of weighting pseudometrics. Using the convenience function in (17), for dissipative forces we define $\mathcal{F}_{D,(p,v)} \triangleq \prod_{i\in\mathcal{I}(p,v)} \mathcal{F}_{D,i}$. We now assume that for every $(p,v) \in TM$ we have the following:

$(A3)$ The dissipative force $\mathcal{F}_{D,(p,v)}$ is strictly dissipative, meaning $\langle \mathcal{F}_{D,(p,v)}\big((df_{(p,v)})_p(v)\big), (df_{(p,v)})_p(v)\rangle < 0$ for $v \neq 0$,

Our Lyapunov stability results are as follows:

Proposition C.1 (Lyapunov Function for Multi-Task PBDS)**.** *Let $\{(f_i,g_i,\Phi_i,\mathcal{F}_{D,i},w_i)\}_{i=1,\dots,k}$ be a multi-task PBDS. Then given the function $V : TM \longrightarrow [0,\infty)$ defined by*

$$V(p,v) = \sum_{i=1}^k \frac{1}{2}\big\|(f_i^* df_i)_p(v)\big\|^2_{(f_i^* g_i)_p} + \Phi_i \circ f_i(p)$$

we have that

$$\frac{dV}{dt}(\dot{\sigma}(t)) = \sum_{i=1}^k \langle \mathcal{F}_{D,i}((df_i)_{\sigma(t)}(\dot{\sigma}(t))), (df_i)_{\sigma(t)}(\dot{\sigma}(t))\rangle < 0$$

for $\dot{\sigma}(t) \neq 0$ along solution curves σ to the multi-task PBDS.

Proof. Taking one element of the sum in $\frac{d}{dt}V(\sigma'(t))$ from the first term we have

$$\begin{aligned}
&\frac{d}{dt}\Big(\frac{1}{2}\big\|(f_i^* df_i)_{\sigma(t)}(\dot{\sigma}(t))\big\|^2_{(f_i^* g_i)_{\sigma(t)}}\Big) \\
&= \frac{d}{dt}\Big(\frac{1}{2}(f_i^* g_i)_{\sigma(t)}\big((f_i^* df_i)_{\sigma(t)}(\dot{\sigma}(t)),(f_i^* df_i)_{\sigma(t)}(\dot{\sigma}(t))\big)\Big) \\
&= (f_i^* g_i)_{\sigma(t)}(f_i^* D_{\sigma,i}\gamma_{\sigma,i}(0),\gamma_{\sigma,i}(0)) \qquad (19) \\
&= (f_i^* g_i)_{\sigma(t)}(f_i^* \mathcal{F}_{D,i}(\gamma_{\sigma,i}(0))^\sharp,\gamma_{\sigma,i}(0)) \\
&\quad - (f_i^* g_i)_{\sigma(t)}(f^*\text{grad}\,\Phi_i(\sigma(t)),\gamma_{\sigma,i}(0)) \\
&= \langle f_i^* \mathcal{F}_{D,i}(\gamma_{\sigma,i}(0)),\gamma_{\sigma,i}(0)\rangle \\
&\quad - \langle f_i^* \text{grad}\Phi_i(\sigma(t))^\flat,\gamma_{\sigma,i}(0)\rangle \\
&= \langle \mathcal{F}_{D,i}((df_i)_{\sigma(t)}(\dot{\sigma}(t))),(df_i)_{\sigma(t)}(\dot{\sigma}(t))\rangle \\
&\quad - B_{\Phi_i}(\sigma(t))\big((f_i^* df_i)_{\sigma(t)}(\dot{\sigma}(t))\big) \\
&= \langle \mathcal{F}_{D,i}((df_i)_{\sigma(t)}(\dot{\sigma}(t))),(df_i)_{\sigma(t)}(\dot{\sigma}(t))\rangle \qquad (20) \\
&\quad - d(\Phi_i \circ f_i)(\dot{\sigma}(t)),
\end{aligned}$$

where for (19) we use the compatibility of the pullback metric and the pullback connection, and for (20) we use (13). Likewise, for the second term we simply have

$$\frac{d}{dt}\Big(\Phi_i \circ f_i(\sigma(t))\Big) = d(\Phi_i \circ f_i)(\dot{\sigma}(t)).$$

Thus, we have

$$\frac{d}{dt}\big(V(\dot\sigma(t))\big) = \sum_{i=1}^{k}\langle \mathcal{F}_{D,i}((df_i)_{\sigma(t)}(\dot\sigma(t))), (df_i)_{\sigma(t)}(\dot\sigma(t))\rangle.$$

Then from assumption $(A2)$ and the fact that all forces $\mathcal{F}_{D,i}$ are dissipative, we have that $\dot V(\dot\sigma(t)) < 0$ for $\dot\sigma(t) \neq 0$. □

We now provide global stability results:

Theorem C.1 (Global Stability of Multi-Task PBDS)**.** *Let $\{(f_i, g_i, \Phi_i, \mathcal{F}_{D,i}, w_i)\}_{i=1,\dots,k}$ be a multi-task PBDS where $\Phi_i \circ f_i$ are proper maps. Then given the Lyapunov function V of Proposition C.1, the multi-task PBDS satisfies the following:*

- *The sublevel sets of V are compact and positively invariant for $\zeta : TM \longrightarrow TTM$, so that every solution curve σ is defined in the whole interval $[0, +\infty)$.*
- *For every $\beta > 0$, every solution curve σ in M starting at $\dot\sigma(0) \in V^{-1}([0,\beta])$ converges in $V^{-1}([0,\beta])$ as $t \to +\infty$ to an equilibrium configuration $(p, 0) \in V^{-1}([0,\beta])$ such that grad $\Phi_i(p) = 0$ if $w_i(p, 0) \neq 0$.*

