

Geometric Fabrics: Generalizing Classical Mechanics to Capture the Physics of Behavior

Karl Van Wyk^{1*}, Mandy Xie^{1,2*}, Anqi Li^{1,3}, Muhammad Asif Rana^{1,2}, Buck Babich¹, Bryan Peele¹, Qian Wan¹, Ireteayo Akinola¹, Balakumar Sundaralingam¹, Dieter Fox^{1,3}, Byron Boots^{1,3}, and Nathan D. Ratliff¹

*Equal contrib, ¹NVIDIA, ²Georgia Tech, ³University of Washington.

Abstract—Classical mechanical systems are central to controller design in energy shaping methods of geometric control. However, their expressivity is limited by position-only metrics and the intimate link between metric and geometry. Recent work on Riemannian Motion Policies (RMPs) has shown that shedding these restrictions results in powerful design tools, but at the expense of theoretical stability guarantees. In this work, we generalize classical mechanics to what we call geometric fabrics, whose expressivity and theory enable the design of systems that outperform RMPs in practice. Geometric fabrics strictly generalize classical mechanics forming a new physics of behavior by first generalizing them to Finsler geometries and then explicitly bending them to shape their behavior while maintaining stability. We develop the theory of fabrics and present both a collection of controlled experiments examining their theoretical properties and a set of robot system experiments showing improved performance over a well-engineered and hardened implementation of RMPs, our current state-of-the-art in controller design.

Index Terms—Dynamics, Optimization and Optimal Control

I. INTRODUCTION

CLASSICAL mechanical systems [1] are the workhorse of controller design in geometric control¹ [4]. Their popularity derives from both powerful conservation properties [5] lending to simply Lyapunov stability and convergence analysis [6] and their link to Riemannian geometry (see geometric mechanics [4, 7]), enabling the shaping of, for instance, straight-line end-effector behavior. However, classical mechanical systems have fundamentally limited expressivity.

When designing behavior in parts—such as combining target reaching with obstacle avoidance, joint limit avoidance, posturing, and speed regulation—these classical mechanical systems combine as metric-weighted averages of component policies (see Section III-B). Three limitations are clear from this perspective: (1) priority metrics depend only on position and not full system state; (2) the geometry is fundamentally linked to the metric making it impossible to design both independently; and (3) controllers, as a result, overly rely on potentials, dampers and unnatural task hierarchies. Recent work in Riemannian Motion Policies (RMPs) [8], demonstrated that removing the restrictions of classical mechanics can result in powerful design tools. However, RMPs sacrifice theoretical insight for expressivity, forcing practitioners to gain experience before becoming proficient.

Geometric fabrics, a new physics of behavior, generalize classical mechanics while retaining the powerful mathematics

of classical systems and promoting flexibility to circumvent the above issues. Geometric fabrics are what we call *bent* Finsler geometries [7]—the Finsler system generalizes the Riemannian geometry of classical systems giving us velocity-dependent metrics, and the bending terms enable the independent shaping of associated policies. Constraint forces are simple examples of bending terms commonly deployed to follow trajectories or trace surfaces [9], but our bending terms extend far beyond constraints. We derive fabrics that bend Finsler geometries to match any given nonlinear geometry using a technique called *geometry energization*.

One may view fabrics as a class of systems that instantaneously look like classical systems (with mass matrices, corresponding inertias, and forces), but which change from moment to moment as a function of state, similar to variable impedance systems; even concepts such as contact models generalize. In most cases, techniques for classical systems apply to fabrics, but now with the expressivity of RMPs.

Concretely, this work: (1) generalizes classical mechanics to capture the flexibility required for behavioral design; (2) develops an expressive class of RMPs that are stable, path consistent, and agnostic to parameterization (covariant); (3) develops a fabric component algebra analogous to the RMP algebra; and (4) presents experiments validating the theory and showing we can exploit the theoretical properties to outperform a well-tuned implementation of RMPflow.

We start by reviewing background in Section III and discuss the limitations of classical systems. We then define geometric fabrics in Section IV-B, show that constrained fabrics are themselves fabrics, and present a simple stability and convergence theorem. Subsection IV-D presents energization, showing that fabrics are in a sense complete and can take on the shape of any geometry. Finally, we discuss computation in generalized coordinates in Subsection IV-E and use those results to define an algebra of *fabric components* in Section V which can be used to derive computational algorithms. Section VI then analyzes empirically the advantages of geometric fabrics over standard unbent Riemannian and Finsler systems, and Section VII studies a full robot system implementation of fabrics, comparing it to a well-engineered and hardened implementation of RMPs.

The appendix and our website² contains extended discussion, proofs, and additional experimental demonstrations.

¹Geometric control encompasses operational space control [2] and potential field methods [3]; we refer to them all here as geometric control.

²<https://sites.google.com/nvidia.com/geometric-fabrics-behavior>

II. RELATED WORK

Second-order systems are natural for control since robots, themselves, are second-order systems (see [10] for similar arguments), so we restrict our attention here to those.

Geometric control [4, 2, 9], which uses energy shaping of classical mechanical systems for behavioral design, is the most directly related area. We discuss the limitations of classical systems in Section III-B and propose geometric fabrics to eliminate them. Energy shaping methods can leverage this broader class of systems for more expressive design. Dynamic Movement Primitives (DMPs) [10] are another closely related area. DMPs operate by perturbing stable systems with a forcing term that either vanishes or is periodic—the resulting system inherits its stability from the underlying stable system. This idea is again highly-compatible with geometric fabrics since we can swap out the underlying stable system with any stable second-order system, including geometric fabrics (see Appendix B-A for details). Optimal control [11] is another relevant area and again highly compatible. Any optimal controller optimizes over a dynamics function. Traditionally, that dynamics function is derived from classical mechanics, but fabrics are a viable generalized alternative that would enable building behavior into the dynamics to aid the optimization.

An alternative approach to fabrics, developed independent of this work, is the Pullback Bundle Dynamical System (PBDS), derived as a class of stable Riemannian systems on the tangent bundle [12]. PBDSs separate priority metric and policy design but in turn require task policies to be strictly Riemannian, a subset of the HD2 geometries of fabrics. Contact dynamics is also less clear, although such extensions are likely possible. Another related area is Control Lyapunov Functions (CLF) [13] which constitute a general technique for stabilizing systems. CLFs work by projecting unstable systems onto a Lyapunov stable class introducing potentially large projection errors. With fabrics we directly design inherently stable nonlinear systems so the final system both captures the desired behavior and is stable by construction. Finally, earlier work introduced Geometric Dynamical Systems (GDS) [8] as a stable class of RMPs. We now know that GDSs are better expressed as Finsler systems which are unbent fabrics. Appendix B-B details this relation and Subsection VI shows empirically their limitations.

Other control approaches, including λ_0 -PMP models [14], are first-order differential IK models related to classical Riemannian systems but lack the second-order fidelity and velocity dependence we address here. Fractal impedance methods [15] define locally optimal controllers that asymptotically track simultaneous position and force profiles. However, these are specialized controllers unlike geometric fabrics which describe a very general class of second order dynamical systems without any explicit notion of trajectory tracking or specific objectives. Moreover, composable energy policies [16] optimize actions over a product of stochastic policies to resolve multiple policy conflicts. Geometric fabrics address this issue as well through geometric policies. Alternative works try to further resolve multi-policy conflicts with hierarchical prioritization and null-space projections [17, 2, 18]. However,

these structures are overly strict and relative priorities should instead continuously adjust.

Finally, Finsler and spray geometry in mathematics [19] are, of course, highly relevant. That literature, however, remains insurmountably challenging for most roboticists. We work from the re-derivations presented in [7]; it's an area of future work to lower the barrier-to-entry and leverage related ideas from the literature. To the best of our knowledge geometric fabrics and results on constrained fabrics and energization are novel, as are the fabric component algebra and resulting tools for putting them into practice.

III. BACKGROUND

A. Finsler and general nonlinear geometries

We briefly review some of the main ideas behind Finsler and general nonlinear geometries of paths (Homogeneous of Degree 2 (HD2) geometries)—see [7] for details. We call any smooth differential equation $\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{0}$ an HD2 geometry when \mathbf{h} is HD2 in velocities so that $\mathbf{h}(\mathbf{q}, \alpha\dot{\mathbf{q}}) = \alpha^2\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$. We often write it in resolved form $\ddot{\mathbf{q}} = -\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \pi(\mathbf{q}, \dot{\mathbf{q}})$ and refer to $\pi(\mathbf{q}, \dot{\mathbf{q}})$ as a *geometric policy* to give policy semantics. The HD2 property enforces that integral curves are *path consistent*, i.e., they follow geometric speed-independent paths [7]. *Finsler geometries* are HD2 geometries defined by their minimum length geodesics using the calculus of variations. Their generalized arc-length measures are defined by a Finsler structure $\mathcal{L}_g(\mathbf{q}, \dot{\mathbf{q}})$:

Definition III.1. A Finsler structure is a Lagrangian with the following three properties:

- 1) $\mathcal{L}_g(\mathbf{q}, \dot{\mathbf{q}}) \geq 0$ with equality if and only if $\dot{\mathbf{q}} = \mathbf{0}$.
- 2) \mathcal{L}_g is positive homogeneous (HD1) in $\dot{\mathbf{q}}$ so that $\mathcal{L}_g(\mathbf{q}, \alpha\dot{\mathbf{q}}) = \alpha\mathcal{L}_g(\mathbf{q}, \dot{\mathbf{q}})$ for $\alpha \geq 0$.
- 3) $\partial_{\dot{\mathbf{q}}\dot{\mathbf{q}}}^2\mathcal{L}_e \succeq 0$ when $\dot{\mathbf{q}} \neq \mathbf{0}$ where $\mathcal{L}_e = \frac{1}{2}\mathcal{L}_g^2$.

The first two properties enforce that $\mathcal{L}_g(\mathbf{q}, \dot{\mathbf{q}})$ is a speed-independent measure of length (see [7]). The third property ensures the equations of motion of the associated *Finsler energy* \mathcal{L}_e , derived from the Euler-Lagrange equation as $\frac{d}{dt}\partial_{\dot{\mathbf{q}}}\mathcal{L}_e - \partial_{\mathbf{q}}\mathcal{L}_e = \mathbf{M}(\mathbf{q}, \dot{\mathbf{q}})\ddot{\mathbf{q}} + \boldsymbol{\xi}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{0}$, with $\mathbf{M} = \partial_{\dot{\mathbf{q}}\dot{\mathbf{q}}}^2\mathcal{L}_e$ (mass / metric) and $\boldsymbol{\xi} = \partial_{\mathbf{q}\mathbf{q}}\mathcal{L}_e\dot{\mathbf{q}} - \partial_{\mathbf{q}}\mathcal{L}_e$ (geometric terms), are well-formed with positive definite mass. These (closed) equations of motion (absent potential, damping, or external forces) define an HD2 (Finsler) geometry [7].

A simple example of a Finsler geometry is Riemannian geometry. The Riemannian Finsler structure measures length $\mathcal{L}_g(\mathbf{q}, \dot{\mathbf{q}}) = (\dot{\mathbf{q}}^T\mathbf{M}(\mathbf{q})\dot{\mathbf{q}})^{\frac{1}{2}} = \|\dot{\mathbf{q}}\|_{\mathbf{M}}$ using a symmetric positive definite metric $\mathbf{M}(\mathbf{q})$ dependent on position only. The corresponding energy is the well-known classical kinetic energy $\mathcal{L}_e = \frac{1}{2}\mathcal{L}_g^2 = \frac{1}{2}\|\dot{\mathbf{q}}\|_{\mathbf{M}}^2 = \frac{1}{2}\dot{\mathbf{q}}^T\mathbf{M}(\mathbf{q})\dot{\mathbf{q}}$ for mass \mathbf{M} . This link between geometry and system energy is reflected more generally in the link between Finsler geometry and Finsler energy, with equations of motion tracing geodesics.

Every Lagrangian system has an associated conserved energy (its Hamiltonian) $\mathcal{H} = \partial_{\dot{\mathbf{q}}}\mathcal{L}^T\dot{\mathbf{q}} - \mathcal{L} = \mathbf{p}^T\dot{\mathbf{q}} - \mathcal{L}$ where $\mathbf{p} = \partial_{\dot{\mathbf{q}}}\mathcal{L}$ is the *generalized momentum*. When the Lagrangian is a Finsler energy $\mathcal{L}_e = \frac{1}{2}\mathcal{L}_g^2$ with equations of motion $\mathbf{M}(\mathbf{q}, \dot{\mathbf{q}})\ddot{\mathbf{q}} + \boldsymbol{\xi}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{0}$, the Hamiltonian can be expressed

$\mathcal{H}_e = \mathcal{L}_e = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M} \dot{\mathbf{q}}$, showing that \mathcal{L}_e itself is the energy and can be expressed in the classical form [7].

The Finsler metric $\mathbf{M}(\mathbf{q}, \dot{\mathbf{q}})$ is Homogeneous of Degree 0 (HD0) which means $\mathbf{M}(\mathbf{q}, \alpha \dot{\mathbf{q}}) = \mathbf{M}(\mathbf{q}, \dot{\mathbf{q}})$. \mathbf{M} is, therefore, dependent on velocity direction $\hat{\mathbf{q}}$ but not on speed. That means, for any system $\mathbf{M}(\mathbf{q}, \dot{\mathbf{q}}) \ddot{\mathbf{q}} + \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{0}$ (where \mathbf{f} may capture more than just ξ), the resolved equation $\ddot{\mathbf{q}} = -\mathbf{M}^{-1} \mathbf{f}$ defines a geometric policy if and only if \mathbf{f} is HD2.

B. Energy Shaping with Classical Mechanical Systems

Classical mechanical systems have traditionally been important tools for energy shaping in controller design [4]. The main idea of energy shaping is to design a virtual classical mechanical system that captures a desired behavior, then control the physical system to behave like it thereby inheriting the stability of the virtual system. This idea is powerful which is why it has been so important and popular in control. But we show here that classical systems have limited expressivity as tools for behavioral design.

Let \mathcal{Q} be a generalized coordinate domain constrained by $\mathbf{C}(\mathbf{q}) = \mathbf{0}$ with Jacobian $\mathbf{J}(\mathbf{q})$. A constrained classical mechanical system evolves according to the equation [1]:

$$\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \xi(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{J}(\mathbf{q})^T \lambda(\mathbf{q}, \dot{\mathbf{q}}) = -\partial_{\mathbf{q}} \psi(\mathbf{q}) - \mathbf{B}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}},$$

where $\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \xi(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{0}$ form the equations of motion of an unforced system with kinetic energy $\mathcal{K}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}}$ (the term $\xi(\mathbf{q}, \dot{\mathbf{q}})$ captures fictitious forces which we call here *geometric forces* in reference to their role in Finsler geometry), the function $\psi(\mathbf{q})$ is a potential function, $\mathbf{B}(\mathbf{q}, \dot{\mathbf{q}})$ is a positive definite damping matrix, and $\lambda(\mathbf{q}, \dot{\mathbf{q}})$ are the Lagrange multipliers of the constraint force. With $\mathbf{B} = \mathbf{0}$ the system conserves total energy $\mathcal{E}(\mathbf{q}, \dot{\mathbf{q}}) = \mathcal{K}(\mathbf{q}, \dot{\mathbf{q}}) + \psi(\mathbf{q})$; with nonzero \mathbf{B} the energy dissipates, and the system converges to a local minimum of the potential on the constrained domain and is therefore inherently stable.

Unfortunately, a simple analysis shows that these systems have limited expressivity. Consider multiple classical systems $\mathbf{M}_i(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{f}_i(\mathbf{q}, \dot{\mathbf{q}}) = \boldsymbol{\tau}_i$, where now \mathbf{f}_i captures all forces excluding constraints (fictitious, potential, and damping—constraint forces are often added to the combined system after summing, so we ignore them here) and we single out $\boldsymbol{\tau}_i$ as the force the other systems place on system i . For instance, these component systems may express reaching a target, avoiding obstacles, joint limit avoidance, and posturing of the arm. The internal forces $\boldsymbol{\tau}_i$ must sum to zero $\sum_i \boldsymbol{\tau}_i = \mathbf{0}$, so we have $\sum_i (\mathbf{M}_i \ddot{\mathbf{q}} + \mathbf{f}_i) = \sum_i \boldsymbol{\tau}_i = \mathbf{0}$. Assuming each \mathbf{M}_i is full rank, we can write $\pi_i = -\mathbf{M}_i^{-1} \mathbf{f}_i$ and express the resulting combined acceleration as³

$$\ddot{\mathbf{q}} = \left(\sum_i \mathbf{M}_i \right)^{-1} \sum_i \mathbf{M}_i \pi_i, \quad (1)$$

which is simply a metric-weighted average of constituent policies π_i , where each policy takes the form:

$$\pi_i(\mathbf{q}, \dot{\mathbf{q}}) = -\mathbf{M}_i^{-1} \left(\xi_i(\mathbf{q}, \dot{\mathbf{q}}) + \partial_{\mathbf{q}} \psi_i(\mathbf{q}) + \mathbf{B}_i(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} \right).$$

³A similar result holds if \mathbf{M}_i is not full rank. Intuitively, we can add a small ϵ to the diagonal then take the limit $\epsilon \rightarrow 0$ in Equation 1.

Each component policy is defined by geometric forces, its potential function, and its damping matrix; they averaged together using the mass matrices as spectral priority weights (with Eigenvalues expressing different priorities along each Eigenvector). Three limitations manifest from this analysis.

First, priority metrics are functions of position only and not full position-velocity state. Many tasks are best expressed with velocity aware priorities which classical metrics cannot represent. E.g. any barrier avoidance task (obstacle, self, joint limit avoidance) should care about the barrier while moving toward it, but should forget about it when moving away.

Second, geometric forces can be powerful—they define the Riemannian geometry of the system associated with the metric $\mathbf{M}(\mathbf{q})$ [4, 7]—but in classical systems they are intrinsically linked to the metric which also plays the role of the policy's priority matrix. Only one of the two can be designed, the other becomes an artifact. This coupling results in fighting between components (when the metric is compromised) or forces an over-reliance on potentials and dampers (when the geometry is compromised). An example of a Riemannian geometry working against the policy is the robot's own (unshaped) inertia which will fight a potential. Likewise, straight-line end-effector motion is a useful shaped geometry, but that defines a largely irrelevant priority metric.

Third, when the geometric forces cannot be used, we must shape the behavior entirely using potential functions and dampers. Potential functions add to the total potential energy affecting where local minima fall resulting in task conflict. Moreover, potentials are only position-dependent and can induce spring-like oscillations when damping is insufficient. Alternatively, one can increase damping, or even entirely shape the behavior using dampers, but that can slow the system when multiple contributions combine. Geometric forces, on the other hand, are path consistent and will shape the behavior without shifting the system local minima.