Proof. First, we aim to show that the sublevel sets of V are compact. Following the argument in the proof of [23, Theorem 6.47], since we know $\Phi_i \circ f_i$ are proper, it is sufficient to show that

$$f^*g : TM \times TM \longrightarrow \mathbb{R}$$

$$((p, v_1), (p, v_2)) \mapsto \sum_{i=1}^{k}(f_i^* g_i)_p\big((f^*df_i)_p(v_1), (f^*df_i)_p(v_2)\big).$$

defines a metric. Symmetry and bilinearity follow easily from the properties of the task pullback metrics f^*g_i. For positive-definiteness, we first take assumption $(A2)$, which also gives that the differential df_p is always of rank m, for the product map $f \triangleq \prod_{i=1}^k f_i$. Now note that if $f^*g_p(v, v) = 0$, then we must have $(f_i^* g_i)_p\big((f^*df_i)_p(v), (f^*df_i)_p(v)\big) = 0$ for each task. With df_p of full rank, this implies that $v = 0$. Thus it is clear that f^*g is positive-definite and is a metric, which in turn implies that V is proper. In addition, since we have $\dot V(\dot\sigma(t)) < 0$ from Proposition C.1, the level sets of V are positively invariant.

Now, let $\beta > 0$ and define the set

$$\begin{aligned} A = \Big\{&(p, v) \in V^{-1}([0,\beta]) : \\ &\langle \mathcal{F}_{D,(p,v)}\big((df_{(p,v)})_p(v)\big), (df_{(p,v)})_p(v)\rangle = 0\Big\} \\ = \Big\{&(p, v) \in V^{-1}([0,\beta]) : v = 0\Big\}. \end{aligned}$$

where the last equality follows from $(A3)$. First, we wish to show that the set

$$\begin{aligned} B = \Big\{&(p, v) \in V^{-1}([0,\beta]) : \\ &v = 0,\ \text{grad}\Phi_i(p) = 0,\ i \in \mathcal{I}(p, v)\Big\} \end{aligned}$$

is the largest positively invariant set in A for the multi-task PBDS. By its definition, we have $B \subseteq A$. Now suppose by contradiction that we have a point $(p, 0)$ in the largest positively invariant set of A such that $(p, 0) \notin B$. Then grad $\Phi_(p) \neq 0$ for some $i \in \mathcal{I}(p, v)$. Let the multi-task PBDS system evolve starting from $(p, 0)$, and let $\sigma_{(p,0)}$ be its solution. If $\dot\sigma_{(p,0)}(t) = 0$ for all $t \in [0, +\infty)$, then $\dot\sigma_{(p,0)}(0) = 0$, and $\sigma_{(p,0)}(t) = p$ does not satisfy the system. Thus, there must be $\bar t > 0$ such that $\dot\sigma_{(p,0)}(\bar t) \neq 0$, in contradiction with $\sigma_{(p,0)}(t) \in A$ for every $t > 0$.

Now using the LaSalle Invariance Principle, we can conclude that for solution curves σ to the multi-task PBDS having initial condition $\dot\sigma(0) \in V^{-1}([0,\beta])$, we have $\lim_{t\to+\infty} \text{dist}(\dot\sigma(t), B) = 0$, and the conclusion follows. □

APPENDIX D
MATHEMATICAL DETAILS FOR PBDS POLICIES ON TANGENT BUNDLE TASKS

Here we provide further details for the extension of the PBDS framework to tangent bundle tasks, as used in Sec. IV-B. This framework is mainly instructive to show the challenges both in performing such an extension in a way that is geometrically well-defined and in finding practical application for using velocity-dependent Riemannian metrics to drive task behavior. Indeed, in this work we were not able to find useful robotic tasks whose behavior could not be replicated using the basic multi-task PBDS framework.

Given a task map $f : M \longrightarrow N$, consider the higher-order task map $F : TM \longrightarrow TN : (p, v) \mapsto (f(p), df_p(v))$. It would be convenient if we could follow the same steps we used to extract a robot acceleration policy from task maps mapping from M. However, task maps mapping from TM introduce two major new challenges.

First, the previous procedure applied in this case would produce curves on TM rather than on M as we desire. Unfortunately, not every curve in TM correctly represents a curve in M. In particular, given a curve $\mu : I \longrightarrow TM$, the velocity curve $\dot\mu$ takes points in TTM, which have the form $((p, v), (b, a))$, where (p, v) is the base point from TM and (b, a) is the vector portion. If μ is to correctly represent a curve $\sigma : I \longrightarrow M$, we must have that $\dot\mu(t) = ((\sigma(t), \dot\sigma(t)), (\dot\sigma(t), \ddot\sigma(t)))$ for every $t \in I$. This clearly imposes a velocity constraint on admissible curves in TM that we can consider: namely, points $((p, v), (b, a))$ in the velocity of admissible curves curves in TM must satisfy $v = b$ (in local coordinates). The way we choose to handle this is through a subbundle constraint, as we will see.

Second, since velocities in this case are on TTM, an acceleration operator applied to a curve μ in TM gives twice as many acceleration vectors as an acceleration operator applied to a curve in M. In particular, if μ should represent a curve σ in M, the acceleration operator gives vectors that should represent both the second and third time derivatives of σ, i.e. both the acceleration on the jerk. Thus to meet our goal of an acceleration policy, it is desirable to project away the jerk components. The GDS framework in RMPflow accomplished this by applying a projection onto the horizontal bundle [20]. However, as discussed previously, the horizontal bundle is chart-dependent, so this projection is not globally well-defined.