Note that the second and third limitations are largely the motivation for hierarchical task prioritization [18]—when multiple potentials may conflict or priority metrics are compromised, designers preserve task fidelity by instituting a task hierarchy whereby lower-priority tasks are projected into the null-space of higher-priority tasks [17, 2]. However, these task hierarchies are often pragmatic solutions to these fundamental problems and somewhat artificial in practice. For instance, the relative priorities of reaching targets and avoiding obstacles should adjust continuously; it is unclear which is globally higher priority.

Geometric fabrics remove these limitations by (1) introducing velocity-dependent metrics using Finsler geometries, and (2) decoupling system geometry from priority metric using bending terms. Now, the system's behavior is primarily directed by the fabric, the potential is applied for a specific task, and damping for convergence and speed control.

IV. GEOMETRIC FABRICS: A NEW PHYSICS OF BEHAVIOR

Geometric fabrics are a strict generalization of classical mechanics (with classical systems being a type of fabric) which retain the powerful mathematical properties exploited in

control while removing the expressivity limitations outlined in Subsection III-B. In particular, fabrics' priority metrics depend on full state and are decoupled from their associated geometric policy. Stability of these systems is easily analyzed using energy conservation (see Theorem IV.4). We derive fabrics as bent Finsler geometries using what we call *bending terms* which generalize constraint forces. These bending terms are powerful and can be used to bend a Finsler geometry to align with any desired HD2 geometry using a process we call geometry *energization*. We additionally derive formulas for calculation in generalized coordinates which we use in Section V to construct an algebra of fabric components to aid systems engineering.

A. Bending Finsler geometries

An important concept in geometric fabrics is the *bending* of a Finsler geometry. Geometric bending is implemented using system modification terms that both perform no work (conserve energy) and are geometric (HD2). An example of such a bending term is the classical constraint force that keeps a system moving along a constraint surface. These forces are zero-work by design (see the Principal of Virtual Work [1]) and are known to preserve the geometric character of the system. Bending terms, as defined and characterized in the following Lemma, generalize classical constraints.

Lemma IV.1. *Let \mathcal{L}_e be a Finsler energy with unforced equations of motion $\mathbf{M}(\mathbf{q}, \dot{\mathbf{q}})\ddot{\mathbf{q}} + \boldsymbol{\xi}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{0}$, and let $\psi(\mathbf{q})$ be a potential function and $\mathbf{B}(\mathbf{q}, \dot{\mathbf{q}})$ be a positive definite damping matrix. The modified system $\mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\xi} + \mathbf{f}_f = -\partial\psi - \mathbf{B}\dot{\mathbf{q}}$ with modifying term $\mathbf{f}_f(\mathbf{q}, \dot{\mathbf{q}})$ has total energy $\mathcal{H} = \mathcal{L}_e + \psi$ that decreases at a rate*

$$\dot{\mathcal{H}} = -\dot{\mathbf{q}}^T \mathbf{B}\dot{\mathbf{q}} \quad (2)$$

if and only if $\dot{\mathbf{q}}^T \mathbf{f}_f = \mathbf{0}$ everywhere. We call such a term \mathbf{f}_f a zero-work modification. When \mathbf{f}_f is additionally HD2, the unforced system (with $\psi = 0$ and $\mathbf{B} = \mathbf{0}$) is geometric and we say \mathbf{f}_f is a bending term that bends the Finsler geometry. Moreover, if ψ is lower bounded, \mathcal{H} is lower bounded.

Proof. See Appendix D-A. \square

Bending terms are geometric forces that are everywhere orthogonal to the direction of motion and thereby conserve energy by performing no work.

B. Geometric fabrics

Geometric fabrics generalize classical systems in two steps: (1) they generalize the classical Riemannian geometries to the broader class of Finsler geometries; and (2) they bend the resulting Finsler geometry using bending terms which include, but are not limited to, constraint forces.

Definition IV.2 (Geometric fabric). Let $\mathcal{L}_e = \frac{1}{2}\mathcal{L}_g^2$ be a Finsler energy derived from Finsler structure \mathcal{L}_g with equations of motion $\mathbf{M}(\mathbf{q}, \dot{\mathbf{q}})\ddot{\mathbf{q}} + \boldsymbol{\xi}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{0}$. A geometric fabric is a system that evolves according to

$$\mathbf{M}(\mathbf{q}, \dot{\mathbf{q}})\ddot{\mathbf{q}} + \boldsymbol{\xi}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{f}_f(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{0}, \quad (3)$$

where $\mathbf{f}_f(\mathbf{q}, \dot{\mathbf{q}})$ is any bending term. Geometric fabrics are denoted $\mathcal{F} = (\mathcal{L}_e, \mathbf{f}_f)$.

The fabric defines a nominal behavior independent of a specific task. Potential functions then push away from that nominal behavior to drive the system toward task goals, and damping dissipation ensures convergence. The full equations of motion of the system then become:⁴

$$\begin{aligned} \mathbf{M}(\mathbf{q}, \dot{\mathbf{q}})\ddot{\mathbf{q}} + \boldsymbol{\xi}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{f}_f(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{J}(\mathbf{q})^T \boldsymbol{\lambda}(\mathbf{q}, \dot{\mathbf{q}}) \\ = -\partial_{\mathbf{q}}\psi(\mathbf{q}) - \mathbf{B}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}, \end{aligned} \quad (4)$$

where $\psi(\mathbf{q})$ is a potential function, $\mathbf{B}(\mathbf{q}, \dot{\mathbf{q}})$ is a positive definite damping matrix, and $\boldsymbol{\lambda}$ are the Lagrange multipliers of constraint forces with constraint Jacobian \mathbf{J} . We see many similarities to the classical equations given in Subsection III-B. The potential function injects energy into the system and drives it toward its local minimum, while the damper bleeds energy off for stability. Lemma IV.3 characterizes constrained geometric fabrics and shows that a constrained geometric fabric is itself a geometric fabric. Theorem IV.4 proves stability of these systems.

Lemma IV.3. *Let $(\mathcal{L}_e, \mathbf{f}_f)$ be a geometric fabric with Finsler energy \mathcal{L}_e , bending term \mathbf{f}_f , and associated equations of motion $\mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\xi} + \mathbf{f}_f = -\partial\psi - \mathbf{B}\dot{\mathbf{q}}$ for potential ψ and damping matrix \mathbf{B} . Under equality constraint $\mathbf{C}(\mathbf{q}) = \mathbf{0}$, assuming $(\mathbf{q}, \dot{\mathbf{q}})$ already satisfy the constraint, the resulting constrained equations of motion are*

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{P}_{\parallel}(\boldsymbol{\xi} + \mathbf{f}_f) + \mathbf{P}_{\perp}\mathbf{M}\dot{\mathbf{J}}\dot{\mathbf{q}} = -\mathbf{P}_{\parallel}(\partial\psi + \mathbf{B}\dot{\mathbf{q}}) \quad (5)$$

where, $\mathbf{J} = \partial_{\mathbf{q}}\mathbf{C}$ denotes the constraint Jacobian, $\mathbf{P}_{\perp} = \mathbf{J}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}\mathbf{J}\mathbf{M}^{-1}$ projects orthogonally to the constraints, and $\mathbf{P}_{\parallel} = \mathbf{I} - \mathbf{P}_{\perp}$ projects parallel to the constraints. Importantly, the left hand side defines a fabric $(\mathcal{L}_e, \tilde{\mathbf{f}}_f)$ where $\tilde{\mathbf{f}}_f = \mathbf{P}_{\perp}(\mathbf{M}\dot{\mathbf{J}}\dot{\mathbf{q}} - \boldsymbol{\xi}) + \mathbf{P}_{\parallel}\mathbf{f}_f$ is a constrained fabric.

Proof. See Appendix D-B. \square

The constrained system in Equation 5 is intuitive. $\mathbf{P}_{\parallel}(\boldsymbol{\xi} + \mathbf{f}_f)$ projects the original fabric to operate parallel to the constraint and $-\mathbf{P}_{\parallel}(\partial\psi + \mathbf{B}\dot{\mathbf{q}})$ does the same with the forcing terms. $\mathbf{P}_{\perp}\mathbf{M}\dot{\mathbf{J}}\dot{\mathbf{q}}$ independently creates the required curvature force to keep the system moving along the constraint, modifying the original fabric's bending term.

C. Stability analysis

The following theorem presents a general result on the stability and convergence of forced geometric fabrics.

Theorem IV.4 (Geometric fabric stability). *Suppose $\mathcal{F} = (\mathcal{L}_e, \mathbf{f}_f)$ is geometric fabric operating under equality constraints $\mathbf{c}(\mathbf{q}) = \mathbf{0}$ and inequality constraints $\mathbf{g}(\mathbf{q}) \geq \mathbf{0}$ (both of which may be empty). Then for any $\psi(\mathbf{q})$ lower-bounded on the constrained domain and any bounded positive-definite damping matrix $\mathbf{B}(\mathbf{q}, \dot{\mathbf{q}})$ the system evolving according to*

⁴A broader theory of *optimization* fabrics [20] drops the geometric requirement but retains the optimization properties outlined in Theorem IV.4. Geometric fabrics are a type of fabric that most naturally generalizes classical mechanics. We refer to geometric fabrics simply as fabrics.

Equation 4 (with \mathbf{C} consisting of \mathbf{c} and the active subset of \mathbf{g}) converges to a local minimum of the optimization problem

$$\min_{\mathbf{q} \in \mathcal{Q}} \psi(\mathbf{q}) \quad \text{s.t.} \quad \mathbf{c}(\mathbf{q}) = \mathbf{0} \quad \text{and} \quad \mathbf{g}(\mathbf{q}) \geq \mathbf{0} \quad (6)$$

under any contact model contributing as equality constraints with dissipation when in contact and conserving or dissipating energy on impact.

Proof. See Appendix D-C \square

D. Completeness: Energizing HD2 geometries

The next theorem is central to the theory of fabrics. It shows fabrics are complete in the sense that they can be bent to align with any given geometry. Completeness shows we have independent control over priority metric design (Finsler metric) and geometric policy design (desired geometry).

Theorem IV.5 (Energized fabrics). *Let $\ddot{\mathbf{q}} = \pi(\mathbf{q}, \dot{\mathbf{q}})$ be an HD2 geometry, and suppose \mathcal{L}_e is a Finsler energy with equations of motion $\mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\xi} = \mathbf{0}$ and energy $\mathcal{H}_e = \mathcal{L}_e$. Then $\ddot{\mathbf{q}} = \pi(\mathbf{q}, \dot{\mathbf{q}}) + \alpha_{\mathcal{H}_e} \dot{\mathbf{q}}$ is energy conserving when*

$$\alpha_{\mathcal{H}_e} = -(\dot{\mathbf{q}}^T \mathbf{M} \dot{\mathbf{q}})^{-1} \dot{\mathbf{q}}^T [\mathbf{M} \boldsymbol{\pi} + \boldsymbol{\xi}], \quad (7)$$

and differs from the original system by only an acceleration along the direction of motion. The new system is a geometric fabric, known as an energized fabric, expressed as:

$$\mathbf{M} \ddot{\mathbf{q}} + \boldsymbol{\xi} - \mathbf{P}_e [\mathbf{M} \boldsymbol{\pi} + \boldsymbol{\xi}] = \mathbf{0}, \quad (8)$$

where $\mathbf{P}_e = \mathbf{M}^{\frac{1}{2}} [\mathbf{I} - \hat{\mathbf{v}} \hat{\mathbf{v}}^T] \mathbf{M}^{-\frac{1}{2}}$ is a metric-weighted projection matrix with $\mathbf{v} = \mathbf{M}^{\frac{1}{2}} \dot{\mathbf{x}}$ and $\hat{\mathbf{v}} = \mathbf{v} / \|\mathbf{v}\|$.

Proof. See Appendix D-D. \square

We denote the energization operation using an operator $\text{energize}_{\mathcal{L}_e}[\cdot]$ that outputs a geometric fabric given an HD2 geometry π so that

$$\text{energize}_{\mathcal{L}_e}[\pi] = \left(\mathcal{L}_e, -\mathbf{P}_e [\mathbf{M} \boldsymbol{\pi} + \boldsymbol{\xi}] \right) \quad (9)$$

is the energized geometry. Note that the energization operator can be applied to a non-geometric policy as well. In that case, we write the output in the same form using the same energization coefficient calculation in Equation 7, but with the knowledge that the resulting transformed system will conserve energy but not be a geometric fabric.

An informative example is when $\mathcal{L}_e = \frac{1}{2} \|\dot{\mathbf{q}}\|^2$ measures the squared velocity norm. Here $\mathbf{M} = \mathbf{I}$ and $\boldsymbol{\xi} = \mathbf{0}$, so under Equation 7 the system reduces to

$$\ddot{\mathbf{q}} = \left[\mathbf{I} - \hat{\mathbf{q}} \hat{\mathbf{q}}^T \right] \pi(\mathbf{q}, \dot{\mathbf{q}}), \quad (10)$$

where $\hat{\mathbf{q}} = \dot{\mathbf{q}} / \|\dot{\mathbf{q}}\|$ is the normalized velocity. In other words, the energized system projects accelerations of the geometric policy π orthogonal to the direction of motion so they curve the system without changing its speed.

E. Constraints in generalized coordinates

We showed above that we can construct fabrics by energizing geometries (see Theorem IV.5) and constraining them

(see Theorem IV.3). There we calculated explicit constraint forces in the ambient space for analysis, but in many cases, it's natural to express constrained systems in generalized coordinates such as the joint angles of the robot. This section presents a fundamental result showing how to calculate an energized policy in generalized coordinates.

First we formally derive how to constrain systems using generalized coordinates. Let $\mathbf{M}(\mathbf{x}, \dot{\mathbf{x}}) \ddot{\mathbf{x}} + \mathbf{f}(\mathbf{x}, \dot{\mathbf{x}}) = \mathbf{0}$ be any system of this form defined on a space \mathcal{X} (\mathbf{M} need not derive from a Finsler energy and \mathbf{f} captures all system forces). Suppose $\phi : \mathcal{Q} \rightarrow \mathcal{X}$ defines a surface in \mathcal{X} with generalized coordinates \mathcal{Q} and Jacobian $\mathbf{J} = \partial_{\mathbf{q}} \phi$ (note that the rows of \mathbf{J} here are orthogonal to those of the implicit surface Jacobian due to the differing representation). We have $\mathbf{x} = \phi(\mathbf{q})$, $\dot{\mathbf{x}} = \mathbf{J} \dot{\mathbf{q}}$, and $\ddot{\mathbf{x}} = \mathbf{J} \ddot{\mathbf{q}} + \dot{\mathbf{J}} \dot{\mathbf{q}}$; plugging that last expression for $\ddot{\mathbf{x}}$ into the system equation and multiplying through by \mathbf{J} gives an expression for the constrained system in generalized coordinates:

$$\left(\mathbf{J}^T \mathbf{M} \mathbf{J} \right) \ddot{\mathbf{q}} + \mathbf{J}^T (\mathbf{f} + \mathbf{M} \dot{\mathbf{J}} \dot{\mathbf{q}}) = \mathbf{0}. \quad (11)$$

This equation has the same form as the original system in \mathcal{X} , but with $\mathbf{M} \rightarrow \mathbf{J}^T \mathbf{M} \mathbf{J}$ and $\mathbf{f} \rightarrow \mathbf{J}^T (\mathbf{f} + \mathbf{M} \dot{\mathbf{J}} \dot{\mathbf{q}})$. This transformation is known as the *pullback* of the system into the generalized coordinates, and can be expressed as an operation on system parameters

$$\text{pull}_{\phi}[(\mathbf{M}, \mathbf{f})] = \left(\mathbf{J}^T \mathbf{M} \mathbf{J}, \mathbf{J}^T (\mathbf{f} + \mathbf{M} \dot{\mathbf{J}} \dot{\mathbf{q}}) \right). \quad (12)$$

For Lagrangian systems, this pullback operation is consistent with the generalized coordinate calculation of the Euler-Lagrange equation. Specifically, if $\mathcal{L}(\mathbf{x}, \dot{\mathbf{x}})$ is any Lagrangian in \mathcal{X} , we can define

$$\text{pull}_{\phi}[\mathcal{L}] = \mathcal{L}(\phi(\mathbf{q}), \frac{d}{dt} \phi(\mathbf{q})). \quad (13)$$

Then if $\text{EL}[\mathcal{L}] = (\mathbf{M}, \boldsymbol{\xi})$ denotes the application of the Euler-Lagrange equation to construct equations of motion $\mathbf{M} \ddot{\mathbf{x}} + \boldsymbol{\xi} = \mathbf{0}$, we have

$$\text{pull}_{\phi}[\text{EL}[\mathcal{L}]] = \text{EL}[\text{pull}_{\phi}[\mathcal{L}]] = \text{pull}_{\phi}[(\mathbf{M}, \boldsymbol{\xi})]. \quad (14)$$

The following theorem reveals that a constrained energized system in generalized coordinates can be constructed by first constraining the metric-weighted geometry then energizing directly in the generalized coordinates.

Theorem IV.6. *Let $\mathcal{L}_e^{\mathcal{X}}(\mathbf{x}, \dot{\mathbf{x}})$ be a Finsler energy on \mathcal{X} with equations of motion $\mathbf{M} \ddot{\mathbf{x}} + \boldsymbol{\xi} = \mathbf{0}$ and let $\pi(\mathbf{x}, \dot{\mathbf{x}})$ be an HD2 geometry on \mathcal{X} . If $\phi : \mathcal{Q} \rightarrow \mathcal{X}$ is a full rank differentiable map defining a constraint in \mathcal{X} with generalized coordinates \mathcal{Q} , then*

$$\text{pull}_{\phi}[\text{energize}_{\mathcal{L}_e^{\mathcal{X}}}[\pi]] = \text{energize}_{\mathcal{L}_e^{\mathcal{Q}}}[\text{pull}_{\phi}[(\mathbf{M}, -\mathbf{M} \boldsymbol{\pi})]],$$

where $\mathcal{L}_e^{\mathcal{Q}} = \text{pull}_{\phi}[\mathcal{L}_e^{\mathcal{X}}]$ and $(\mathbf{M}, -\mathbf{M} \boldsymbol{\pi})$ denotes the parameters of the metric weighted geometry $\mathbf{M}(\ddot{\mathbf{x}} - \pi(\mathbf{x}, \dot{\mathbf{x}})) = \mathbf{0}$.

Proof. See Appendix D-E. \square

V. DESIGNING WITH FABRICS

Theorem IV.6 is fundamental to fabric design because it enables us to design using transform trees. A transform tree is a tree of differentiable maps, where each node is a task space and each edge is a differentiable map mapping a parent space (domain) to a child space (co-domain). The tree-structure expresses sparsity patterns important for computational efficiency, but conceptually we can think of the tree as a single differentiable map linking the root space to the stacked joint output space of all nodes. From that view it's clear that we can exploit Theorem IV.6 to say that the resulting constrained system constraining the joint ambient space fabric is equivalent to the system defined by pulling the ambient space fabric to the root. In light that, it's often useful to design geometric fabrics in parts, viewing each part as an HD2 geometric policy π with an associated priority metric defined by a Finsler energy \mathcal{L}_e , each living on a node space of a transform tree. Then computationally we can pull the pieces back to the root exploiting sparsity in the same way as in earlier work [8]. When designing we can reason about what the geometry of each part should be and how we should prioritize it, understanding that they will combine a metric-weighted average of geometries.

Let $\mathcal{G} = \{\mathcal{L}, \mathbf{f}\}$ compactly denote a system of the form $\mathbf{M}\ddot{\mathbf{x}} + \mathbf{f} = \mathbf{0}$ paired with a Finsler energy from which \mathbf{M} derives. We call \mathcal{G} a *fabric component*. Note that this system by itself is not a geometric fabric, which must be a bent Finsler system. Here \mathbf{f} can be arbitrary. For instance, it often denotes $\mathbf{f} = -\mathbf{M}\pi$ derived from geometric policy π .

The scaled sum of two fabric components $\mathcal{G}_1 = \{\mathcal{L}_1, \mathbf{f}_1\}$ and $\mathcal{G}_2 = \{\mathcal{L}_2, \mathbf{f}_2\}$ is given by

$$\alpha\mathcal{G}_1 + \beta\mathcal{G}_2 = \{\alpha\mathcal{L}_1 + \beta\mathcal{L}_2, \alpha\mathbf{f}_1 + \beta\mathbf{f}_2\}, \quad (15)$$

where $\alpha, \beta \in \mathbb{R}_+$. Likewise, the pullback of a fabric component across a map $\mathbf{x} = \phi(\mathbf{q})$ with Jacobian \mathbf{J} is

$$\text{pull}_\phi[\{\mathcal{L}, \mathbf{f}\}] = \left\{ \text{pull}_\phi[\mathcal{L}], \mathbf{J}^T(\mathbf{f} + \mathbf{M}\dot{\mathbf{J}}\dot{\mathbf{q}}) \right\}. \quad (16)$$

See Appendix E-A for formal derivations of these formulas. These expressions are easily derived using the basic definitions of Finsler energy pullback and fabric pullback from Section IV-E. It is easy to show that summation is linear (which implies associativity and commutativity), and pullback across the composition of functions is equal to the pullback across each $\text{pull}_{\phi_2 \circ \phi_1}[\mathcal{G}] = \text{pull}_{\phi_1}[\text{pull}_{\phi_2}[\mathcal{G}]]$.

Given a transform tree, we populate the nodes with these fabric components and exploit this algebra to recursively pull each component from child to parent and sum the results at the parent (with optional scalar weights). What results is a final fabric component at the root $\tilde{\mathcal{G}} = \{\tilde{\mathcal{L}}, \tilde{\mathbf{f}}\}$ and associated Finsler energy $\tilde{\mathcal{L}}$. $\tilde{\mathcal{G}}$ represents system $\tilde{\mathbf{M}}\ddot{\mathbf{q}} + \tilde{\mathbf{f}} = \mathbf{0}$ in generalized coordinates defining an HD2 geometric policy $\tilde{\pi} = -\tilde{\mathbf{M}}^{-1}\tilde{\mathbf{f}}$ since it is formed as a metric-weighted average of pullback geometries from each component (and the metrics are HD0). The final fabric is simply the energization of this geometry with the associated Finsler energy:

$$\mathcal{F} = \text{energize}_{\tilde{\mathcal{L}}}[\tilde{\pi}]. \quad (17)$$

In practice, we also leverage the geometric path consistency of the final geometric policy to regulate speed without affecting the path traced during execution (see [7]).

A. Speed control through damping regulation

Because we have the flexibility now to encode the nominal behavior into the fabric itself we can leverage accelerations along the direction of motion (known to leave the geometric paths unchanged) to design dampers that regulate speed without affecting the behavior. We use dampers of the form

$$\mathbf{f}_{\text{damp}} = -\beta(\mathbf{q}, \dot{\mathbf{q}})\mathbf{M}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}, \quad (18)$$

with $\beta \in \mathbb{R}_+$ so that the resulting acceleration $\mathbf{M}^{-1}\mathbf{f}_{\text{damp}} = -\beta\dot{\mathbf{q}}$ acts solely along the direction of motion. The potential then has a dual role to (1) speed the system (the acceleration component parallel to the direction of motion) and (2) force the system off a nominal path toward the goal (the acceleration component orthogonal to the direction of motion). See Appendix E-B for a discussion on how to calculate β to effectively control a target measure of speed.

B. Tools for fabric design

Finsler energies and HD2 geometries share a common requirement of being Homogeneous of Degree 2 (HD2). Appendix E-C presents a number of methods for designing such functions which we use throughout our experiments. At a high-level these tend to revolve around designing HD0 terms (which can rely on velocity directionality but not the norm (speed)), and scaling them by some measure of Finsler energy (such as the squared norm of the velocity) to bring them from HD0 to HD2. We use these techniques to design the fabrics used in our experiments as described in Sections VI and VII-A.

VI. PARTICLE EXPERIMENTS

Geometric fabrics have two parts to their generalization: (1) moving from Riemannian geometries to Finsler geometries; (2) bending those Finsler geometries. These experiments explore how both of those pieces affect the system by comparing four types of systems: (i) strictly Riemannian systems (classical mechanical systems); (ii) standard Finsler systems (enabling velocity-dependent metrics, but without bending terms); (iii,iv) bent variants of each of the first two. In the discussion below, we often refer to both Riemannian and Finsler systems as *unbent fabrics* and the bent variants as geometric fabrics; there are four combinations in total: unbent Riemannian, unbent Finsler, bent Riemannian, bent Finsler (this last one being a full geometric fabric).

We start with a collection of particles at rest to the right of a circular obstacle and we observe the behavior of different variants controlled to two different speeds attracting the particles to a point to the left of the obstacle (small square) under the influence of obstacle avoidance.

A. Point Attraction

This term uses task map $\mathbf{x} = \phi(\mathbf{q}) = \mathbf{q} - \mathbf{q}_d$ where $\mathbf{q}, \mathbf{q}_d \in \mathbb{R}^2$ are the current and desired particle position in Euclidean space. The metric is designed as $\mathbf{G}_\psi(\mathbf{x}) = (\bar{m} - m)e^{-(\alpha_m \|\mathbf{x}\|)^2} \mathbf{I} + m\mathbf{I}$, where $\bar{m}, m \in \mathbb{R}^+$ are the upper and lower isotropic masses, respectively, and $\alpha_m \in \mathbb{R}^+$ controls the width of the radial basis function. We design the potential gradient directly as $\partial_{\mathbf{q}}\psi(\mathbf{x}) = \mathbf{M}_\psi(\mathbf{x})\partial_{\mathbf{q}}\psi_1(\mathbf{x})$, with $\psi_1(\mathbf{x}) = k\left(\|\mathbf{x}\| + \frac{1}{\alpha_\psi} \log(1 + e^{-2\alpha_\psi \|\mathbf{x}\|})\right)$, where $k \in \mathbb{R}^+$ controls the overall gradient strength, $\alpha_\psi \in \mathbb{R}^+$ controls the transition rate of $\partial_{\mathbf{q}}\psi_1(\mathbf{x})$ from a positive constant to 0. For the unbent fabric, the system derives from $\mathcal{L} = \dot{\mathbf{x}}^T \mathbf{G}_\psi(\mathbf{x})\dot{\mathbf{x}} - \psi(\mathbf{x})$. For the geometric fabric, we use fabric component defined by geometry $\ddot{\mathbf{x}} = -\partial_{\mathbf{x}}\psi_1(\mathbf{x})$ and energy $\mathcal{L}_e = \dot{\mathbf{x}}^T \mathbf{G}_\psi(\mathbf{x})\dot{\mathbf{x}}$.

B. Circular Object Repulsion

This term uses task map $x = \phi(\mathbf{q}) = \frac{\|\mathbf{q} - \mathbf{q}_o\|}{r} - 1$, where \mathbf{q}_o is the origin of the circle and r is its radius. Two different metrics are designed to prioritize object repulsion, $G_b(x) = \frac{k_b}{x^2}$ and $G_b(x, \dot{x}) = s(\dot{x})\frac{k_b}{x^2}$. Moreover, $k_b \in \mathbb{R}^+$ is a barrier gain and $s(\dot{x}) = 1$ if $\dot{x} < 0$ and $s(\dot{x}) = 0$, otherwise. We design the potential gradient as $\partial_{\mathbf{q}}\psi_b(x) = M_b(x)\partial_{\mathbf{q}}\psi_{1,b}(x)$ with $\psi_{1,b}(x) = \frac{\alpha_b}{2x^3}$, and $\alpha_b \in \mathbb{R}^+$. For the unbent fabric, the system derives from either $\mathcal{L}_1 = G_b(x)\dot{x}^2 - \psi_b(x)$ (Riemannian) or $\mathcal{L}_2 = G_b(x, \dot{x})\dot{x}^2 - \psi_b(x)$ (Finsler). For the geometric fabric, we use fabric component defined by $\ddot{x} = -s(\dot{x})\dot{x}^2\partial_x\psi_{1,b}(x)$ and energy either \mathcal{L}_1 or \mathcal{L}_2 .

C. Discussion

Fig. 1 visualizes the results of all for variants each under two different speeds $v_d = 2, 4$ to analyze degree of consistency under speed changes. For each variant, all particles avoid the object and reach the target position except the one shot directly at the object’s local minimum. However, we observe several differences in behavior.

First, the unbent fabrics (left column) have more pronounced changes in paths across the different speed levels. This is amplified for Finsler systems (bottom two rows) using $G_b(x, \dot{x})$ since the velocity gate does not modulate the obstacle avoidance policy. Instead, the mass of the obstacle avoidance policy vanishes, while components of its force remain. This effect is amplified when traveling at a higher velocity, producing the “launching” artifacts. In contrast, geometric fabrics (right column) produce more consistent paths across speed levels without any launching artifacts.

Second, Finsler metrics (bottom two rows) facilitate straight-line motion to the desired location once past the obstacle by allowing the system to forget about it as a function of directionality. Riemannian metrics are unable to represent that directional dependence (top two rows) and can be seen clinging to the obstacle even once past it.

Finally, the repulsive forces from the unbent fabrics (right column) consistently push the center particle farther out from the object than with geometric fabrics (right column). Importantly, geometric fabrics allow for heightened obstacle avoidance behavior without shifting the system minima.

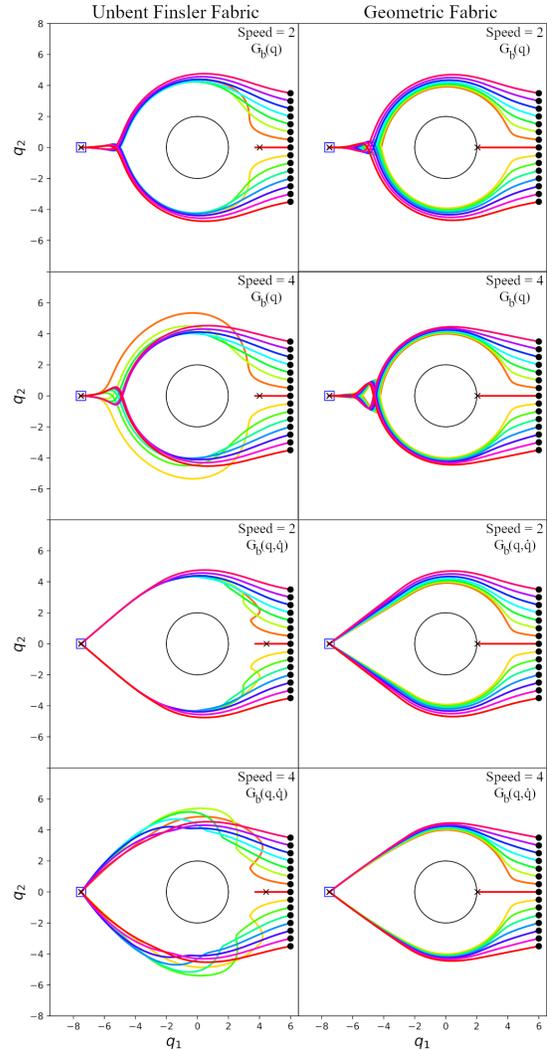


Fig. 1. Particle behavior for an unbent fabric and a geometric fabrics with different metric designs. See Section VI-C for discussion.

VII. FRANKA PANDA EXPERIMENTS

Our second set of experiments is a study of geometric fabrics on a full 7 degree-of-freedom robot arm. The geometric fabric is designed once, and tested across a variety of problems. We compare its performance to our current, best performing Riemmanian Motion Policy (RMP) that has powered our manipulation processes for a variety of settings. Both policies are evaluated at 100 Hz and integrated with first-order Euler routines. Note that since this is purely a motion generation problem, the sim-to-real gap is minimal since smooth trajectories can easily be executed on physical robots using standard trajectory following controllers. We, therefore, focus only a simulated analysis here.

A. Geometric Fabrics Design

The geometric fabric policy ultimately consists of only three differently designed components. Each component consists of a Finsler energy $\mathcal{L}_e(\mathbf{x}, \dot{\mathbf{x}})$ and policy $\pi(\mathbf{x}, \dot{\mathbf{x}})$.

1) *Joint Space Attraction*: The Finsler energy is designed as $\mathcal{L}_e = \frac{m}{2}\|\dot{\mathbf{x}}\|^2$, where $m \in \mathbb{R}^+$ is the amount of isotropic mass. The policy is designed as $\pi = -\|\dot{\mathbf{x}}\|^2\partial_{\mathbf{x}}\psi(\mathbf{x})$ with $\psi(\mathbf{x}) = k \log(\cosh(\alpha(\mathbf{x}_d - \mathbf{x})))$ where $k, \alpha \in \mathbb{R}^+$ are gains

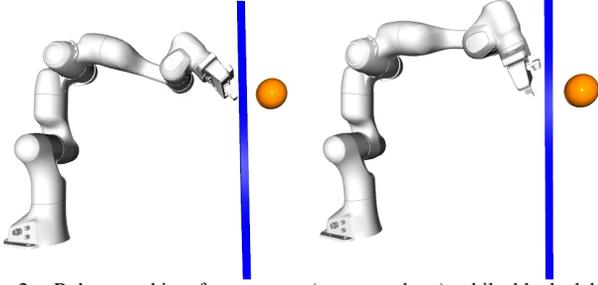


Fig. 2. Robot reaching for a target (orange sphere) while blocked by an obstacle (blue) under a geometric fabric policy (left) and an RMP (right).

that control scaling and sharpness, respectively, and \mathbf{x} and \mathbf{x}_d are the current and desired joint space positions. This component acts to posture the arm largely within the null space of higher-priority tasks such as target reaching.

2) *Distance Space Repulsion*: The Finsler energy is designed as $\mathcal{L}_e = \frac{k}{2x}s(\dot{x})\dot{x}^2$, where $k \in \mathbb{R}^+$ scales the barrier function and $s(\dot{x}) = 1$ if $\dot{x} < 0$ and $s(\dot{x}) = 0$, otherwise. We define a potential $\psi(x) = \frac{k_b}{x} + \frac{k_r}{\alpha} \log(1 + e^{-\alpha(x-x_o)})$, where $k_b, k_r \in \mathbb{R}^+$ control the scaling of the barrier and soft rectified linear unit (ReLU) functions, respectively and $x \in \mathbb{R}^+$ is a signed distance space (e.g., distance to object). Moreover, $\alpha \in \mathbb{R}^+$ controls the sharpness in the ReLU and $x_o \in \mathbb{R}^+$ shifts the activation of the ReLU. We deploy this potential alongside a geometric variant $\pi = -\dot{x}^2 \partial_x \psi(x)$, for joint-limit avoidance, self-collision avoidance, and object-collision avoidance. The geometric policies are weighed significantly higher than potentials, which play the role of soft-penalty on the obstacle constraint here. The potential policy is included to shift the system minima away from numerical singularities, e.g., as $x \rightarrow 0$, then $\dot{x} \rightarrow 0$ and $\partial_x \psi(x) \rightarrow \infty$, which generates numerical issues for calculating the geometric policy.

3) *End-effector Space Attraction*: Using $\bar{m}, \underline{m} \in \mathbb{R}^+$ to denote maximum and minimum isotropic masses, respectively, the Finsler energy is designed as $\mathcal{L}_e = \|\dot{\mathbf{x}}\|^2 (\frac{1}{2}(\bar{m} - \underline{m})(\tanh(-\alpha\|\mathbf{x}_d - \mathbf{x}\|) + 1) + \underline{m})$, where $\alpha \in \mathbb{R}^+$ controls the sharpness of the hyperbolic tangent function, and $\mathbf{x}, \mathbf{x}_d \in \mathbb{R}^3$ are the position and target position of the end-effector, respectively. The potential is defined as $\psi(\mathbf{x}) = k \log(\cosh(\alpha(\mathbf{x}_d - \mathbf{x})))$, where $k, \alpha \in \mathbb{R}^+$ are gains that control scaling and sharpness, respectively. We deploy both the potential and a geometric variant $\pi = -\|\dot{\mathbf{x}}\|^2 \partial_{\mathbf{x}} \psi(\mathbf{x})$. Each are individually tuned, with the geometric variant being generally more active and the potential there to ensure convergence in alignment with our theory. Applying this design to two axes at the end-effector can also realize orientation control as in [21].

B. Wall Barrier Approach

To assess how well the policies can reach 3D end-effector position targets in the presence of boundary constraints, we incrementally advance the end-effector target every five seconds towards a wall boundary and monitor convergence (see Fig. 2). Ideally, a policy should accurately converge to the target while respecting the boundary constraint, but in practice boundary potentials can prevent that. In this experiment, we

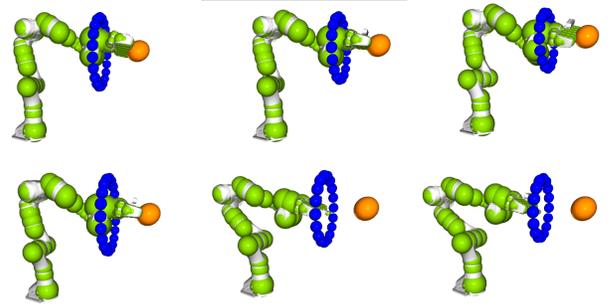


Fig. 3. Robot executing reach to target (orange spheres) through a series of obstacle rings (blue) of increasingly smaller diameter with a geometric fabric (top row) and RMP (bottom row). Green spheres approximate robot geometry for collision avoidance.

see geometric fabrics converge more accurately in general to these targets than the RMP policy. Both policies exhibit sub-millimeter convergence error when the target is 10 cm or more from the wall, but convergence incrementally degrades as the target approaches the wall due to conflicting potentials. RMPs already start at degrading at ~ 9 cm while geometric fabrics begin degrading only at ~ 6 cm since much of the boundary policy is encoded in a geometric policy. Moreover, geometric fabrics on average approach the wall about 50% closer than the RMP. Fig. 2 depicts this difference in convergence near the wall.