Instead, we choose to handle this by applying a permutation to swap the position of the jerk and acceleration vectors, an operation which is globally well-defined for parallelizable manifolds that are embedded in the Euclidean space, such as Lie groups. Then the global vertical bundle projection will correctly remove the jerk components, leaving us with the acceleration components that we can use to build an acceleration policy. Note that an alternative is to apply the vertical bundle projection without a permutation and recover a jerk policy, but leave that alternative open to future work.

With this motivation and roadmap, we begin by considering enforcement of velocity constraints.

I. Enforcing Velocity Constraints

We will enforce the velocity constraints discussed above using subbundle constraints. For this, we first consider the augmented task map

$$\begin{aligned} F \times \pi_M : TM &\longrightarrow TN \times M \\ (p,v) &\mapsto (F(p,v), \pi_M(p,v)) = ((p, df_p(v)), p) \end{aligned}$$

This would suffice if we aimed to recover a jerk policy. However, now consider the permutation map $\sigma_{TN} : TN \longrightarrow \mathbb{R}^{2d} : (p,v) \mapsto (d\bar{\varphi}_p(v), \bar{\varphi}(p))$, where N has an embedding $\bar{\varphi} : N \longrightarrow \mathbb{R}^d$. Note that this map is globally well-defined for parallelizable manifolds that are embedded in Euclidean space, such as all Lie groups (though unfortunately, a notable non-parallelizable manifold is $\mathbb{S}^2$). Using this and the embedded task map $\bar{f} : M \longrightarrow \mathbb{R}^d$, we can form the composition

$$\begin{aligned} \hat{F} \triangleq \sigma_{TN} \circ F : TM &\longrightarrow \mathbb{R}^{2d} \\ (p,v) &\mapsto (d\bar{f}_p(v), \bar{f}(p)) \end{aligned}$$

We now construct the full augmented permuted task map which we will use:

$$\begin{aligned} \breve{F} \triangleq \hat{F} \times \pi_M : TM &\longrightarrow \mathbb{R}^{2d} \times M \\ (p,v) &\mapsto (d\bar{f}_p(v), \bar{f}(p), p). \end{aligned}$$

Now given the modified task tangent bundle $T(\mathbb{R}^{2d} \times M)$, we can construct a pullback bundle $\breve{F}^*T(\mathbb{R}^{2d} \times M)$. Note that this is a vector bundle which is isomorphic to the pullback direct sum vector bundle $\hat{F}^*(T\mathbb{R}^{2d}) \oplus \pi_M^* TM$.

Thus using straightforward identifications, we can define the map

$$\begin{aligned} &\hat{F}^*\mathcal{L} : TTM \longrightarrow \hat{F}^*T\mathbb{R}^{2d} \oplus \pi_M^* TM \\ &((p,v),(b,a)) \mapsto ((p,v), \\ &\qquad d\hat{F}_{(p,v)}(b,a) + ((d\pi_M)_{(p,v)}(b,a) - v)). \end{aligned}$$

We can also define the two vector subbundles of the pullback bundle $\breve{F}^*T(\mathbb{R}^{2d} \times M)$:

$$\begin{aligned} \breve{F}^*G_{\text{VB}} = \Big\{ &((p,v), c_i^1 \partial x^i_{\mathbb{R}^{2d}} + c_i^2 \partial v^i_{\mathbb{R}^{2d}} + c_i^3 \partial x^i_M) \\ &\in \hat{F}^*T\mathbb{R}^{2d} \oplus \pi_M^* TM \\ &\mid c_i^1 = 0 \text{ for } i = 1, \ldots, d, c_i^3 = 0 \text{ for } i = 1, \ldots, m \Big\}, \\ \breve{F}^*G_{\mathbb{R}^{2d}} = \Big\{ &((p,v), c_i^1 \partial x^i_{\mathbb{R}^{2d}} + c_i^2 \partial v^i_{\mathbb{R}^{2d}} + c_i^3 \partial x^i_M) \\ &\in \hat{F}^*T\mathbb{R}^{2d} \oplus \pi_M^* TM \mid c_i^3 = 0 \text{ for } i = 1, \ldots, m \Big\}, \end{aligned}$$

which are globally well-defined and smooth, in the first case due to its vertical bundle structure and in the second case due to standard projection arguments exploiting the direct sum structure of the pullback bundle described earlier. Note that we have $\breve{F}^*G_{\text{VB}} \subseteq \breve{F}^*G_{\mathbb{R}^{2d}}$ as vector bundles. We also use these to define a further subbundle of the pullback bundle $\breve{F}^*G = \breve{F}^*G_{\text{VB}} \oplus \breve{F}^*G^\perp_{\mathbb{R}^{2d}}$, where $\breve{F}^*G^\perp_{\mathbb{R}^{2d}}$ is the orthogonal complement of $\breve{F}^*G_{\mathbb{R}^{2d}}$ in the pullback bundle $\breve{F}^*T(\mathbb{R}^{2d} \times M)$. This orthogonal complement can be defined by

$$\begin{aligned} \breve{F}^*G^\perp_{\mathbb{R}^{2d}} = \{&(p,v) \in \breve{F}^*T(\mathbb{R}^{2d} \times M) \\ &: \breve{F}^* g_p(v,w) = 0, \forall w \in \breve{F}^*G_{\mathbb{R}^{2d}} \}, \end{aligned}$$

given some pullback metric $\breve{F}^*g$ for $F^*T(\mathbb{R}^{2d} \times M)$.

Now suppose we have a robot acceleration $\sigma'' = ((p,v),(b,a)) \in TTM$, with corresponding point $\hat{F}^*\mathcal{L}(\sigma'') \in \breve{F}^*T(\mathbb{R}^{2d} \times M)$ in the pullback bundle. If we also have $\hat{F}^*\mathcal{L}(\sigma'') \in \breve{F}^*G_{\mathbb{R}^{2d}}$, this implies that $(d\pi_M)_{(p,v)}(b,a) - v = 0$, or componentwise $b - v = 0$. This is exactly the velocity constraint we aim to achieve. Thus, we will enforce this constraint it by ensuring our dynamics evolve on the subbundle $\breve{F}^*G_{\mathbb{R}^{2d}}$.

In order to do so, we need to develop a connection on the pullback bundle $\breve{F}^*T(\mathbb{R}^{2d} \times M)$ which is guaranteed to provide curves that stay in the subbundle $\breve{F}^*G_{\mathbb{R}^{2d}}$ if they start in the subbundle. First we define a projection $P_{\breve{F}^*G^\perp_{\mathbb{R}^{2d}}}$ from $\breve{F}^*T(\mathbb{R}^{2d} \times M)$ onto $\breve{F}^*G^\perp_{\mathbb{R}^{2d}}$, which is easily seen to be globally well-defined. Now consider the tensorization of the $P_{\breve{F}^*G^\perp_{\mathbb{R}^{2d}}}$ operator:

$$\begin{aligned} P^T_{\breve{F}^*G^\perp_{\mathbb{R}^{2d}}} : TM &\longrightarrow \breve{F}^*T(\mathbb{R}^{2d} \times M) \otimes \breve{F}^*T^*(\mathbb{R}^{2d} \times M) \\ (p,v) &\mapsto \breve{F}^*E_i^*(v_p) \left(P_{\breve{F}^*G^\perp_{\mathbb{R}^{2d}}}(\breve{F}^*E_j(v_p)) \right) \\ &\qquad \breve{F}^*E_i(v_p) \otimes \breve{F}^*E_j^*(p), \end{aligned}$$

where $(\breve{F}^*E_i^*)$ is the dual local frame of $\breve{F}^*T^*(\mathbb{R}^{2d} \times M)$ associated to the local frame $(\breve{F}^*E_i)$ of $\breve{F}^*T(\mathbb{R}^{2d} \times M)$.

Now let $\nabla_{\mathbb{R}^{2d}}$ and ∇_M be connections over $\mathbb{R}^{2d}$ and M, respectively (e.g., $\nabla_{\mathbb{R}^{2d}}$ can be the pushforward of a connection on TN through σ_{TN}). Then we can combine $\nabla_{\mathbb{R}^{2d}}$ and ∇_M to form a connection $\breve{\nabla}$ over $\mathbb{R}^{2d} \times M$. From here, we can build a pullback connection $\breve{F}^*\breve{\nabla}$ over TM using the usual definition, having associated Christoffel symbols $\breve{F}^*\breve{\Gamma}^k_{ij}$.

Then for every vector field X over TM and section V of $\breve{F}^*T(\mathbb{R}^{2d} \times M)$ we can build a well-defined operator

$$\breve{F}^*\breve{\nabla} P^T_{\breve{F}^*G^\perp_{\mathbb{R}^{2d}}}(\cdot,\cdot) : TM \to \breve{F}^*T(\mathbb{R}^{2d} \times M) \otimes \breve{F}^*T^*(\mathbb{R}^{2d} \times M)$$

such that we may consider the identification

$$\breve{F}^*\breve{\nabla} P^T_{\breve{F}^*G^\perp_{\mathbb{R}^{2d}}}(\cdot, V) : TM \to \breve{F}^*T(\mathbb{R}^{2d} \times M).$$

From this we can compute the coordinate representation

$$\begin{aligned} \breve{F}^*\breve{\nabla} P^T_{\breve{F}^*G^\perp_{\mathbb{R}^{2d}}}(\cdot, V) = (&\mathbb{1}_{\{h > 2d, k \leq 2d\}}(h,k) \\ &- \mathbb{1}_{\{h \leq 2d, k > 2d\}}(h,k)) \breve{F}^*\breve{\Gamma}^k_{jh} X^j V^h \mathfrak{n}_k, \end{aligned}$$

where $\mathbb{1}$ is an indicator function, and $(\mathfrak{n}_i)$ is an orthonormal basis for $\breve{F}^*T^*(\mathbb{R}^{2d} \times M)$. This allows us to define a new connection over the pullback bundle $\breve{F}^*T^*(\mathbb{R}^{2d} \times M)$ as

$$\begin{aligned} &{}^{\breve{F}^*G_{\mathbb{R}^{2d}}}\breve{F}^*\breve{\nabla}_X V \triangleq \breve{F}^*\breve{\nabla}_X V + \breve{F}^*\nabla P^T_{\breve{F}^*G^\perp_{\mathbb{R}^{2d}}}(\cdot, V), \end{aligned}$$

whose Christoffel symbols take the form

$$\begin{aligned} {}^{\breve{F}^*G_{\mathbb{R}^{2d}}}\breve{F}^*\breve{\Gamma}^k_{jh} = (1 + &\mathbb{1}_{\{h > 2d, k \leq 2d\}}(h,k) \\ &- \mathbb{1}_{\{h \leq 2d, k > 2d\}}(h,k)) \breve{F}^*\breve{\Gamma}^k_{jh}. \end{aligned} \tag{21}$$