C. Ring Constricted Navigation

To assess convergence to end-effector targets in the presence of increasingly constricted passages, we create a sequence of rings with incrementally smaller radii that the end-effector must pass through to reach its target. The ring radius starts at 17 cm and decreases to 13 cm. The rings are constructed from small spheres (each with radius 2.5 cm) as shown in Fig. 3. Distance spaces are pair-wise between robot spheres and obstacle spheres. As shown, the navigation from a geometric fabric allows the robot to successfully pass through all rings and reach its target. In contrast, the RMP cannot pass through the last two rings of the smallest diameter. Again, this difference arises with primarily encoding avoidance into the fabric's geometry alongside a diminished repulsive potential function.

D. Dynamic Obstacle Reaching

By nature, both the geometric fabric policy and RMP are reactive. To assess the quality of reactivity, we create a reaching task with both static and dynamic obstacles. The goal is for the policies to reach 19 randomly generated end-effector targets (with one target issued every 5 seconds) while avoiding collision with both static and dynamic obstacles (see Fig. 4). The policies are only aware of the object positions while objects achieve speeds of up to 0.39 m/s. With the geometric fabric policy, the robot reached 16 targets whereas the robot with the RMP reached 11 targets. Moreover, the geometric fabric policy yielded a 0.4% collision rate in contrast to the RMP's 8.18%, where collision rate is the percentage of time steps in collision. Since the geometric fabric obstacle

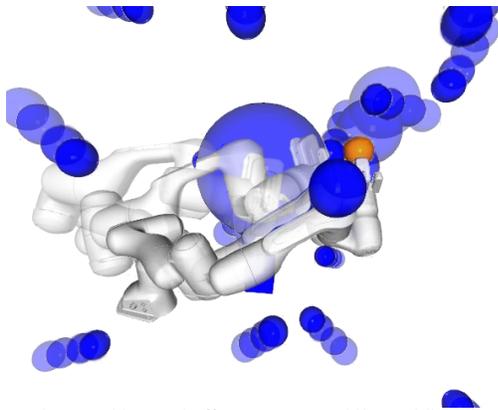


Fig. 4. Robot reaching end-effector targets while avoiding dynamic objects.

avoidance component is predominantly geometric in nature, it can simultaneously be highly influential while not preventing optimization of the end-effector potential, resulting in higher target acquisition rates. The forcing component of the geometric fabric policy aggressively engages only when the robot is close to collision. Altogether, this allows strong obstacle avoidance behavior both close to and far from objects in a way that minimally impedes target acquisition. In contrast, the RMP had to strike a balance between forcing repulsion without impairing the ability to acquire end-effector targets too much. This compromise resulted in diminished performance.

VIII. CONCLUSION

Geometric fabrics naturally generalize classical mechanical systems to form a new physics of behavior. These systems provide important design expressivity while maintaining the theoretical properties that make classical systems powerful in control. Many existing techniques in the literature can naturally leverage fabrics (e.g., reinforcement learning as in [22], imitation learning, planning), and it is a point of future work to explore many of these areas. Our theoretical and experimental analyses already demonstrate their utility but we believe the bulk of their application has yet to come.

REFERENCES

- [1] J. R. Taylor, *Classical Mechanics*. University Science Books, 2005.
- [2] O. Khatib, “A unified approach for motion and force control of robot manipulators: The operational space formulation,” *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.
- [3] —, “Real-time obstacle avoidance for manipulators and mobile robots,” in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, Mar 1985, pp. 500–505.
- [4] F. Bullo and A. D. Lewis, *Geometric control of mechanical systems: modeling, analysis, and design for simple mechanical control systems*. Springer Science & Business Media, 2004, vol. 49.
- [5] L. Susskind, *The Theoretical Minimum: Classical Mechanics*. Stanford: Continuing Studies, 2011. [Online]. Available: <http://theoreticalminimum.com/courses/classical-mechanics/2011/fall>
- [6] H. Khalil, *Nonlinear Systems*, ser. Pearson Education. Prentice Hall, 2002.
- [7] N. Ratliff, K. Van Wyk, M. Xie, A. Li, and M. Rana, “Generalized nonlinear and finsler geometry for robotics,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.
- [8] 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 *The 13th Intl. Workshop on the Algorithmic Foundations of Robotics*, 2018.
- [9] J. Peters, M. Mistry, F. Udwadia, J. Nakanishi, and S. Schaal, “A unifying framework for robot control with redundant DOFs,” *Autonomous Robots*, vol. 24, no. 1, pp. 1–12, 2008.
- [10] 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, Feb 2013.
- [11] R. Stengel, *Optimal Control and Estimation*. Dover, New York, 1994.
- [12] A. Bylard, R. Bonalli, and M. Pavone, “Composable geometric motion policies using multi-task pullback bundle dynamical systems,” in *Proc. IEEE Conf. on Robotics and Automation*, Xi’an, China, may 2021. [Online]. Available: <https://arxiv.org/abs/2101.01297>
- [13] A. Li, C. Cheng, B. Boots, and M. Egerstedt, “Stable, concurrent controller composition for multi-objective robotic tasks,” in *IEEE Conference on Decision and Control (CDC)*, 2019.
- [14] P. Tommasino and D. Campolo, “Task-space separation principle: a force-field approach to motion planning for redundant manipulators,” *Bioinspiration & biomimetics*, vol. 12, no. 2, p. 026003, 2017.
- [15] C. Tiseo, V. Ivan, W. Merkt, I. Havoutis, M. Mistry, and S. Vijayakumar, “A passive navigation planning algorithm for collision-free control of mobile robots,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 8223–8229.
- [16] J. Urain, A. Li, P. Liu, C. D’Eramo, and J. Peters, “Composable energy policies for reactive motion generation and reinforcement learning,” *arXiv preprint arXiv:2105.04962*, 2021.
- [17] L. Sentis and O. Khatib, “A whole-body control framework for humanoids operating in human environments,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2006, pp. 2641–2648.
- [18] A. Dietrich, C. Ott, and J. Park, “The hierarchical operational space formulation: Stability analysis for the regulation case,” *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1120–1127, 2018.
- [19] I. Bucataru and R. Miron, *Finsler-Lagrange Geometry: Applications to dynamical systems*. Editura Academiei Romane, 2007.
- [20] N. D. Ratliff, K. V. Wyk, M. Xie, A. Li, and A. M. Rana, “Optimization fabrics for behavioral design,” *arXiv:2010.15676 [cs.RO]*, 2020.
- [21] N. D. Ratliff, J. Issac, D. Kappler, S. Birchfield, and D. Fox, “Riemannian motion policies,” *arXiv:1801.02854*, 2018.
- [22] A. Li, C.-A. Cheng, M. A. Rana, M. X. K. V. Wyk, N. Ratliff, and B. Boots, “RMP2: A structured composable policy class for robot learning,” in *Robotics: Science and Systems (RSS)*, 2021.
- [23] J. Lee, *Introduction to Smooth Manifolds*, 2nd ed. Springer, 2012.
- [24] F. E. Udwadia and R. E. Kalaba, *Analytical Dynamics: A New Approach*. Cambridge University Press, 1996.
- [25] J. M. Lee, *Riemannian Manifolds: An Introduction To Curvature*. Springer, 1997.
- [26] P. Flores, M. Machado, E. Seabra, and M. Tavares da Silva, “A parametric study on the baumgarte stabilization method for forward dynamics of constrained multibody systems,” *Journal of computational and nonlinear dynamics*, vol. 6, no. 1, 2011.
- [27] D. Stewart and J. C. Trinkle, “An implicit time-stepping scheme for rigid body dynamics with coulomb friction,” in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 1. IEEE, 2000, pp. 162–169.

- [28] M. Xie, K. V. Wyk, A. Li, M. A. Rana, D. Fox, B. Boots, and N. Ratliff, “Geometric fabrics for the acceleration based design of robotic motion,” *arXiv:2010.14750 [cs.RO]*, 2020. [Online]. Available: <https://arxiv.org/abs/2010.14750>
- [29] M. Xie, A. Li, K. V. Wyk, F. Dellaert, B. Boots, and N. Ratliff, “Imitation learning via simultaneous optimization of policies and auxiliary trajectories,” 2021.

APPENDIX A PRELIMINARIES

A. Derivations on manifolds

Nonlinear geometry is most commonly developed in the mathematics literature in terms of smooth manifolds, and often in an abstract coordinate-free notation which can be challenging for practitioners. Even tensor notation can be a barrier to entry for readers unfamiliar with the algebra. To make our presentation more accessible, we will stick to coordinate descriptions and simpler vector notations from advanced calculus (see Appendix A-C) similar to many books on classical mechanics aimed at an engineering audience.

Manifolds will be formally defined in the traditional way (see [23] for a complete introduction), but for our purposes, we will consider them simply as d -dimensional spaces \mathcal{Q} with elements identified with $\mathbf{q} \in \mathbb{R}^d$ in d generalized coordinates, and when we say $\mathbf{q} \in \mathcal{Q}$ we mean \mathbf{q} is the coordinate representation of the manifold point described in our chosen coordinates. Often we implicitly assume a system evolves over time t in a trajectory $\mathbf{q}(t)$ with velocity $\dot{\mathbf{q}} = \frac{d\mathbf{q}}{dt}$, and say that coordinate velocity vector $\dot{\mathbf{q}}$ is an element of the tangent space $\mathcal{T}_{\mathbf{x}}\mathcal{Q}$, which can be thought of as the space of all velocity vectors that can be generated in that way. When discussing general tangent space vectors (elements of the tangent space without reference to a specific trajectory), we often use the notation $\mathbf{v} \in \mathcal{T}_{\mathbf{q}}\mathcal{Q}$.

B. A note on covariance and transform trees

Geometric consistency across changes of coordinates on a manifold is an important concept since many manifolds cannot be described globally by a single set of coordinates. Algorithms and systems should not change their fundamental behavior on the manifold based on the coordinate system they’re described in. When an algorithm or system has such a coordinate independence, it’s called *covariant*.

It is well known that the Euler-Lagrange equation is covariant to changes of coordinate [5], so anything derived here from the Euler-Lagrange equation is automatically covariant. For everything else, such as HD2 geometries and bending terms, we are not rigorous with developing specific intrinsically covariant subclasses. Such covariant subclasses exist and can be derived, but the mathematics behind that can be complicated and unnecessarily restrictive. In practice, we instead use transform trees to derive covariant systems.

A transform tree is simply a tree whose nodes are different spaces and whose edges are the maps linking those spaces. For instance, the simplest tree is a single branch $\phi : \mathcal{Q} \rightarrow \mathcal{X}$ mapping generalized coordinates $\mathcal{Q} \subset \mathbb{R}^d$ to a single space $\mathcal{X} \subset \mathbb{R}^n$. In fact, all trees can be expressed in this form since

paths from the root to any given node are unique and the maps encountered along that path can be combined into a single composed map. The single branch tree is constructed by stacking all nodes into a single space and defining a map by stacking these unique path-composed maps into a single map. In practice, more expressive tree structures are important for computational performance, but conceptually it suffices to think of simply a single link ϕ as described here. (Note that $n \geq d$ when the tree is well-formed so that policies on the nodes can fully constrain the space.)

Equation 14 shows that the pullback operation defined on a transform tree is compatible with the pullback implicit in the covariant Euler-Lagrange equation. This is because pullback is compatible with the definition of covariance. Calculationally, we prove that a system is covariant if, under a change of coordinates, the resulting system is simply scaled by the Jacobian (transpose) of the change of coordinates map. For instance, if ϕ denotes a change of coordinates (i.e. when $n = d$ and ϕ is smooth and invertible making it what is known as a *diffeomorphism*), a term $\mathbf{h}(\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}})$ is covariance if its expression in \mathcal{Q} amounts to $\tilde{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = \mathbf{J}^T \mathbf{h}(\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}})$ where $\mathbf{J} = \partial\phi$ is the Jacobian of ϕ and $\dot{\mathbf{J}}$ is its time derivative (and $\mathbf{x} = \phi(\mathbf{q})$, $\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}$, $\ddot{\mathbf{x}} = \mathbf{J}\ddot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{q}}$). The utility of this definition of covariance is that if $\tilde{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = \mathbf{0}$ then in \mathcal{Q} we get the same

$$\begin{aligned} \tilde{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) &= \mathbf{0} \\ \Rightarrow \mathbf{J}^T \mathbf{h}(\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}) &= \mathbf{0} \\ \Rightarrow \mathbf{h}(\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}) &= \mathbf{0} \end{aligned} \quad (19)$$

since \mathbf{J} is invertible when ϕ is a change of coordinates. In other words, covariant equations hold in all coordinate systems.

Now, looking at Equation 12 and its derivation leading to Equation 11, we see that when ϕ is a change of coordinates, the system transforms covariantly by definition. Specifically, when the term is $\mathbf{h}(\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}) = \mathbf{M}(\mathbf{x}, \dot{\mathbf{x}})\ddot{\mathbf{x}} + \mathbf{f}(\mathbf{x}, \dot{\mathbf{x}})$, the pullback operation computes $(\mathbf{J}^T \mathbf{M} \mathbf{J})\ddot{\mathbf{q}} + (\mathbf{f} + \mathbf{M} \dot{\mathbf{J}} \dot{\mathbf{q}}) = \mathbf{J}^T (\mathbf{M} \ddot{\mathbf{x}} + \mathbf{f}) = \mathbf{J} \mathbf{h}(\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}})$ (we’re just reversing the argument used to derive it). Therefore, if we have any arbitrary transform tree $\phi : \mathcal{Q} \rightarrow \mathcal{X}$ (viewed as a single branch) and we want to change the coordinates using $\zeta : \tilde{\mathcal{Q}} \rightarrow \mathcal{Q}$, we can simply build a transform tree linking these transforms

$$\tilde{\mathcal{Q}} \xrightarrow{\zeta} \mathcal{Q} \xrightarrow{\phi} \mathcal{X} \quad (20)$$

and use standard pullback operations to get covariance automatically. For instance, if \mathcal{Q} is the generalized coordinates of the robot, we can define a transform tree with \mathcal{Q} at the root and a collection of task spaces extending from it such as the end-effector space, body points, distances to obstacles, etc. Then, if we change to a different coordinate representation of the generalized coordinates we simply express that as part of the transform tree and the fundamental behavior of our system does not change because of the natural covariance of the transform tree itself. An important example is when the generalized coordinates includes a free flying SE3 base such as in a drone with an SO3 rotation component. SO3 cannot be globally expressed in a single coordinate system, so we inevitably need to change coordinates at times. (We can also

change representations, between Euler angles, quaternions, axis-angle and matrix representations, as long as the transform is represented as part of the tree.) Covariance ensures that the system behavior remains consistent under these representational transformations.

Another way to understand the transform tree is that it represents an optimization problem. Pullback can equivalently be viewed as the solution to a least squares optimization problem

$$\begin{aligned} \min_{\dot{\mathbf{q}}} \frac{1}{2} \|\ddot{\mathbf{x}}^d - \ddot{\mathbf{x}}\|_{\mathbf{M}}^2 \\ \text{s.t. } \ddot{\mathbf{x}} = \mathbf{J}\ddot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{q}}, \end{aligned} \quad (21)$$

analogous to the least squares formulation of classical mechanics under Gauss's Principle of Least Constraint [24]. Because it's an optimality principle, the fundamental minimum does not change under changes of coordinates.

C. Multivariate calculus notation

We avoid tensor notation here whenever possible in favor of the more compact and familiar matrix-vector notation of advanced calculus.

We assume each partial derivative generates a new index ranging over the partials. The partial $\partial_{\mathbf{x}}f(\mathbf{x})$ of a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is treated as a column vector. And if $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is a map, $\mathbf{g}(\mathbf{x})$ itself is treated as a column vector and $\partial_{\mathbf{x}}\mathbf{g}(\mathbf{x}) \in \mathbb{R}^{m \times n}$ denotes a matrix with partials ranging over the columns. When there are multiple partials as in $\partial_{\mathbf{x}\mathbf{y}}h(\mathbf{x}, \mathbf{y})$, we apply the partials in the order listed. In this case, we apply \mathbf{x} first to create a column vector $\partial_{\mathbf{x}}h(\mathbf{x}, \mathbf{y})$ then apply \mathbf{y} to create a matrix $\partial_{\mathbf{y}}(\partial_{\mathbf{x}}h)$. When more partials are involved, the resulting object becomes a higher-order tensor, but we use the convention that multiplication on the right by a matrix or vector contracts the indices in the reverse order they were generated. For instance, if $b(\mathbf{x}, \mathbf{y}, \mathbf{z})$ is a function, $\partial_{\mathbf{x}\mathbf{y}\mathbf{z}}b$ is a three index tensor, but $\partial_{\mathbf{x}\mathbf{y}\mathbf{z}}b\mathbf{v}$ will unambiguously contract along the \mathbf{z} partial first leaving a matrix whose rows align with the partials along \mathbf{x} and whose columns align with the partials along \mathbf{y} . Specifically, $\partial_{\mathbf{x}\mathbf{y}\mathbf{z}}b\mathbf{v} = \sum_k \frac{\partial}{\partial z^k} (\partial_{\mathbf{x}\mathbf{y}}b)v^k$. This convention is applied recursively when more than three variables are involved.

APPENDIX B

ADDITIONAL COMMENTS ON RELATED WORK

A. Generalization of Dynamic Movement Primitive (DMP) principles

In the Related Work section (see Section II), we discussed how Dynamic Movement Primitives (DMPs) are perturbations from a stable second-order differential equation and, in principle, can easily be generalized to leverage stable differential equations beyond simple linear equations. We describe that connection in more detail here. Studying these generalized perturbed systems is an area of future work, but we note here that geometric fabrics are not inherently competitive with the principles of DMPs, but rather complimentary.

A DMP is a nonlinear force perturbation away from an intrinsically stable linear second-order differential equation

[10]. When the force perturbation vanishes over time the system dynamics is asymptotically governed by the linear differential equation and therefore becomes a point attractor; likewise, when the nonlinear force is periodic the system converges to a stable limit cycle. In all cases, the key innovation is perturbing a stable differential equation using either a vanishing or cyclic perturbation so that stability of the combined system is governed by the stability of the underlying differential equation.