In particular, notice when that when $V(\breve{F}(p,v)) \in \breve{F}^*G_{\mathbb{R}^{2d}}$

for every $(p, v) \in TM$, we have

$$\breve{F}^*\breve{\nabla} P^T_{\breve{F}^* G^{\perp}_{\mathbb{R}^{2d}}}(\cdot, V)(\breve{F}(p, v)) = -\sum_{k=2d+1}^{2d+m} \left(\sum_{j=1}^{2m} \sum_{h=1}^{2d} \breve{F}^* \breve{\Gamma}^k_{jh} X^j V^h \right) \mathfrak{n}_k \in \breve{F}^* G^{\top}_{\mathbb{R}^{2d}}.$$

In short, this new connection $\overset{\breve{F}^* G_{\mathbb{R}^{2d}}}{\breve{F}^*\breve{\nabla}}$ effectively removes the connection terms which may cause a curve to leave the subbundle $\breve{F}^* G_{\mathbb{R}^{2d}}$ where our desired curves should stay. For brevity, we omit the full proof of this result which generalizes [23, Prop. 4.85].

II. Multi-Task Acceleration Policy Optimization

Now using this connection, we can build local pullback bundle dynamical systems having curves that satisfy our desired velocity constraints:

Definition D.1 (Local PBDS on Tangent Bundle Task)**.** *Let $f : M \longrightarrow N$ be a smooth task map, where N is embedded in $\mathbb{R}^d$, and associate to f the augmented, permuted task map $\breve{F} : TM \longrightarrow \mathbb{R}^{2d} \times M$. Also let $\breve{g}$ be a Riemannian metric on $\mathbb{R}^{2d} \times M$, let $\breve{\Phi} : \mathbb{R}^{2d} \times M \longrightarrow \mathbb{R}_+$ be a smooth potential function, and let $\breve{\mathcal{F}}_D : T(\mathbb{R}^{2d} \times M) \longrightarrow T^*(\mathbb{R}^{2d} \times M)$ be dissipative forces. Then for each $(p, v) \in M$, we can choose a curve $\alpha_{(p,v)} : (-\varepsilon, \varepsilon) \longrightarrow TM$ resulting in $\gamma_{\alpha_{(p,v)}} : (-\varepsilon, \varepsilon) \longrightarrow \breve{F}^* T(\mathbb{R}^d \times M)$ for some $\varepsilon > 0$ such that $(f, g, \Phi, \mathcal{F}_D, \alpha_{(p,v)})$ forms a local Pullback Bundle Dynamical System (PBDS) satisfying*

$$\textit{PBDS}_{\alpha_{(p,v)}} \begin{cases} P_{\breve{F}^* G} \left(\overset{\breve{F}^* G_{\mathbb{R}^{2d}}}{\breve{F}^* D_{\alpha_{(p,v)}}} \left(P_{\breve{F}^* G}(\gamma_{\alpha_{(p,v)}}(s)) \right) \right) \\ \quad = P_{\breve{F}^* G} \left(\breve{F}^* \breve{\mathcal{F}}(\gamma_{\alpha_{(p,v)}}(s))^\sharp \right) \\ \gamma_{\alpha_{(p,v)}}(0) = \breve{F}^* d\breve{F}_{\alpha_{(p,v)}(0)}(\alpha'_{(p,v)}(0)), \\ \alpha_{(p,v)}(0) = (p, v), \end{cases}$$

where we denote total pullback acceleration $\breve{F}^*\breve{\mathcal{F}}(\cdot)^\sharp$, analogous to Appx. C. As before, given $(p, v) \in TM$, we have that $\gamma'_{\alpha_{(p,v)}}(0)$ from a corresponding local PBDS does not depend on the choice of curve $\alpha_{(p,v)}$, so we will again use the shorthand $\gamma'_{v_p}(0) \triangleq \gamma'_{\alpha_{(p,v)}}(0)$.

Now, in order to extract a single robot acceleration policy from a set of tasks, we trace the constructions in Appx. C.III with some modifications to form the required geometric optimization problem. For k tasks, consider task maps $\{f_i : M \to N_i\}_{i=1,\dots,k}$, where each N_i is embedded in $\mathbb{R}^{d_i}$, giving corresponding embedded task map $\bar{f}_i$. We equip each associated embedding space $\mathbb{R}^{2d_i}$ for TN_i with a Riemannian metric $g_{\mathbb{R}^{2d_i}}$ and a potential function $\Phi_{\mathbb{R}^{2d_i}}$ depending only on v for $(p, v) \in T\mathbb{R}^{2d_i}$. We also equip each $T\mathbb{R}^{2d_i}$ with a weighting pseudometric $w_{T\mathbb{R}^{2d_i}}$ and dissipative forces $\mathcal{F}_{D,R^{2d_i}}$, both dependent only on the base points of $T\mathbb{R}^{2d_i}$, as will be important for defining our geometric optimization problem. We also set the first d components of $\mathcal{F}_{D,R^{2d_i}}$ to be zero. Next, we define similar objects g_M, Φ_M on M and w_M, $\mathcal{F}_{D,M}$ on TM, although their design is inconsequential as their contributions are ultimately projected away.

From these we can form on each augmented, permuted task space $\mathbb{R}^{2d_i} \times M$ the product metric $\breve{g}_i = g_{\mathbb{R}^{2d_i}} \oplus g_M$ and potential forces $\breve{\Phi}_i$ such that $\breve{\Phi}_i(v, q, p) = \Phi_{\mathbb{R}^{2d_i}}(v, q) + \Phi_M(p)$. Likewise on each $T(\mathbb{R}^{2d_i} \times M)$ we form the weight product pseudometic $\breve{w}_i = w_{T\mathbb{R}^{2d_i}} \oplus w_{TM}$ and dissipative forces $\breve{\mathcal{F}}_{D,i} = \mathcal{F}_{D,R^{2d_i}} \oplus \mathcal{F}_{D,M}$.

Then we define two distributions of $T(\breve{F}_i^* T(\mathbb{R}^{2d_i} \times M)) \cong T(\hat{F}^* T\mathbb{R}^{2d_i}) \oplus T(\pi_M^* TM)$ analogous to the subbundles $\breve{F}^* G_{\text{VB}}$ and $\breve{F}^* G_{\mathbb{R}^{2d}}$:

$$\mathcal{D}_{\text{VB},i} = \Big\{ \big(((p, v), a), c_j^1 \partial x_M^j + c_j^2 \partial v_{M,1}^j + c_j^3 \partial a^j_{\mathbb{R}^{2d_i}} + c_j^4 \partial v^j_{\mathbb{R}^{2d_i}} + c_j^5 \partial v_{M,2}^j \big) \in T(\hat{F}_i^* T\mathbb{R}^{2d_i}) \oplus T(\pi_M^* TM) \mid c_j^1 = c_j^2 = 0 \text{ for } j = 1, \dots, m, c_j^3 = 0 \text{ for } j = 1, \dots, d \Big\},$$

$$\mathcal{D}_{\mathbb{R}^{2d_i}} = \Big\{ \big(((p, v), a), c_j^1 \partial x_M^j + c_j^2 \partial v_{M,1}^j + c_j^3 \partial a^j_{\mathbb{R}^{2d_i}} + c_j^4 \partial v^j_{\mathbb{R}^{2d_i}} + c_j^5 \partial v_{M,2}^j \big) \in T(\hat{F}_i^* T\mathbb{R}^{2d_i}) \oplus T(\pi_M^* TM) \mid c_j^5 = 0 \text{ for } j = 1, \dots, m \Big\},$$

and define projections $P_{\mathcal{D}_{\text{VB},i}}$ and $P_{\mathcal{D}_{\mathbb{R}^{2d_i}}}$ from the tangent pullback bundle $T(\breve{F}_i^* T(\mathbb{R}^{2d_i} \times M))$ onto these distributions. As with the subbundles, these distributions are smooth and globally well-defined due to their vertical bundle structures.

Now as before, we use the local PBDS construction above to form for each task a section of the tangent pullback bundle associating each robot position/velocity pair with the corresponding desired task pullback acceleration:

$$\begin{aligned} S_i : TM &\longrightarrow T(\breve{F}_i^* T(\mathbb{R}^{2d_i} \times M)) \\ (p, v) &\mapsto P_{\mathcal{D}_{\mathbb{R}^{2d_i}}} \circ P_{\mathcal{D}_{\text{VB},i}}(\gamma'_{(p,v)}(0)) \\ &= (((p, v), a), (0, (0, \gamma'^a_{(p,v)}(0), 0))). \end{aligned}$$

Next we map robot accelerations and jerks to their corresponding task tangent pullback bundle accelerations:

$$\begin{aligned} Z_i : TTTM &\longrightarrow T(\breve{F}_i^* T(\mathbb{R}^{2d_i} \times M)) \\ (((p, v), a), \kappa) &\mapsto P_{\mathcal{D}_{\mathbb{R}^{2d_i}}} \circ P_{\mathcal{D}_{\text{VB},i}}(d(\breve{F}_i^* d\breve{F}_i)_{((p,v),a)}(\kappa)). \end{aligned}$$

Now given a weighting pseudometric $\breve{w}_i$ over $T(\mathbb{R}^{2d_i} \times M)$ for each task, we can use a further higher-order task map $\mathfrak{F}_i : TTM \longrightarrow T(\mathbb{R}^{2d_i} \times M) : ((p, v), a) \mapsto (\breve{F}_i(p, v), (d\breve{F}_i)_{(p,v)}(a))$ to form the pullback pseudometric $\mathfrak{F}_i^* \breve{w}_i$ over TTM.

Then we can form our function returning the robot acceleration policy output for a given robot position and velocity:

$$\begin{aligned} \zeta : TM &\longrightarrow \mathcal{D} \subset TTTM \\ (p, v) &\mapsto \underset{\kappa \in \mathcal{D}_{(p,v)}}{\arg\min} \sum_{i=1}^{k} \frac{1}{2} \| Z_i(\kappa) - S_i(p, v) \|^2_{\mathfrak{F}_i^* \breve{w}_i}, \end{aligned} \tag{22}$$

where $\mathcal{D}$ is the globally well-defined distribution of $TTTM$ such that the subspace $\mathcal{D}_{(p,v)} \subseteq T_{((p,v),(v,a))} TTM$ satisfies $\kappa^v = v$ and $\kappa^v = \kappa^b = a$ componentwise for each $(((p, v), (v, a)), (\kappa^p, \kappa^v, \kappa^b, \kappa^a)) \in \mathcal{D}_{(p,v)}$. Note that because $w_{T\mathbb{R}^{2d_i}}$ depends only on the base points of $T\mathbb{R}^{2d_i}$, given a point $(p, v) \in TM$, the coordinates of the pullback pseudometric $\mathfrak{F}_i^* \breve{w}_i$ do not depend on the particular $((p, v), a) \in T_{(p,v)} TM$ at which $\mathfrak{F}_i^* \breve{w}_i$ is evaluated. Thus in practice we can simply evaluate it at $((p, v), 0) \in TTM$.