These ideas can easily be generalized to geometric mechanics noting that inherently stable classical mechanical systems can be used as a drop-in replacement for the linear second-order differential equation to inherit stability. Since geometric fabrics are a generalization of those classical mechanical systems, the same applies for fabrics. Geometric fabrics can act as a stable underlying nonlinear second-order differential equation that we can perturb away from using a trained perturbation component generalizing the techniques studied in DMPs while gaining more expressive underlying system behaviors from this broader class of geometric fabric equations.

B. Geometric Dynamical Systems vs Finsler Systems

Geometric Dynamical Systems (GDSs) were introduced in [8] as a stable class of RMPs for stable system design. But we now understand GDSs are more elegantly and concretely expressed as Finsler systems characterized in Subsection III-A. Moreover, such systems are inherently covariant since they derive from the Euler-Lagrange equations. This section presents that connection in detail.

We start by reviewing Geometric Dynamical Systems (GDS) and express them in a (tensor notion) form that makes it easier to see the relationship to Lagrangian and Finsler systems. Let $\mathbf{G}(\mathbf{q}, \dot{\mathbf{q}})$ be a velocity dependent metric tensor and define an energy function as $\mathcal{K}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2}\dot{\mathbf{q}}^T \mathbf{G}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$. Following the notation of [8], we denote the columns of \mathbf{G} by \mathbf{g}_i so that $\mathbf{G} = [\mathbf{g}_1, \dots, \mathbf{g}_d]$. An unforced GDS is defined as

$$\mathbf{M}(\mathbf{q}, \dot{\mathbf{q}})\ddot{\mathbf{q}} + \boldsymbol{\xi}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{0}.$$

where $\mathbf{M} = \mathbf{G} + \boldsymbol{\Xi}$ has additional mass defined by $\boldsymbol{\Xi} = \frac{1}{2} \sum_i \dot{q}^i \partial_{\dot{\mathbf{q}}} \mathbf{g}_i$ (here \dot{q}^i denotes the i th component of $\dot{\mathbf{q}}$) and $\boldsymbol{\xi} = \mathbf{G}\dot{\mathbf{q}} - \frac{1}{2}\partial_{\mathbf{q}}(\dot{\mathbf{q}}^T \mathbf{G}\dot{\mathbf{q}})$ using $\mathbf{G} = [\mathbf{h}_1, \dots, \mathbf{h}_d]$ with $\mathbf{h}_i = (\partial_{\mathbf{q}} \mathbf{g}_i)\dot{\mathbf{q}}$. When \mathbf{G} is independent of $\dot{\mathbf{q}}$, these equations of motion reduce to the classical equations since $\mathbf{M} \rightarrow \mathbf{G}$ and $\mathbf{G} \rightarrow \dot{\mathbf{G}}$.

GDS systems conserve total energy $\mathcal{E} = \mathcal{K} + \psi(\mathbf{q})$ when forced by a potential function $\psi(\mathbf{q})$ and are shown to optimize ψ when energy is bled off using a damping term $-\mathbf{B}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ for positive definite \mathbf{B} similar to the setting outlined in Theorem IV.4.

Using tensor notation with the Einstein summation convention, this system can be expressed $M_{kl}\ddot{q}^l + \xi_{kij}\dot{q}^i\dot{q}^j = 0$ where $M_{kl} = G_{kl} + \Xi_{kl}$ with

$$\Xi_{kl} = \frac{1}{2} \frac{\partial G_{ik}}{\partial \dot{q}^l} \dot{q}^i \quad (22)$$

$$\xi_{kij} = \frac{1}{2} \left(\frac{\partial G_{kj}}{\partial \dot{q}^i} + \frac{\partial G_{ik}}{\partial \dot{q}^j} - \frac{\partial G_{ij}}{\partial \dot{q}^k} \right), \quad (23)$$

where ξ_{kij} are the standard Christoffel symbols of the first kind [25]. The only augmentation to the classical system is the term Ξ which vanishes when \mathbf{G} is independent of velocity.

A GDS by itself is not covariant to change of coordinates [12], but can be made covariant with the use of transform trees (see *structured GDS* in [8] and the discussion above in Appendix A-B). However, the next theorem shows that Lagrangian systems, and specifically Finsler systems derived from energies of the form $\mathcal{L}_e(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{G}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}}$, make a qualitatively similar, but inherently covariant, alternative to GDS systems.

Theorem B.1. *Let $\mathcal{L} = \frac{1}{2} G_{ij} \dot{q}^i \dot{q}^j$ be a Finsler energy with $G_{ij}(q, \dot{q})$ a function of both position and velocity. Then the Finsler system's equations of motion (defined by the Euler-Lagrange equation) can be expressed*

$$M_{kl} \ddot{q}^l + \Upsilon_{kij} \dot{q}^i \dot{q}^j = 0, \quad (24)$$

where $M_{kl} = G_{kl} + \Xi_{kl}$ with velocity curvature mass

$$\Xi_{kl} = \frac{\partial G_{lj}}{\partial \dot{q}^k} \dot{q}^j + \frac{\partial G_{kj}}{\partial \dot{q}^l} \dot{q}^j + \frac{1}{2} \frac{\partial^2 G_{ij}}{\partial \dot{q}^l \partial \dot{q}^k} \dot{q}^i \dot{q}^j, \quad (25)$$

and velocity-augmented Christoffel symbols (of the first kind) given by

$$\Upsilon_{kij} = \frac{1}{2} \left(\frac{\partial G_{kj}}{\partial q^i} + \frac{\partial G_{ik}}{\partial q^j} - \frac{\partial G_{ij}}{\partial q^k} + \frac{\partial^2 G_{ij}}{\partial q^l \partial \dot{q}^k} \dot{q}^l \right). \quad (26)$$

Proof. This proof amounts primarily to calculation and gathering together the terms appropriately to construct M_{kl} and Υ_{kij} .

The Lagrangian is $\mathcal{L} = \frac{1}{2} G_{ij} \dot{q}^i \dot{q}^j$ and the Euler-Lagrange equation defines $\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}^k} - \frac{\partial \mathcal{L}}{\partial q^k} = 0$. That gives

$$\underbrace{\frac{d}{dt} \left[\frac{\partial}{\partial \dot{q}^k} \left(\frac{1}{2} G_{ij} \dot{q}^i \dot{q}^j \right) \right]}_{\text{term 1}} - \frac{1}{2} \frac{\partial G_{ij}}{\partial q^k} \dot{q}^j \dot{q}^i = 0.$$

Term 1 becomes

$$\begin{aligned} \frac{d}{dt} \left[\frac{\partial}{\partial \dot{q}^k} \left(\frac{1}{2} G_{ij} \dot{q}^i \dot{q}^j \right) \right] &= \frac{d}{dt} \left[\frac{1}{2} \frac{\partial G_{ij}}{\partial \dot{q}^k} \dot{q}^i \dot{q}^j + G_{kj} \dot{q}^j \right] \\ &= \frac{1}{2} \left(\frac{\partial^2 G_{ij}}{\partial q^l \partial \dot{q}^k} \dot{q}^l + \frac{\partial^2 G_{ij}}{\partial \dot{q}^l \partial \dot{q}^k} \dot{q}^l \right) \dot{q}^i \dot{q}^j + \frac{1}{2} \frac{\partial G_{ij}}{\partial \dot{q}^k} \dot{q}^i \dot{q}^j \\ &\quad + \frac{1}{2} \frac{\partial G_{ij}}{\partial \dot{q}^k} \dot{q}^i \dot{q}^j + \left(\frac{\partial G_{kj}}{\partial q^l} \dot{q}^l + \frac{\partial G_{kj}}{\partial \dot{q}^l} \dot{q}^l \right) \dot{q}^j + G_{kj} \ddot{q}^j. \end{aligned}$$

Putting this together with the other terms and grouping them as needed, we get

$$\begin{aligned} \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}^k} - \frac{\partial \mathcal{L}}{\partial q^k} &= \left(G_{kl} + \frac{1}{2} \left[\frac{\partial G_{lj}}{\partial \dot{q}^k} \dot{q}^j + \frac{\partial G_{il}}{\partial \dot{q}^k} \dot{q}^i \right] + \frac{\partial G_{kj}}{\partial \dot{q}^l} \dot{q}^j \right. \\ &\quad \left. + \frac{1}{2} \frac{\partial^2 G_{ij}}{\partial \dot{q}^l \partial \dot{q}^k} \dot{q}^i \dot{q}^j \right) \ddot{q}^l \\ &\quad + \left(\frac{1}{2} \frac{\partial^2 G_{ij}}{\partial q^l \partial \dot{q}^k} \dot{q}^l + \frac{\partial G_{kj}}{\partial q^i} - \frac{1}{2} \frac{\partial G_{ij}}{\partial q^k} \right) \dot{q}^i \dot{q}^j. \end{aligned}$$

Since G_{ij} is symmetric, we have $\frac{\partial G_{lj}}{\partial \dot{q}^k} \dot{q}^j = \frac{\partial G_{il}}{\partial \dot{q}^k} \dot{q}^i$, and

$$\begin{aligned} \frac{\partial G_{kj}}{\partial \dot{q}^i} \dot{q}^i \dot{q}^j &= \frac{1}{2} \frac{\partial G_{kj}}{\partial q^i} \dot{q}^i \dot{q}^j + \frac{1}{2} \frac{\partial G_{jk}}{\partial q^i} \dot{q}^i \dot{q}^j \\ &= \frac{1}{2} \left(\frac{\partial G_{kj}}{\partial q^i} + \frac{\partial G_{ik}}{\partial q^j} \right) \dot{q}^i \dot{q}^j, \end{aligned}$$

so we can rewrite the above expression in its final form

$$\begin{aligned} \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}^k} - \frac{\partial \mathcal{L}}{\partial q^k} &= \left(G_{kl} + \frac{\partial G_{lj}}{\partial \dot{q}^k} \dot{q}^j + \frac{\partial G_{kj}}{\partial \dot{q}^l} \dot{q}^j + \frac{1}{2} \frac{\partial^2 G_{ij}}{\partial \dot{q}^l \partial \dot{q}^k} \dot{q}^i \dot{q}^j \right) \ddot{q}^l \\ &\quad + \frac{1}{2} \left(\frac{\partial^2 G_{ij}}{\partial q^l \partial \dot{q}^k} \dot{q}^l + \frac{\partial G_{kj}}{\partial q^i} + \frac{\partial G_{ik}}{\partial q^j} - \frac{\partial G_{ij}}{\partial q^k} \right) \dot{q}^i \dot{q}^j \\ &= \left(G_{kl} + \Xi_{kl} \right) \ddot{q}^l + \Upsilon_{kij} \dot{q}^i \dot{q}^j, \end{aligned}$$

with Ξ_{kl} and Υ_{kij} as given in the above theorem. \square

Comparing the terms of Theorem B.1 to Equations 22 and 23 we see that Finsler systems of this form contains additional terms deriving from the velocity dependence in both the added mass and the augmented Christoffel symbols. Since these equations are derived using the Euler-Lagrange equation, they're known to be covariant to reparameterization; the added terms are critical for that covariance property to hold. Moreover, the class of all Finsler systems is a superset of these GDS-like Finsler systems, so Finsler systems broadly have many advantages over GDSs.

Both GDSs and Finsler systems, however, generalize metrics to be velocity dependent but remain subject to the other limitations outlined in Subsection III-B. Namely, their metrics and associated policies remain coupled. In practice, in the earlier results of [8], high gains on forcing potentials and damping terms were used in order to leverage the systems for metric design while also reject the resulting irrelevant geometric terms that resulted. Geometric fabrics added a new dimension of flexibility by introducing bending terms to circumvent these issues.

APPENDIX C ADDITIONAL LEMMAS

This Lemma gives the time derivative of a Lagrangian's energy (the Hamiltonian), which is a common calculation needed for multiple proofs below.

Lemma C.1. *Let $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})$ be any Lagrangian with Hamiltonian (energy) $\mathcal{H}_{\mathcal{L}} = \partial_{\dot{\mathbf{q}}} \mathcal{L}^T \dot{\mathbf{q}} - \mathcal{L}$. The energy time derivative is*

$$\dot{\mathcal{H}}_{\mathcal{L}} = \dot{\mathbf{q}}^T (\mathbf{M} \ddot{\mathbf{q}} + \boldsymbol{\xi}), \quad (27)$$

where \mathbf{M} and $\boldsymbol{\xi}$ come from the Lagrangian's equations of motion $\mathbf{M}(\mathbf{q}, \dot{\mathbf{q}}) \ddot{\mathbf{q}} + \boldsymbol{\xi}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{0}$.

Proof. The calculation is a straightforward time derivative of

the Hamiltonian:

$$\begin{aligned}\dot{\mathcal{H}}_{\mathcal{L}} &= \frac{d}{dt} \left[\partial_{\dot{\mathbf{q}}} \mathcal{L}^T \dot{\mathbf{q}} - \mathcal{L} \right] \\ &= (\partial_{\dot{\mathbf{q}}\dot{\mathbf{q}}}^2 \mathcal{L} \ddot{\mathbf{q}} + \partial_{\dot{\mathbf{q}}\mathbf{q}} \mathcal{L} \dot{\mathbf{q}})^T \dot{\mathbf{q}} + \partial_{\dot{\mathbf{q}}} \mathcal{L}^T \ddot{\mathbf{q}} \\ &\quad - (\partial_{\dot{\mathbf{q}}} \mathcal{L}^T \ddot{\mathbf{q}} + \partial_{\dot{\mathbf{q}}} \mathcal{L}^T \dot{\mathbf{q}}) \\ &= \dot{\mathbf{q}}^T \left(\partial_{\dot{\mathbf{q}}\dot{\mathbf{q}}}^2 \mathcal{L} \ddot{\mathbf{q}} + \partial_{\dot{\mathbf{q}}\mathbf{q}} \mathcal{L} \dot{\mathbf{q}} - \partial_{\dot{\mathbf{q}}} \mathcal{L} \right) \\ &= \dot{\mathbf{q}}^T (\mathbf{M} \ddot{\mathbf{q}} + \boldsymbol{\xi}).\end{aligned}$$

where $\mathbf{M} = \partial_{\dot{\mathbf{q}}\dot{\mathbf{q}}}^2 \mathcal{L}$ and $\boldsymbol{\xi} = \partial_{\dot{\mathbf{q}}\mathbf{q}} \mathcal{L} \dot{\mathbf{q}} - \partial_{\dot{\mathbf{q}}} \mathcal{L}$. \square

The following lemma collects some results around common matrices and operators that arise when analyzing energy conservation.

Lemma C.2. *Let \mathbf{M} be a symmetric positive definite matrix and define $\mathbf{p} = \mathbf{M}\dot{\mathbf{x}}$. Then*

$$\mathbf{R}_{\mathbf{p}} = \mathbf{M}^{-1} - \frac{\dot{\mathbf{x}} \dot{\mathbf{x}}^T}{\dot{\mathbf{x}}^T \mathbf{M} \dot{\mathbf{x}}} \quad (28)$$

has null space spanned by \mathbf{p} and

$$\mathbf{R}_{\dot{\mathbf{x}}} = \mathbf{M} - \frac{\mathbf{p} \mathbf{p}^T}{\mathbf{p}^T \mathbf{M}^{-1} \mathbf{p}} \quad (29)$$

has null space spanned by $\dot{\mathbf{x}}$. These matrices are related by $\mathbf{R}_{\dot{\mathbf{x}}} = \mathbf{M} \mathbf{R}_{\mathbf{p}} \mathbf{M}$ and the matrix $\mathbf{M}^{-1} \mathbf{R}_{\dot{\mathbf{x}}} = \mathbf{R}_{\mathbf{p}} \mathbf{M} = \mathbf{P}_e$ is a projection operator of the form

$$\mathbf{P}_e = \mathbf{M}^{\frac{1}{2}} \left[\mathbf{I} - \hat{\mathbf{v}} \hat{\mathbf{v}}^T \right] \mathbf{M}^{-\frac{1}{2}} \quad (30)$$

where $\mathbf{v} = \mathbf{M}^{\frac{1}{2}} \dot{\mathbf{x}}$ and $\hat{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$ is the normalized vector. Moreover, $\dot{\mathbf{x}}^T \mathbf{P}_e = \mathbf{0}$.

Proof. Right multiplication of $\mathbf{R}_{\mathbf{p}}$ by \mathbf{p} gives:

$$\begin{aligned}\mathbf{R}_{\mathbf{p}} \mathbf{p} &= \left(\mathbf{M}^{-1} - \frac{\dot{\mathbf{x}} \dot{\mathbf{x}}^T}{\dot{\mathbf{x}}^T \mathbf{M} \dot{\mathbf{x}}} \right) \mathbf{M} \dot{\mathbf{x}} \\ &= \dot{\mathbf{x}} - \dot{\mathbf{x}} \left(\frac{\dot{\mathbf{x}}^T \mathbf{M} \dot{\mathbf{x}}}{\dot{\mathbf{x}}^T \mathbf{M} \dot{\mathbf{x}}} \right) = \mathbf{0},\end{aligned}$$

so \mathbf{p} lies in the null space. Moreover, the null space is no larger since each matrix is formed by subtracting off a rank 1 term from a full rank matrix.

The relation between $\mathbf{R}_{\mathbf{p}}$ and $\mathbf{R}_{\dot{\mathbf{x}}}$ can be shown algebraically

$$\begin{aligned}\mathbf{M} \mathbf{R}_{\mathbf{p}} \mathbf{M} &= \mathbf{M} \left(\mathbf{M}^{-1} - \frac{\dot{\mathbf{x}} \dot{\mathbf{x}}^T}{\dot{\mathbf{x}}^T \mathbf{M} \dot{\mathbf{x}}} \right) \mathbf{M} \\ &= \mathbf{M} - \frac{\mathbf{M} \dot{\mathbf{x}} \dot{\mathbf{x}}^T \mathbf{M}}{\dot{\mathbf{x}}^T \mathbf{M} \dot{\mathbf{x}}} \\ &= \mathbf{M} - \frac{\mathbf{p} \mathbf{p}^T}{\mathbf{p}^T \mathbf{M}^{-1} \mathbf{p}} = \mathbf{R}_{\dot{\mathbf{x}}}.\end{aligned}$$

Since \mathbf{M} has full rank, $\mathbf{R}_{\dot{\mathbf{x}}}$ has the same rank as $\mathbf{R}_{\mathbf{p}}$ and its null space must be spanned by $\dot{\mathbf{x}}$ since $\mathbf{R}_{\dot{\mathbf{x}}} \dot{\mathbf{x}} = \mathbf{M} \mathbf{R}_{\mathbf{p}} \mathbf{M} \dot{\mathbf{x}} = \mathbf{M} \mathbf{R}_{\mathbf{p}} \mathbf{p} = \mathbf{0}$.