To derive the local form of this acceleration policy, notice that the differential $d(\breve{F}_i^* d\breve{F}_i)$ can be written locally as the

Jacobian

$$\boldsymbol{J}(\breve{\boldsymbol{F}}_i^* \boldsymbol{d}\breve{\boldsymbol{F}}_i) = \begin{bmatrix} \boldsymbol{I}_m & 0 & 0 & 0 \\ 0 & \boldsymbol{I}_m & 0 & 0 \\ \boldsymbol{J}\dot{\bar{\boldsymbol{f}}}_i & \boldsymbol{J}\dot{\bar{\boldsymbol{f}}}_i & \boldsymbol{J}\dot{\bar{\boldsymbol{f}}}_i & \boldsymbol{J}\bar{\boldsymbol{f}}_i \\ \boldsymbol{J}\dot{\bar{\boldsymbol{f}}}_i & 0 & \boldsymbol{J}\bar{\boldsymbol{f}}_i & 0 \\ 0 & 0 & 0 & \boldsymbol{I}_m \end{bmatrix},$$

where we denote $J\dot{\bar{f}}_i$ analogous to Appx. C. From this we can see that given a point having componentwise identifications $q = (((p,v),(v,a)),(v,a,a,\kappa)) \in \mathcal{D}_{(p,v)}$ we can locally compute

$$\begin{aligned} &\|Z_i(q) - S_i(p,v)\|^2_{\mathfrak{F}_i^* \breve{w}_i} \\ &= \|\boldsymbol{J}\bar{\boldsymbol{f}}_i(p)\boldsymbol{a} + \boldsymbol{J}\dot{\bar{\boldsymbol{f}}}_i(p,v)\boldsymbol{v} - \dot{\boldsymbol{\gamma}}^{\kappa}_{v_p,i}(0)\|^2_{\breve{\boldsymbol{w}}_i^a(\mathfrak{F}_i(v_p,0)))}, \end{aligned} \tag{23}$$

where $\dot{\boldsymbol{\gamma}}_{v_p} = (\dot{\boldsymbol{\gamma}}^v_{v_p}, \dot{\boldsymbol{\gamma}}^a_{v_p}, \dot{\boldsymbol{\gamma}}^b_{v_p}, \dot{\boldsymbol{\gamma}}^{\kappa}_{v_p}, \dot{\boldsymbol{\gamma}}^M_{v_p})$, $\breve{\boldsymbol{w}}_i^a$ is the lower-right quadrant of $\boldsymbol{w}_{T\mathbb{R}^{2d_i}}$, and we define $\mathfrak{F}_i^0 : TM \longrightarrow T(\mathbb{R}^{2d_i} \times M) : (\breve{F}_i(p,v),(0,v))$. Then adapting (10) and using the computed Christoffel symbols of (21), we can compute the components of $\dot{\boldsymbol{\gamma}}_{v_p,i}$ along the indices of interest $j = 2m + d_i + 1, \ldots, 2m + 2d_i$. First for forces, we have

$$\begin{aligned} &\Big(P_{\breve{F}_i^* G}\Big(\breve{F}_i^* \breve{\mathcal{F}}_i(\gamma_{v_p,i}(0))^\sharp\Big)\Big)^j \\ &\qquad = g^{jh}_{\mathbb{R}^{2d_i}}(\hat{F}_i(v_p))\Big(\breve{\mathcal{F}}^h_{D,i}(\gamma_{v_p,i}(0)) - \frac{\partial \breve{\Phi}_i}{\partial x^h}(\breve{F}_i(v_p))\Big). \end{aligned}$$

Then for the Christoffel symbols term, denoting $\boldsymbol{\gamma}_{v_p} = (\boldsymbol{\gamma}^v_{v_p}, \boldsymbol{\gamma}^a_{v_p}, \boldsymbol{\gamma}^M_{v_p})$ and $\dot{\alpha}_{v_p} = (\dot{\alpha}^v_{v_p}, \dot{\alpha}^a_{v_p})$ we have

$$\begin{aligned} &P_{\breve{F}_i^* G}\Big(\dot{\alpha}^{\ell}_{v_p,i}(0) P_{\breve{F}_i^* G}\Big(\gamma_{v_p,i}(0)\Big)^h \breve{F}_i^* \breve{\Gamma}^{j-2m}_{\ell h}(v_p)\Big) \\ &= P_{\breve{F}_i^* G}\Big(\dot{\alpha}^{\ell}_{v_p,i}(0) P_{\breve{F}_i^* G}\Big((\breve{F}_i^* J\breve{F}_i)_{hr}\dot{\alpha}^r_{v_p,i}(0)\Big) \\ &\quad (J\breve{F}_i)_{s\ell}(v_p)\breve{\Gamma}^{j-2m}_{sh}(\breve{F}_i(v_p))\Big) \\ &= (\dot{\alpha}^v_{v_p,i})^{\ell}(0)(J\bar{f}_i)_{hr}(p)(\dot{\alpha}^v_{v_p,i})^r(0) \\ &\qquad (J\hat{F}_i)_{s\ell}(v_p)\breve{\Gamma}^{j-2m}_{s(h+d)}(\hat{F}_i(v_p)) \\ &+ (\dot{\alpha}^a_{v_p,i})^{\ell}(0)(J\bar{f}_i)_{hr}(p)(\dot{\alpha}^v_{v_p,i})^r(0) \\ &\qquad (J\hat{F}_i)_{s(\ell+d)}(v_p)\breve{\Gamma}^{j-2m}_{s(h+d)}(\hat{F}_i(v_p)) \\ &= (\Xi^v_i)_{(j-2m-d_i)\ell}(v_p)(\dot{\alpha}^v_{v_p,i})^{\ell}(0) \\ &\quad + (\Xi^a_i)_{(j-2m-d_i)\ell}(v_p)(\dot{\alpha}^a_{v_p,i})^{\ell}(0), \end{aligned}$$