With a slight algebraic manipulation, we get

$$\begin{aligned}\mathbf{M} \mathbf{R}_{\mathbf{p}} &= \mathbf{M} \left(\mathbf{M}^{-1} - \frac{\dot{\mathbf{x}} \dot{\mathbf{x}}^T}{\dot{\mathbf{x}}^T \mathbf{M} \dot{\mathbf{x}}} \right) \\ &= \mathbf{M}^{\frac{1}{2}} \left(\mathbf{I} - \frac{\mathbf{M}^{\frac{1}{2}} \dot{\mathbf{x}} \dot{\mathbf{x}}^T \mathbf{M}^{\frac{1}{2}}}{\dot{\mathbf{x}}^T \mathbf{M}^{\frac{1}{2}} \mathbf{M}^{\frac{1}{2}} \dot{\mathbf{x}}} \right) \mathbf{M}^{-\frac{1}{2}} \\ &= \mathbf{M}^{\frac{1}{2}} \left(\mathbf{I} - \frac{\mathbf{v} \mathbf{v}^T}{\mathbf{v}^T \mathbf{v}} \right) \mathbf{M}^{-\frac{1}{2}} \\ &= \mathbf{P}_e\end{aligned}$$

since $\frac{\mathbf{v} \mathbf{v}^T}{\mathbf{v}^T \mathbf{v}} = \hat{\mathbf{v}} \hat{\mathbf{v}}^T$. Moreover,

$$\begin{aligned}\mathbf{P}_e \mathbf{P}_e &= \mathbf{M}^{\frac{1}{2}} (\mathbf{I} - \hat{\mathbf{v}} \hat{\mathbf{v}}^T) \mathbf{M}^{-\frac{1}{2}} \\ &\quad \cdot \mathbf{M}^{\frac{1}{2}} (\mathbf{I} - \hat{\mathbf{v}} \hat{\mathbf{v}}^T) \mathbf{M}^{-\frac{1}{2}} \\ &= \mathbf{M}^{\frac{1}{2}} (\mathbf{I} - \hat{\mathbf{v}} \hat{\mathbf{v}}^T) (\mathbf{I} - \hat{\mathbf{v}} \hat{\mathbf{v}}^T) \mathbf{M}^{-\frac{1}{2}} \\ &= \mathbf{M}^{\frac{1}{2}} (\mathbf{I} - \hat{\mathbf{v}} \hat{\mathbf{v}}^T) \mathbf{M}^{-\frac{1}{2}} = \mathbf{P}_e,\end{aligned}$$

since $\mathbf{P}_{\perp} = \mathbf{I} - \hat{\mathbf{v}} \hat{\mathbf{v}}^T$ is an orthogonal projection operator. Therefore, $\mathbf{P}_e^2 = \mathbf{P}_e$ showing that it is a projection operator as well.

Finally, we have

$$\begin{aligned}\dot{\mathbf{x}}^T \mathbf{P}_e &= \dot{\mathbf{x}}^T \mathbf{M} \mathbf{R}_{\mathbf{p}} = \dot{\mathbf{x}}^T \mathbf{M} \left(\mathbf{M}^{-1} - \frac{\dot{\mathbf{x}} \dot{\mathbf{x}}^T}{\dot{\mathbf{x}}^T \mathbf{M} \dot{\mathbf{x}}} \right) \\ &= \left[\dot{\mathbf{x}}^T - \left(\frac{\dot{\mathbf{x}}^T \mathbf{M} \dot{\mathbf{x}}}{\dot{\mathbf{x}}^T \mathbf{M} \dot{\mathbf{x}}} \right) \dot{\mathbf{x}}^T \right] = \mathbf{0}.\end{aligned}$$

\square

APPENDIX D

PROOFS OF LEMMAS AND THEOREMS IN THE PAPER

A. Lemma IV.1 on energy conservation and dissipation

Proof of Lemma IV.1 on energy conservation and dissipation of Finsler systems with zero-work modification terms. See Subsection IV-A for the full statement.

Proof. This energy is conserved if its time derivative is zero. Substituting $\ddot{\mathbf{q}} = -\mathbf{M}^{-1}(\boldsymbol{\xi} + \mathbf{f}_f + \partial\psi + \mathbf{B}\dot{\mathbf{q}})$ into Equation 27 of Appendix Lemma C.1 using Lagrangian $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = \mathcal{L}_e - \psi$ gives

$$\begin{aligned}\dot{\mathcal{H}}_{\mathcal{L}_e} &= \dot{\mathbf{q}}^T \left(\mathbf{M} (-\mathbf{M}^{-1}(\boldsymbol{\xi} + \mathbf{f}_f + \partial\psi + \mathbf{B}\dot{\mathbf{q}})) + \boldsymbol{\xi} + \partial\psi \right) \\ &= \dot{\mathbf{q}}^T \left(-(\boldsymbol{\xi} + \partial\psi) + (\boldsymbol{\xi} + \partial\psi) - (\mathbf{f}_f + \mathbf{B}\dot{\mathbf{q}}) \right) \\ &= -\dot{\mathbf{q}}^T \mathbf{f}_f - \dot{\mathbf{q}}^T \mathbf{B}\dot{\mathbf{q}}.\end{aligned}$$

Therefore, energy reduces at a rate $-\dot{\mathbf{q}}^T \mathbf{B}\dot{\mathbf{q}}$ (and is conserved when $\mathbf{B} = \mathbf{0}$) if and only if \mathbf{f}_f is zero-work with $\dot{\mathbf{q}}^T \mathbf{f}_f = 0$ everywhere.

Finally, when \mathbf{f}_f is geometric $\mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\xi} + \mathbf{f}_f = \mathbf{0}$ is geometric since \mathbf{M} is HD0 and $\boldsymbol{\xi}$ is HD2. Moreover, since \mathcal{L}_e is lower bounded by 0, $\mathcal{H} = \mathcal{L}_e + \psi$ is lower bounded when ψ is. \square

B. Lemma IV.3 on constrained fabrics

Proof of Lemma IV.3 in Subsection IV-B showing that constrained fabrics are also fabrics.

Proof. The formula derives from the standard algebra of constrained systems used in classical mechanics [1], which we derive here for completeness. The Principle of Virtual Work tells us the constraint force takes the form $\mathbf{f}_c = \mathbf{J}^T \boldsymbol{\lambda}$ for unknown Lagrange multipliers $\boldsymbol{\lambda}$. (Since \mathbf{J} 's rows are orthogonal to the constraint's tangent space, \mathbf{f}_c defined this way is always zero work for any valid velocity consistent with the constraints.) The constraint $\mathbf{C}(\mathbf{q}) = \mathbf{0}$ induces two additional constraints by its time derivatives $\dot{\mathbf{C}} = \mathbf{J}\dot{\mathbf{q}} = \mathbf{0}$ and $\ddot{\mathbf{C}} = \mathbf{J}\ddot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{q}} = \mathbf{0}$. The constraint and its first derivative are satisfied by assumption, so it's the third that enters into the system:

$$\begin{aligned} \mathbf{M}\ddot{\mathbf{q}} + \mathbf{f} + \mathbf{J}^T \boldsymbol{\lambda} &= \mathbf{0} \\ \mathbf{J}\ddot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{q}} &= \mathbf{0}, \end{aligned}$$

where we've captured all force terms in $\mathbf{f} = \boldsymbol{\xi} + \mathbf{f}_f + \partial\psi + \mathbf{B}\dot{\mathbf{q}}$. Multiplying the first equation by $\mathbf{J}\mathbf{M}^{-1}$ and subtracting the second, we can solve for $\boldsymbol{\lambda}$:

$$\boldsymbol{\lambda} = (\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}[-\mathbf{J}\mathbf{M}^{-1}\mathbf{f} + \dot{\mathbf{J}}\dot{\mathbf{q}}].$$

These Lagrange multipliers define the constraint forces; plugging them back into the first equation and collecting terms gives:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{P}_{//}\mathbf{f} + \mathbf{P}_{\perp}\mathbf{M}\dot{\mathbf{J}}\dot{\mathbf{q}} = \mathbf{0}.$$

This result gives Equation 5 once \mathbf{f} is split apart. Additionally, adding and subtracting $\boldsymbol{\xi}$ gives

$$\begin{aligned} \mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\xi} - \boldsymbol{\xi} + \mathbf{P}_{//}(\boldsymbol{\xi} + \mathbf{f}_f) + \mathbf{P}_{\perp}\mathbf{M}\dot{\mathbf{J}}\dot{\mathbf{q}} \\ = \mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\xi} + \mathbf{P}_{\perp}(\mathbf{M}\dot{\mathbf{J}}\dot{\mathbf{q}} - \boldsymbol{\xi}) + \mathbf{P}_{//}\mathbf{f}_f, \end{aligned}$$

which derives the expression for $\tilde{\mathbf{f}}_f = \mathbf{P}_{\perp}(\mathbf{M}\dot{\mathbf{J}}\dot{\mathbf{q}} - \boldsymbol{\xi}) + \mathbf{P}_{//}\mathbf{f}_f$

Observing that \mathbf{M} is HD0 (see discussion in Section III-A), we see that both $\mathbf{P}_{//}$ and \mathbf{P}_{\perp} are HD0 (i.e. they do not depend on $\|\dot{\mathbf{q}}\|$). Since \mathbf{f}_f is HD2 by definition, its projection is as well. Moreover, $\boldsymbol{\xi}$ is known to be HD2 and using tensor notation, we see $\dot{\mathbf{J}}\dot{\mathbf{q}} = \frac{\partial^2 \mathbf{C}}{\partial q^i \partial q^j} \dot{q}^i \dot{q}^j$ is HD2, which means means both $\mathbf{M}\dot{\mathbf{J}}\dot{\mathbf{q}} - \boldsymbol{\xi}$ and its projection are HD2 as well. Together these results show that $\tilde{\mathbf{f}}_f$ is HD2, and since we know it derives from the original bending term by adding a zero-work constraint force, \mathbf{f}_f must be zero work as well and is hence a bending term. This proves that $(\mathcal{L}_e, \tilde{\mathbf{f}}_f)$ is a geometric fabric. \square

C. Theorem IV.4 on Geometric Fabric Stability

Proof of theorem in Subsection IV-C characterizing the stability of geometric fabrics and convergence to KKT solutions of the constrained potential.

Proof. By Lemma IV.1, the forced system will reduce total energy $\mathcal{H}(\mathbf{q}, \dot{\mathbf{q}}) = \mathcal{L}_e(\mathbf{q}, \dot{\mathbf{q}}) + \psi(\mathbf{q})$, at a rate $\dot{\mathcal{H}} \leq -\dot{\mathbf{q}}^T \mathbf{B}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}}$ (the inequality is due to the impact model). Since $\mathcal{L}_e \geq 0$ and $\psi(\mathbf{q})$ is lower bounded, \mathcal{H} is lower bounded and it must be that $\dot{\mathcal{H}} \rightarrow 0$. Therefore, $\dot{\mathbf{q}}^T \mathbf{B}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} \rightarrow 0$ which implies $\dot{\mathbf{q}} \rightarrow \mathbf{0}$ since \mathbf{B} is positive definite. Since velocity approaches zero we must also have $\ddot{\mathbf{q}} \rightarrow \mathbf{0}$. At convergence, a subset of inequality constraints will be active resulting in an equality constraint for

those dimensions which we denote $\tilde{\mathbf{g}} = \mathbf{0}$. Combining those active inequality constraints with the equality constraints \mathbf{c} into $\mathbf{C} = [\mathbf{c}; \tilde{\mathbf{g}}]$ with Jacobian $\mathbf{J} = \partial_{\mathbf{q}} \mathbf{C}$, the constrained the equations of motion at convergence can be expressed

$$\mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\xi} + \mathbf{f}_f + \mathbf{J}^T \boldsymbol{\lambda} = -\partial_{\mathbf{q}} \psi - \mathbf{B}\dot{\mathbf{q}}. \quad (31)$$

Since $\ddot{\mathbf{q}} \rightarrow \mathbf{0}$ we have $\mathbf{M}\ddot{\mathbf{q}} \rightarrow \mathbf{0}$ (\mathbf{M} is HD0) and since $\dot{\mathbf{q}} \rightarrow \mathbf{0}$ we have $\boldsymbol{\xi} + \mathbf{f}_f \rightarrow \mathbf{0}$ (both $\boldsymbol{\xi}$ and \mathbf{f}_f are HD2). That leaves just $\mathbf{J}^T \boldsymbol{\lambda}$ on the left hand side. On the right hand side we have $\mathbf{B}\dot{\mathbf{q}} \rightarrow \mathbf{0}$ since \mathbf{B} is bounded, so the system approaches

$$\mathbf{J}^T \boldsymbol{\lambda} = -\partial_{\mathbf{q}} \psi. \quad (32)$$

Therefore, it approaches a KKT solution since the system evolution constraints are everywhere satisfied and contact models require $\boldsymbol{\lambda}$ satisfy the required complementarity conditions. \square

D. Theorem IV.5 on Energized Fabrics

Proof of Theorem IV.5 in Subsection IV-D showing how to energize arbitrary HD2 geometries and proving the resulting system is a geometric fabric.

Proof. We derive these results using notation $\mathbf{h} = -\pi(\mathbf{q}, \dot{\mathbf{q}})$ so that the HD2 geometry can be expressed $\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{0}$. Using $\mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\xi} = \mathbf{0}$ to denote the unforced equations of motion, Equation 27 of Appendix Lemma C.1 gives the time derivative of the energy $\dot{\mathcal{H}}_e = \dot{\mathbf{q}}^T (\mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\xi})$. Substituting a system of the form $\ddot{\mathbf{q}} = -\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) - \alpha_{\mathcal{H}_e} \dot{\mathbf{q}}$, setting it to zero, and solving for $\alpha_{\mathcal{H}_e}$ gives

$$\begin{aligned} \dot{\mathcal{H}}_e &= \dot{\mathbf{q}}^T [\mathbf{M}(-\mathbf{h} - \alpha_{\mathcal{H}_e} \dot{\mathbf{q}}) + \boldsymbol{\xi}] = \mathbf{0} \\ &\Rightarrow -\dot{\mathbf{q}}^T \mathbf{M}\mathbf{h} - \alpha_{\mathcal{H}_e} \dot{\mathbf{q}}^T \mathbf{M}\dot{\mathbf{q}} + \dot{\mathbf{q}}^T \boldsymbol{\xi} = \mathbf{0} \\ &\Rightarrow \alpha_{\mathcal{H}_e} = -\frac{\dot{\mathbf{q}}^T \mathbf{M}\mathbf{h} - \dot{\mathbf{q}}^T \boldsymbol{\xi}}{\dot{\mathbf{q}}^T \mathbf{M}\dot{\mathbf{q}}} \\ &= -(\dot{\mathbf{q}}^T \mathbf{M}\dot{\mathbf{q}})^{-1} \dot{\mathbf{q}}^T [\mathbf{M}\mathbf{h} - \boldsymbol{\xi}]. \end{aligned}$$

This result gives the formula for Equation 7. Substituting this solution for $\alpha_{\mathcal{H}_e}$ back in gives

$$\begin{aligned} \ddot{\mathbf{q}} &= -\mathbf{h} + \left(\frac{\dot{\mathbf{q}}^T}{\dot{\mathbf{q}}^T \mathbf{M}\dot{\mathbf{q}}} [\mathbf{M}\mathbf{h} - \boldsymbol{\xi}] \right) \dot{\mathbf{q}} \\ &= -\mathbf{h} + \left[\frac{\dot{\mathbf{q}} \dot{\mathbf{q}}^T}{\dot{\mathbf{q}}^T \mathbf{M}\dot{\mathbf{q}}} \right] (\mathbf{M}\mathbf{h} - \boldsymbol{\xi}). \end{aligned}$$

Algebraically, it helps to introduce $\mathbf{h} = \mathbf{M}^{-1}[\mathbf{f}_f + \boldsymbol{\xi}]$ to first express the result as a difference away from $\boldsymbol{\xi}$; in the end we will convert back to \mathbf{h} . Doing so and moving all the terms to

the left hand side of the equation gives

$$\begin{aligned}
& \ddot{\mathbf{q}} + \mathbf{h} - \left[\frac{\dot{\mathbf{q}}\dot{\mathbf{q}}^T}{\dot{\mathbf{q}}^T\mathbf{M}\dot{\mathbf{q}}} \right] (\mathbf{M}\mathbf{h} - \boldsymbol{\xi}) = \mathbf{0} \\
\Rightarrow & \ddot{\mathbf{q}} + \mathbf{M}^{-1}[\mathbf{f}_f + \boldsymbol{\xi}] \\
& - \left[\frac{\dot{\mathbf{q}}\dot{\mathbf{q}}^T}{\dot{\mathbf{q}}^T\mathbf{M}\dot{\mathbf{q}}} \right] (\mathbf{M}\mathbf{M}^{-1}[\mathbf{f}_f + \boldsymbol{\xi}] - \boldsymbol{\xi}) = \mathbf{0} \\
\Rightarrow & \mathbf{M}\ddot{\mathbf{q}} + \mathbf{f}_f + \boldsymbol{\xi} \\
& - \mathbf{M} \left[\frac{\dot{\mathbf{q}}\dot{\mathbf{q}}^T}{\dot{\mathbf{q}}^T\mathbf{M}\dot{\mathbf{q}}} \right] (\mathbf{f}_f + (\boldsymbol{\xi} - \boldsymbol{\xi})) = \mathbf{0} \\
\Rightarrow & \mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\xi} + \mathbf{M} \left[\mathbf{M}^{-1} - \frac{\dot{\mathbf{q}}\dot{\mathbf{q}}^T}{\dot{\mathbf{q}}^T\mathbf{M}\dot{\mathbf{q}}} \right] \mathbf{f}_f = \mathbf{0} \\
\Rightarrow & \mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\xi} + \mathbf{M}\mathbf{R}_p(\mathbf{M}\mathbf{h} - \boldsymbol{\xi}) = \mathbf{0},
\end{aligned}$$

where $\mathbf{R}_p = \mathbf{M}^{-1} - \frac{\dot{\mathbf{q}}\dot{\mathbf{q}}^T}{\dot{\mathbf{q}}^T\mathbf{M}\dot{\mathbf{q}}}$ matching Equation 28 of Lemma C.2, and we substitute $\mathbf{f}_f = \mathbf{M}\mathbf{h} - \boldsymbol{\xi}$ back in. By Lemma C.2 $\mathbf{M}\mathbf{R}_p = \mathbf{P}_e$, so we get Equation 8. Moreover, again by Lemma C.2, $\dot{\mathbf{q}}^T\mathbf{P}_e = \mathbf{0}$, so $\dot{\mathbf{q}}^T\mathbf{f}_f = -\dot{\mathbf{q}}^T\mathbf{P}_e[\mathbf{M}\pi + \boldsymbol{\xi}] = \mathbf{0}$ showing that the new term is a zero work modification. Finally, both \mathbf{h} and $\boldsymbol{\xi}$ are HD2 by definition, and since \mathbf{P}_e and \mathbf{M} are HD0, the modifying term is HD2 making it a bending term. Therefore, the energized system is a bent Finsler system and hence a geometric fabric. \square

E. Theorem IV.6 on Fabrics in Generalized Coordinates

Proof of Theorem IV.6 showing how to express in generalized coordinates energized systems that are subsequently constrained. See Subsection IV-E.