where we notice the Jacobian $\breve{F}_i^* J\breve{F}_i$ takes the local form

$$\breve{\boldsymbol{F}}_i^* \boldsymbol{J}\breve{\boldsymbol{F}}_i = \begin{bmatrix} \boldsymbol{J}\dot{\bar{\boldsymbol{f}}}_i & \boldsymbol{J}\bar{\boldsymbol{f}}_i & 0 \\ \boldsymbol{J}\bar{\boldsymbol{f}}_i & 0 & 0 \\ 0 & 0 & \boldsymbol{I}_m \end{bmatrix}.$$

From this can compute solutions to (22) locally as

$$\begin{aligned} \boldsymbol{\zeta}(p,v) = &\left(\sum_{i=1}^k \boldsymbol{\mathcal{C}}_i^\top(p,v)\breve{\boldsymbol{w}}_i^a(\mathfrak{F}_i(v_p,0))\boldsymbol{\mathcal{C}}_i(p,v)\right)^\dagger \\ &\left(\sum_{i=1}^k \boldsymbol{\mathcal{C}}_i^\top(p,v)\breve{\boldsymbol{w}}_i^a(\mathfrak{F}_i(v_p,0))\boldsymbol{\mathcal{A}}_i(p,v)\right) \end{aligned}$$

$$\boldsymbol{\mathcal{C}}_i(p,v) = \boldsymbol{J}\bar{\boldsymbol{f}}_i(p) + \boldsymbol{\Xi}^v_i(p,v)$$

$$\boldsymbol{\mathcal{A}_i}(p,v) = (\boldsymbol{g}^{-1}_{\mathbb{R}^{2d_i}})^a(\hat{F}_i(v_p))\Big(\boldsymbol{\mathcal{F}}_{D,\mathbb{R}^{2d_i}}((d\hat{F}_i)_{v_p}(v_p,0)) - \nabla\boldsymbol{\Phi}_{\mathbb{R}^{2d_i}}(\hat{F}_i(v_p))\Big) - (\dot{\boldsymbol{J}}\bar{\boldsymbol{f}}_i(p,v) - \boldsymbol{\Xi}^a_i(p,v))\boldsymbol{v}$$

where $(\boldsymbol{g}^{-1}_{\mathbb{R}^{2d_i}})^a$ is the lower-right quadrant of $\boldsymbol{g}^{-1}_{\mathbb{R}^{2d_i}}$ and the Euclidean gradient $\nabla\boldsymbol{\Phi}_{\mathbb{R}^{2d_i}}$ is taken over the last d_i components. Additionally we have

$$\begin{aligned} (\Xi^v_i)_{j\ell}(p,v) &= (J\hat{F}_i)_{s\ell}(v_p)(\Gamma_{\mathbb{R}^{2d_i}})^{j+d_i}_{s(h+m)}(\hat{F}_i(v_p))J\bar{f}_{hr}(p)v^r \\ (\Xi^a_i)_{j\ell}(p,v) &= (J\hat{F}_i)_{s(\ell+d_i)}(v_p)(\Gamma_{\mathbb{R}^{2d_i}})^{j+d_i}_{s(h+m)}(\hat{F}_i(v_p))J\bar{f}_{hr}(p)v^r. \end{aligned}$$

This leads to the multi-task PBDS policy for tangent bundle tasks:

Definition D.2 (Multi-Task PBDS on Tangent Bundles)**.** *Let $\{f_i : M \longrightarrow N_i\}_{i=1,\ldots,k}$ be smooth task maps, where N_i is embedded in $\mathbb{R}^d$. Then given corresponding Riemannian metrics $\breve{g}_i$ on $\mathbb{R}^{2d_i} \times M$, smooth potential functions $\breve{\Phi}_i : \mathbb{R}^{2d_i} \times M \longrightarrow \mathbb{R}_+$, dissipative forces $\breve{\mathcal{F}}_{D,i} : T(\mathbb{R}^{2d_i} \times M) \longrightarrow T^*(\mathbb{R}^{2d_i} \times M)$, and weighting pseudometrics $\breve{w}_i$ on $T(\mathbb{R}^{2d_i} \times M)$, the set $\{(f_i, g_i, \breve{\Phi}_i, \breve{\mathcal{F}}_{D,i}, \breve{w}_i)\}_{i=1,\ldots,k}$ forms a multi-task PBDS with curves $\sigma : [0,\infty) \longrightarrow M$ satisfying*

$$\begin{cases} \ddot{\sigma}(t) = \zeta(\dot{\sigma}(t)) = \\ \qquad \underset{\kappa \in \mathcal{D}_{\dot{\sigma}(t)}}{\arg\min} \ \sum_{i=1}^k \frac{1}{2}\|Z_i(\kappa) - S_i(\dot{\sigma}(t))\|^2_{\mathfrak{F}_i^* \breve{w}_i} \\ \dot{\sigma}(0) = (p_0, v_0). \end{cases} \tag{24}$$

Similar existence, uniqueness, smoothness, and stability guarantees can be given for this dynamical system as for that of Appx. C, but these are outside the scope of this appendix.