Proof. We will show the equivalence by calculation. Let $\mathbf{h}(\mathbf{x}, \dot{\mathbf{x}}) = -\pi(\mathbf{x}, \dot{\mathbf{x}})$ so that the geometry's equation becomes $\ddot{\mathbf{x}} + \mathbf{h}(\mathbf{x}, \dot{\mathbf{x}}) = \mathbf{0}$. Multiplying through by \mathbf{M} gives its metric-weighted (force) form $\mathbf{M}\ddot{\mathbf{x}} + \mathbf{f} = \mathbf{0}$ where $\mathbf{f} = \mathbf{M}\mathbf{h}$. Using Equation 7 of Theorem IV.5, this system's energization with respect to energy \mathcal{L}_e with equations of motion $\mathbf{M}\ddot{\mathbf{x}} + \boldsymbol{\xi} = \mathbf{0}$ can be expressed $\mathbf{M}\ddot{\mathbf{x}} + \boldsymbol{\xi}^h = \mathbf{0}$ where

$$\boldsymbol{\xi}^h = \boldsymbol{\xi} + \mathbf{M} \left[\mathbf{M}^{-1} - \frac{\dot{\mathbf{x}}\dot{\mathbf{x}}^T}{\dot{\mathbf{x}}^T\mathbf{M}\dot{\mathbf{x}}} \right] (\mathbf{f} - \boldsymbol{\xi}). \quad (33)$$

Let $\mathbf{J} = \partial_{\mathbf{q}}\phi$. The pullback of the energized geometry generator is

$$\begin{aligned}
& \mathbf{J}^T\mathbf{M}(\mathbf{J}\ddot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{q}}) + \mathbf{J}^T\boldsymbol{\xi}^h = \mathbf{0} \\
\Rightarrow & (\mathbf{J}^T\mathbf{M}\mathbf{J})\ddot{\mathbf{q}} + \mathbf{J}^T(\boldsymbol{\xi}^h + \mathbf{M}\dot{\mathbf{J}}\dot{\mathbf{q}}) = \mathbf{0} \\
\Rightarrow & (\mathbf{J}^T\mathbf{M}\mathbf{J})\ddot{\mathbf{q}} + \mathbf{J}^T\boldsymbol{\xi} \\
& + \mathbf{J}^T\mathbf{M} \left[\mathbf{M}^{-1} - \frac{\dot{\mathbf{x}}\dot{\mathbf{x}}^T}{\dot{\mathbf{x}}^T\mathbf{M}\dot{\mathbf{x}}} \right] (\mathbf{f} - \boldsymbol{\xi}) + \mathbf{J}^T\mathbf{M}\dot{\mathbf{J}}\dot{\mathbf{q}} = \mathbf{0} \\
\Rightarrow & \widetilde{\mathbf{M}}\ddot{\mathbf{q}} + \widetilde{\boldsymbol{\xi}} + \mathbf{J}^T\mathbf{M} \left[\mathbf{M}^{-1} - \frac{\dot{\mathbf{x}}\dot{\mathbf{x}}^T}{\dot{\mathbf{x}}^T\mathbf{M}\dot{\mathbf{x}}} \right] (\mathbf{f} - \boldsymbol{\xi}), \quad (34)
\end{aligned}$$

where $\widetilde{\mathbf{M}} = \mathbf{J}^T\mathbf{M}\mathbf{J}$ and $\widetilde{\boldsymbol{\xi}} = \mathbf{J}^T(\boldsymbol{\xi} + \mathbf{M}\dot{\mathbf{J}}\dot{\mathbf{q}})$ form the standard pullback of $(\mathbf{M}, \boldsymbol{\xi})$.

Now we show that pulling back the geometry first then energizing it in \mathbf{q} results in a system matching Equation 34. We can calculate the geometry pullback with respect to the energy metric \mathbf{M} by pulling back the metric weighted force form of the geometry $\mathbf{M}\ddot{\mathbf{x}} + \mathbf{f} = \mathbf{0}$, where again $\mathbf{f} = \mathbf{M}\mathbf{h}$. The pullback is

$$\begin{aligned}
& \mathbf{J}^T\mathbf{M}(\mathbf{J}\ddot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{q}}) + \mathbf{J}^T\mathbf{f} = \mathbf{0} \\
\Rightarrow & (\mathbf{J}^T\mathbf{M}\mathbf{J})\ddot{\mathbf{q}} + \mathbf{J}^T(\mathbf{f} + \mathbf{M}\dot{\mathbf{J}}\dot{\mathbf{q}}) \\
\Leftrightarrow & \widetilde{\mathbf{M}}\ddot{\mathbf{q}} + \widetilde{\mathbf{f}} = \mathbf{0} \quad (35)
\end{aligned}$$

where $\widetilde{\mathbf{M}} = \mathbf{J}^T\mathbf{M}\mathbf{J}$ as before and $\widetilde{\mathbf{f}} = \mathbf{J}^T(\mathbf{f} + \mathbf{M}\dot{\mathbf{J}}\dot{\mathbf{q}})$.

Let $\widetilde{\mathcal{L}}_e = \mathcal{L}_e(\phi(\mathbf{q}), \mathbf{J}\dot{\mathbf{q}})$ be the pullback of the energy function \mathcal{L}_e . We know that the Euler-Lagrange equation commutes with the pullback, so applying the Euler-Lagrange equation to this pullback energy $\widetilde{\mathcal{L}}_e$ is equivalent to pulling back the Euler-Lagrange equation of \mathcal{L}_e . This means we can calculate the Euler-Lagrange equation of $\widetilde{\mathcal{L}}_e$ as

$$(\mathbf{J}^T\mathbf{M}\mathbf{J})\ddot{\mathbf{q}} + \mathbf{J}^T(\boldsymbol{\xi} + \mathbf{M}\dot{\mathbf{J}}\dot{\mathbf{q}}) = \mathbf{0} \quad (36)$$

$$\Leftrightarrow \widetilde{\mathbf{M}}\ddot{\mathbf{q}} + \widetilde{\boldsymbol{\xi}} = \mathbf{0}, \quad (37)$$

with $\widetilde{\mathbf{M}} = \mathbf{J}^T\mathbf{M}\mathbf{J}$ and $\widetilde{\boldsymbol{\xi}} = \mathbf{J}^T(\boldsymbol{\xi} + \mathbf{M}\dot{\mathbf{J}}\dot{\mathbf{q}})$ (both as previously defined). Therefore, energizing 35 with $\widetilde{\mathcal{L}}_e$ gives

$$\begin{aligned}
& \widetilde{\mathbf{M}}\ddot{\mathbf{q}} + \widetilde{\boldsymbol{\xi}} + \widetilde{\mathbf{M}} \left[\widetilde{\mathbf{M}}^{-1} - \frac{\dot{\mathbf{q}}\dot{\mathbf{q}}^T}{\dot{\mathbf{q}}^T\widetilde{\mathbf{M}}\dot{\mathbf{q}}} \right] (\widetilde{\mathbf{f}} - \widetilde{\boldsymbol{\xi}}) = \mathbf{0} \\
\Rightarrow & \widetilde{\mathbf{M}}\ddot{\mathbf{q}} + \widetilde{\boldsymbol{\xi}} \\
& + (\mathbf{J}^T\mathbf{M}\mathbf{J}) \left[\widetilde{\mathbf{M}}^{-1} - \frac{\dot{\mathbf{q}}\dot{\mathbf{q}}^T}{\dot{\mathbf{q}}^T\mathbf{J}^T\mathbf{M}\mathbf{J}\dot{\mathbf{q}}} \right] \\
& \left(\mathbf{J}^T(\mathbf{f} + \mathbf{M}\dot{\mathbf{J}}\dot{\mathbf{q}}) - \mathbf{J}^T(\boldsymbol{\xi} + \mathbf{M}\dot{\mathbf{J}}\dot{\mathbf{q}}) \right) = \mathbf{0} \\
\Rightarrow & \widetilde{\mathbf{M}}\ddot{\mathbf{q}} + \widetilde{\boldsymbol{\xi}} \\
& + (\mathbf{J}^T\mathbf{M})\mathbf{J} \left[(\mathbf{J}^T\mathbf{M}\mathbf{J})^{-1} - \frac{\dot{\mathbf{q}}\dot{\mathbf{q}}^T}{\dot{\mathbf{x}}^T\mathbf{M}\dot{\mathbf{x}}} \right] \mathbf{J}^T(\mathbf{f} - \boldsymbol{\xi}) = \mathbf{0} \\
\Rightarrow & \widetilde{\mathbf{M}}\ddot{\mathbf{q}} + \widetilde{\boldsymbol{\xi}} \quad (38) \\
& + \mathbf{J}^T\mathbf{M} \left[\mathbf{J}(\mathbf{J}^T\mathbf{M}\mathbf{J})^{-1}\mathbf{J}^T - \frac{\dot{\mathbf{x}}\dot{\mathbf{x}}^T}{\dot{\mathbf{x}}^T\mathbf{M}\dot{\mathbf{x}}} \right] (\mathbf{f} - \boldsymbol{\xi}) = \mathbf{0}.
\end{aligned}$$

Since ϕ is full rank $\mathbf{J}^T\mathbf{M}\mathbf{J}$ is invertible, so

$$\mathbf{J}^T\mathbf{M}\mathbf{J}(\mathbf{J}^T\mathbf{M}\mathbf{J})^{-1}\mathbf{J}^T = \mathbf{J}^T = \mathbf{J}^T\mathbf{M}(\mathbf{M}^{-1}),$$

which means we can rewrite Equation 38 as

$$\widetilde{\mathbf{M}}\ddot{\mathbf{q}} + \widetilde{\boldsymbol{\xi}} + \mathbf{J}^T\mathbf{M} \left[\mathbf{M}^{-1} - \frac{\dot{\mathbf{x}}\dot{\mathbf{x}}^T}{\dot{\mathbf{x}}^T\mathbf{M}\dot{\mathbf{x}}} \right] (\mathbf{f} - \boldsymbol{\xi}) = \mathbf{0}. \quad (39)$$

This final expression in Equation 39 matches the expression for the energized geometry pullback in Equation 34. \square

APPENDIX E

DETAILS ON DESIGNING WITH FABRICS

A. Derivations of Fabric Component Algebra

A fabric component was defined in Subsection ?? as a Finsler energy paired with a force term $\{\mathcal{L}_e, \mathbf{f}\}$. We often

refer to this pair as a Finsler energy paired with a *HD2 geometry* $\pi(\mathbf{x}, \dot{\mathbf{x}})$ knowing that the actual component would be $\mathbf{f} = -\mathbf{M}\pi$ where \mathbf{M} is the mass from the energy's equations of motion $\mathbf{M}\ddot{\mathbf{x}} + \boldsymbol{\xi} = \mathbf{0}$. Since the energization formulas in Theorem IV.5 are functions of the Finsler energy \mathcal{L}_e only through its equations of motion parameterized by $(\mathbf{M}, \boldsymbol{\xi})$, we can think of the fabric component as effectively a triple $(\mathbf{M}, \boldsymbol{\xi}, \mathbf{f})$ representing the energy equations alongside an additional forcing term, which itself can be viewed as two pairs $((\mathbf{M}, \boldsymbol{\xi}), (\mathbf{M}, \mathbf{f}))$. The algebra defined in Subsection ?? is equivalent to the reduction of sums and pullback to the sums and pullback of these individual pairs. For concreteness, and in alignment with the terminology of [20], we call these pairs *specs*, short for *spectral semi-sprays*, which is descriptive of the differential equation they represent emphasizing the spectral role of the metric \mathbf{M} .

The one operation on these specs not discussed in Subsection ?? is summation and the linearity of summation. This property can be easily seen by

$$\begin{aligned} & \alpha(\mathbf{M}_1, \mathbf{f}_1) + \beta(\mathbf{M}_2, \mathbf{f}_2) \\ & \Leftrightarrow \alpha(\mathbf{M}_1\ddot{\mathbf{x}} + \mathbf{f}_1) + \beta(\mathbf{M}_2\ddot{\mathbf{x}} + \mathbf{f}_2) = \mathbf{0} \\ & \Rightarrow (\alpha\mathbf{M}_1 + \beta\mathbf{M}_2)\ddot{\mathbf{x}} + (\alpha\mathbf{f}_1 + \beta\mathbf{f}_2) = \mathbf{0} \\ & \Leftrightarrow ((\alpha\mathbf{M}_1 + \beta\mathbf{M}_2), (\alpha\mathbf{f}_1 + \beta\mathbf{f}_2)). \end{aligned}$$

As mentioned in the discussion leading up to Equation 14, the pullback of energy equations is equivalent to applying the Euler-Lagrange equation on the pullback energy. Since application of the Euler-Lagrange equation is linear in the sense $\text{EL}[\alpha\mathcal{L}_1 + \beta\mathcal{L}_2] = \alpha\text{EL}[\mathcal{L}_1] + \beta\text{EL}[\mathcal{L}_2]$, if we treat any two energies that differ by only a constant as equivalent, the mapping defined by the $\text{EL}[\cdot]$ operator defines an isomorphism between energies and the subspace of specs defined by those energies. That means, algebraically, we can treat the two representations as equivalent.

That algebra of specs derives the summation operation on fabric components. For the pullback operation, we can simply note that by Equation 14, the first (energy) spec of the component has a pullback matching the application of the Euler-Lagrange equation applied to the pullback energy, so the pullback of the energy captures that part. It additionally captures the pullback of the mass matrix of the second spec, so we need only additionally account for how that second spec's force term pulls back. Looking at the definition of component pullback in Equation 12 we see that that's exactly what the force term pullback expression computes.

Finally, we note that while the definition of fabric components and its associated algebra as presented in Subsection ?? is compact and gives insight into what fabric components are and how they interact, it may be more efficient in practice to represent them in compact spec form explicitly as triples $(\mathbf{M}, \boldsymbol{\xi}, \mathbf{f})$ in code. That said, leveraging automatic differentiation frameworks such as those used in [22] may lead to alternative efficient representations.

We note that some fabric components may represent potential terms with dampers as their \mathbf{f} term. These components intermix with the other fabric components using the same alge-

bra since a component is defined independent of \mathbf{f} semantics.

B. Formulas for speed control

This section gives explicit formulas for calculating the damping $\beta(\mathbf{q}, \dot{\mathbf{q}})$ described in Subsection V-A for regulating a specific measure of speed.

Once a geometry is energized by a Finsler energy \mathcal{L}_e , its speed profile is defined by what is necessary to conserve \mathcal{L}_e . In practice, we usually want to regulate a different measure of energy (speed), an *execution energy* $\mathcal{L}_e^{\text{ex}}$, such as Euclidean energy in either C-space or end-effector space (or, more commonly, some combination of both). Denoting the fabric by $(\mathcal{L}_e, -\mathbf{M}\pi - \boldsymbol{\xi})$ which defines nominal fabric equations

$$\begin{aligned} & \mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\xi} + \mathbf{f}_f \\ & = \mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\xi} - \mathbf{M}\pi - \boldsymbol{\xi} \\ & = \mathbf{M}(\ddot{\mathbf{q}} - \pi) = \mathbf{0} \end{aligned}$$

so that the resolved behavior matches the geometric policy $\ddot{\mathbf{q}} = \pi$, we can regulate the speed by adjusting α_{reg} in a forced and damped equation of the form

$$\ddot{\mathbf{q}} = -\mathbf{M}^{-1}\partial_{\mathbf{q}}\psi(\mathbf{q}) + \pi(\mathbf{q}, \dot{\mathbf{q}}) + \alpha_{\text{reg}}\dot{\mathbf{q}}. \quad (40)$$

As long as $\alpha_{\text{reg}} < \alpha_{\mathcal{L}_e}$, where $\alpha_{\mathcal{L}_e}$ is the energization coefficient (Equation 7 of Theorem IV.5), the system will remain stable per Theorem IV.4 since α_{reg} always act as a nonzero damper with respect to the energized system. Note that $\alpha_{\mathcal{L}_e}$ need not always be smaller than 0, so we have the flexibility with α_{reg} to both remove and add energy as needed (although, we don't necessarily have to use that flexibility; the method outlined below restricts it to being positive to aid intuition). Below we calculate explicit formulas for α_{reg} to regulate a given measure of speed (energy).

Let $\mathcal{L}_e^{\text{ex}}$ be an *execution energy*, which may differ from the fabric's energy \mathcal{L}_e , and let α_{ex}^0 and α_{ex}^ψ be the energization coefficients, respectively, for $\text{energize}_{\mathcal{L}_e^{\text{ex}}}[\pi]$ and $\text{energize}_{\mathcal{L}_e^{\text{ex}}}\left[-\mathbf{M}^{-1}\partial_{\mathbf{q}}\psi + \pi\right]$. The coefficient α_{ex}^0 ensures that system

$$\ddot{\mathbf{q}} = \pi(\mathbf{q}, \dot{\mathbf{q}}) + \alpha_{\text{ex}}^0\dot{\mathbf{q}} \quad (41)$$

conserves $\mathcal{L}_e^{\text{ex}}$ and the coefficient α_{ex}^ψ ensures that the *forced system*⁵

$$\ddot{\mathbf{q}} = -\mathbf{M}^{-1}\partial_{\mathbf{q}}\psi + \pi(\mathbf{q}, \dot{\mathbf{q}}) + \alpha_{\text{ex}}^\psi\dot{\mathbf{q}} \quad (42)$$

conserves $\mathcal{L}_e^{\text{ex}}$. Likewise, let $\alpha_{\mathcal{L}_e}$ denote the energy coefficient of the actual system energy $\text{energize}_{\mathcal{L}_e}[\pi]$ (designed to conserve \mathcal{L}_e).

The forced and damped energized system

$$\ddot{\mathbf{q}} = -\mathbf{M}^{-1}\partial_{\mathbf{q}}\psi + \pi(\mathbf{q}, \dot{\mathbf{q}}) + \alpha_{\mathcal{L}_e}\dot{\mathbf{q}} - \tilde{\beta}\dot{\mathbf{q}}, \quad (43)$$

⁵As noted above, the energization coefficient calculation in Theorem IV.5 does not require that the underlying differential equation being energized be a geometry (in this case it's a geometry forced by a potential). It simply states that if the underlying differential equation is indeed a geometry, then the resulting energized equation is a geometric fabric. Here we're just using the result to calculate a reference coefficient.

would be stable for $\tilde{\beta} > 0$. Adding and subtracting α_{ex}^0 gives

$$\begin{aligned}\ddot{\mathbf{q}} &= -\mathbf{M}^{-1}\partial_{\mathbf{q}}\psi + \pi(\mathbf{q}, \dot{\mathbf{q}}) + (\alpha_{\mathcal{L}_e} + \alpha_{\text{ex}}^0 - \alpha_{\text{ex}}^0)\dot{\mathbf{q}} - \tilde{\beta}\dot{\mathbf{q}} \\ &= -\mathbf{M}^{-1}\partial_{\mathbf{q}}\psi + \pi(\mathbf{q}, \dot{\mathbf{q}}) + \alpha_{\text{ex}}^0\dot{\mathbf{q}} - \beta\dot{\mathbf{q}} \\ &= -\mathbf{M}^{-1}\partial_{\mathbf{q}}\psi + \text{energize}_{\mathcal{L}_{\text{ex}}}[\pi(\mathbf{q}, \dot{\mathbf{q}})] - \beta\dot{\mathbf{q}},\end{aligned}$$

where $\beta = \tilde{\beta} + \alpha_{\text{ex}}^0 - \alpha_{\mathcal{L}_e}$. The constraint $\tilde{\beta} > 0$ written this way manifests as $\beta > \alpha_{\text{ex}}^0 - \alpha_{\mathcal{L}_e}$.

This calculation suggests we can regulate speed by choosing α_{reg} in Equation 40 as

$$\alpha_{\text{reg}} = \alpha_{\text{ex}}^\eta - \beta_{\text{reg}}(\mathbf{q}, \dot{\mathbf{q}}) + \alpha_{\text{boost}} \quad (44)$$

where $\alpha_{\text{ex}}^\eta = \eta\alpha_{\text{ex}}^0 + (1 - \eta)\alpha_{\text{ex}}^\psi$ for $\eta \in [0, 1]$ and $\alpha_{\text{boost}} \leq 0$ as long as $\beta_{\text{reg}} > \alpha_{\text{ex}}^0 - \alpha_{\mathcal{L}_e}$. The term α_{boost} can be used temporarily in the beginning to boost the system up to speed. Since it is transient, we drop α_{boost} momentarily for simplicity, but return to it at the end of the section to describe a good boosting policy. In our experiments, we use that term only for the cubby navigation problem in order to reduce the strength of the potential function near the beginnings of motions to get a clear signal on what the geometry itself is encoding.

Under this choice of α_{reg} given in Equation 44, using $\mathbf{v}_{\parallel} = (\alpha_{\text{ex}}^\psi - \alpha_{\text{ex}}^0)\dot{\mathbf{q}}$ to denote the component of $-\mathbf{M}^{-1}\partial_{\mathbf{q}}\psi$ additionally removed by including it within the energization operation (and dropping α_{boost}), we can express the system as

$$\ddot{\mathbf{q}} = \eta\mathbf{v}_{\parallel} + \text{energize}_{\mathcal{L}_{\text{ex}}}[-\mathbf{M}^{-1}\partial_{\mathbf{q}}\psi + \pi] - \beta_{\text{reg}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}},$$

using a similar analysis to the above. Intuitively, the energized system will conserve execution energy, $\eta\mathbf{v}_{\parallel}$ can inject energy into the system to speed it up (by an amount parameterized by $\eta \in [0, 1]$), and β_{reg} can bleed off energy. Note that the lower bound on the damper isn't strictly zero $\beta_{\text{reg}} > \alpha_{\text{ex}}^0 - \alpha_{\mathcal{L}_e}$, so there may be both cases where that term is allowed to inject extra energy and when that term is forced to be strictly a margin above zero for stability.

In practice, we enforce that β_{reg} is strictly positive by choosing $\beta_{\text{reg}} = s_\beta(\mathbf{q})B + \underline{B} + \max\{0, \alpha_{\text{ex}}^\eta - \alpha_{\mathcal{L}_e}\} > 0$ with both B and \underline{B} positive. This gives us a more intuitive interface to speed control with η adjusting energy in and β_{reg} adjusting energy out. In this form, \underline{B} acts as a constant baseline damping coefficient, and $s_\beta(\mathbf{q})B$ provides additional damping near the convergence point with $B > 0$ and $s_\beta(\mathbf{q})$ acting as a switch transitioning from 0 to 1 as the system approaches the target. $\max\{0, \alpha_{\text{ex}}^\eta - \alpha_{\mathcal{L}_e}\}$ ensures the stability bound $\alpha_{\text{reg}} = \beta_{\text{reg}} \geq \alpha_{\text{ex}} - \alpha_{\mathcal{L}_e}$ is satisfied with the baseline coefficients making it a strict inequality.

Specifically, in our experiments we use

$$s_\beta(\mathbf{q}) = \frac{1}{2} \left(\tanh(-\alpha_\beta(\|\mathbf{q}\| - r)) + 1 \right) \quad (45)$$

where $\alpha_\beta \in \mathbb{R}^+$ is a gain defining the switching rate, and $r \in \mathbb{R}^+$ is the radius where the switch is half-way engaged. Denoting the desired execution energy as $\mathcal{L}_e^{\text{ex},d}$, we use the following policy for η

$$\eta = \frac{1}{2} \left(\tanh(-\alpha_\eta(\mathcal{L}_e^{\text{ex}} - \mathcal{L}_e^{\text{ex},d}) - \alpha_{\text{shift}}) + 1 \right) \quad (46)$$

where $\alpha_\eta, \alpha_{\text{shift}} \in \mathbb{R}^+$ adjust the rate and offset, respectively, of the switch as an affine function of the speed (execution energy) error.

When used, α_{boost} acts to explicitly inject energy along the direction of motion to quickly boost the system up to speed. When $\dot{\mathbf{q}} = \mathbf{0}$, this term has no effect, so the initial direction of motion is chosen by the potential $\partial_{\mathbf{q}}\psi$. Once $\dot{\mathbf{q}} \neq \mathbf{0}$, α_{boost} quickly accelerates the system to the desired speed. When π is a geometry, quickly reaching a high speed means that the influence of the non-geometric potential is diminished, promoting path consistency. We reduce α_{boost} to zero permanently once the system is close to convergence so it does not affect convergence stability.

In these cases, α_{boost} is modeled as $\alpha_{\text{boost}} = k\eta(1 - s_\beta(\mathbf{q}))\frac{1}{\|\dot{\mathbf{q}}\| + \epsilon}$, where $k \in \mathbb{R}^+$ is a gain that directly sets the desired level of acceleration, η (from above) sets $\alpha_{\text{boost}} = 0$ when the desired speed is achieved, and $1 - s_\beta(\mathbf{q})$ sets $\alpha_{\text{boost}} = 0$ when the system is within the region of higher damping. The normalization by $\|\dot{\mathbf{q}}\| + \epsilon$ ensures that α_{boost} is directly applied along $\hat{\mathbf{q}}$ with a very small positive value for ϵ to ensure numerical stability. This overall design injects more energy into the system when $-\alpha_{\text{boost}} < \alpha_{\text{ex}}^\eta - \alpha_{\mathcal{L}_e}$. Since this injection occurs for finite time, the total system energy is still bounded. The additional switches ensure that the system is still subject to positive damping, guaranteeing convergence.

C. Designing fabric components

This section reviews a number of techniques for designing HD2 functions and maps for the design of Finsler energies and HD2 geometries referenced in Subsection V-B on tools for fabric design.

The design of fabric components largely involves the construction of a geometric policy expressing a desired component behavior and a Finsler energy whose metric will act as that policy's priority metric. Both of these parts are defined by HD2 terms, so the question of design largely reduces to the question of how to design HD2 functions and maps.

As a reminder, a Homogeneous of Degree k (HD k) function is a function $f(\mathbf{x})$ for which $f(\alpha\mathbf{x}) = \alpha^k f(\mathbf{x})$. In our case, our functions are typically HD0 through HD2, specifically in velocity. For instance, if $\pi(\mathbf{x}, \dot{\mathbf{x}})$ is a geometric policy, the *geometric* modifier means it is HD2 in velocity, so $\pi(\mathbf{x}, \alpha\dot{\mathbf{x}}) = \alpha^2\pi(\mathbf{x}, \dot{\mathbf{x}})$, and usually we assuming this holds just for $\alpha \geq 0$. It places no restrictions on how the function should scale with \mathbf{x} .

Often we design HD2 terms by multiplying an HD0 term by an HD2 scaling factor. That HD0 term might be entirely independent of velocity, or it may be dependent on the direction of the velocity, but not its norm (speed). Note that in the 1D case, the velocity direction is a switch, the sign of the scalar one-dimensional velocity, switching from 1 to -1 as velocity passes from positive to negative. This discrete switch is valid with our theory (systems have to come to a rest (zero velocity) before experiencing the switch) and is used commonly in our designs, especially for one-dimensional barrier geometries and energies.

In higher dimensions, switches depending on the norm of a velocity may not result in positive definite mass matrices.

The presentation in [7] allows that case by relaxing the third requirement of Definition III.1 to require only invertibility, which is internally consistent with the geometric theory. But here, as a generalization of classical mechanics, we want mass matrices to remain positive definite, so building such switching terms in 1D spaces (where is naturally positive definite) and pulling them back to higher-dimensional spaces becomes an important technique for Finsler energy design.

A common pattern is to start with a scalar function $\psi(\mathbf{x})$ (often similar in structure and form to standard potential functions), and either use it directly as a position only base function, or compute a base policy as $\pi_0(\mathbf{x}) = -\partial_{\mathbf{x}}\psi$. In this discussion we will use the base policy as an example and construct a geometric policy. This base policy is HD0 simply because it doesn't depend on velocity at all. It's often useful to depend on velocity direction $\hat{\mathbf{x}} = \dot{\mathbf{x}}/\|\dot{\mathbf{x}}\|$, though. Since $\hat{\mathbf{x}}$ is already independent of speed $s = \|\dot{\mathbf{x}}\|$, any function of velocity direction is automatically HD0 as well. That gives $\pi_0^\sigma(\mathbf{x}, \dot{\mathbf{x}}) = \sigma(\hat{\mathbf{x}})\pi_0(\mathbf{x})$ as a new HD0 base. Finally, any Finsler energy $\mathcal{L}_e(\mathbf{x}, \dot{\mathbf{x}}) = \frac{1}{2}\mathcal{L}_g^2(\mathbf{x}, \dot{\mathbf{x}})$ is HD2 by definition, so scaling π_0^σ by \mathcal{L}_e gives us an HD2 geometry

$$\pi(\mathbf{x}, \dot{\mathbf{x}}) = \mathcal{L}_e(\mathbf{x}, \dot{\mathbf{x}}) \pi_0^\sigma(\mathbf{x}, \dot{\mathbf{x}}). \quad (47)$$

For instance, we might choose

$$\psi(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}_0 - \mathbf{x}\|^2 \quad (48)$$

as a quadratic attractor to a target \mathbf{x}_0 . The negative gradient of the potential defines $\pi_0(\mathbf{x}) = -\partial\psi = \mathbf{x}_0 - \mathbf{x}$. Then with a slight abuse of notation, we might switch on the velocity direction using a scalar σ of the form $\sigma(s) = 1/(1 + \exp(-\beta s))$ for $\beta \in \mathbb{R}_+$ with $s = \mathbf{v}^T \hat{\mathbf{x}}$ for some constant vector \mathbf{v} . The resulting HD0 policy becomes

$$\begin{aligned} \pi_0^\sigma(\mathbf{x}, \dot{\mathbf{x}}) &= \sigma(\mathbf{v}^T \hat{\mathbf{x}}) \pi_0(\mathbf{x}) \\ &= \frac{\mathbf{x}_0 - \mathbf{x}}{1 + \exp(-\beta \mathbf{v}^T \hat{\mathbf{x}})}. \end{aligned} \quad (49)$$

Finally, we can choose to scale that using a Euclidean Finsler energy $\mathcal{L}_e = \frac{1}{2}\|\dot{\mathbf{x}}\|^2$ to get a final geometry of the form

$$\pi(\mathbf{x}, \dot{\mathbf{x}}) = \frac{\frac{1}{2}\|\dot{\mathbf{x}}\|^2(\mathbf{x}_0 - \mathbf{x})}{1 + \exp(-\beta \mathbf{v}^T \hat{\mathbf{x}})} \quad (50)$$

Note that another way to design an HD2 geometry is by designing a Finsler geometry derived from a Finsler energy. We can use one Finsler energy to derive the geometry, and a second Finsler energy to define the priority metric. Energization would then bend the geometry associated with the second (priority) energy to align with the first energy's geometry.

A common pattern for designing Finsler energies is to start with a Riemannian (kinetic) energy in 1D of the form $\mathcal{K}(x, \dot{x}) = \frac{1}{2}g(x)\dot{x}^2$ where $g(x)$ is everywhere positive and dependent on position only. We then scale it by a scaling function that depends on velocity directionality. Since we are in 1D, that normalized velocity becomes a switch on the sign of the velocity, so we can design this scaling function in two

parts

$$\sigma(\dot{x}) = \begin{cases} \sigma_-(\dot{x}) & \text{for } \dot{x} \leq 0 \\ \sigma_+(\dot{x}) & \text{for } \dot{x} > 0 \end{cases}. \quad (51)$$

The energy then becomes $\mathcal{L}_e(x, \dot{x}) = \frac{1}{2}\tilde{g}(x, \dot{x})\dot{x}^2$ where $\tilde{g}(x, \dot{x}) = \sigma(\dot{x})g(x)$, which is effectively two different metrics depending on directionality of the velocity.

Note that in the 1D case, the derived $m(x, \dot{x}) = \partial_{\dot{x}\dot{x}}^2 \tilde{g}$ must have the property that

$$\mathcal{L}_e(x, \dot{x}) = \frac{1}{2}\tilde{g}(x, \dot{x})\dot{x}^2 = \frac{1}{2}m(x, \dot{x})\dot{x}^2, \quad (52)$$

which is only true globally when $m = \tilde{g}$. This in general is not necessarily true in higher dimensions.

Finally, we describe a common pattern in designing potential functions. Potential functions define forces that push against the fabric system's mass. The mass defines the fabric priority; the stronger the mass the higher the priority the fabric takes. Thought another way, these masses define the fabric system's inertia; a given force has less effect pushing a sizable mass (high priority) than pushing a small mass (low priority). It is often important that a potential function adjust effectively to the fabric's priority. However, it can be challenging to design potential forces directly to scale in an intuitive way.

For that reason, it's often easier to design a potential function by designing its gradient in terms of a priority metric of its own and a desired *acceleration*. Specifically, if $\tilde{\psi}(\mathbf{x})$ is a function (similar to the potential function in form) defining an acceleration policy $\pi(\mathbf{x}) = -\partial\tilde{\psi}$, and if \mathbf{M} is its associated priority, the resulting potential function $\psi(\mathbf{x})$ would be one for which $-\partial\psi = -\mathbf{M}\partial\tilde{\psi}$ (force equals mass times acceleration) if such a potential function exists. Given an expression for a vector field, it's easy to check whether it is a gradient of a potential by taking its Jacobian and verifying the Jacobian is symmetric (it's a well-known result in multivariate calculus that a vector field is the gradient of a potential if and only if its Jacobian is everywhere symmetric).

One-dimensional vector fields are always symmetric so we can always design one-dimensional potentials using an acceleration policy and an associated priority mass. For higher-dimensional potentials, one can use the results discussed in [8] (Appendix D.4) that show that if the acceleration policy is radially symmetric and the metric is compatibly symmetric along the same radial lines, the resulting vector field defined by scaling the acceleration policy by the metric has symmetric Jacobian and is therefore the gradient of some potential.

When designing potential functions in this way, it is necessary only to show that the potential exists. Knowing that, we can implement the system using only its gradient given by the metric-scaled acceleration policy. The resulting system then fits the theory and we have convergence and stability guarantees.

APPENDIX F

ADDITIONAL EXPERIMENTS: EQUALITY CONSTRAINTS

This supplementary experiment is a specific realization of constrained fabrics. In principle, constrained fabrics can be realized in many different ways, e.g., by explicitly calculating

the constraint Lagrange multiplier and adding the resulting constraint force to the unconstrained fabric as discussed in Appendix D-B. This method would additionally need to employ Baumgarte stabilization [26] to combat numerical drift. Other approaches may be taken as well such as solving a nonlinear program subject to a discretized momentum-impulse form of the fabric and additional equality constraints as done by Stewart and Trinkle [27]. There are also many other different discretization and nonlinear program schemes that could be created to solve constrained fabrics. In this particular case, we create the following discretization,

$$\mathbf{q} = \mathbf{q}_k \quad (53)$$

$$\dot{\mathbf{q}} = \frac{\mathbf{q}_{k+1} - \mathbf{q}_{k-1}}{2\Delta t} \quad (54)$$

$$\ddot{\mathbf{q}} = \frac{\mathbf{q}_{k+1} - 2\mathbf{q}_k + \mathbf{q}_{k-1}}{\Delta t^2} \quad (55)$$

where k indicates the discrete time index and \mathbf{q}_{k-1} and \mathbf{q}_k are known. We then minimize the following cost function,

$$L(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2}(\ddot{\mathbf{q}}_d - \ddot{\mathbf{q}})^T \mathbf{M}(\mathbf{q}, \dot{\mathbf{q}})(\ddot{\mathbf{q}}_d - \ddot{\mathbf{q}}) + \frac{1}{2}\lambda \mathbf{C}(\mathbf{q}_{k+1})^2 \quad (56)$$

where $\ddot{\mathbf{q}}_d$ is the desired unconstrained acceleration of the fabric calculated as $\ddot{\mathbf{q}}_d = -\mathbf{M}(\mathbf{q}, \dot{\mathbf{q}})^{-1}\mathbf{f}(\mathbf{q}, \dot{\mathbf{q}})$, $\mathbf{C}(\mathbf{q}_{k+1})$ is a scalar constraint function of interest, and λ is a weight. The first term solves for system acceleration that naturally follows the fabric while the second term is a penalty on the constraint, where λ can be set to a large value to enforce the constraint. Altogether, this formulation implicitly solves for a new acceleration that adds a constraint force on the unconstrained fabric theoretically realizing (5).

We implemented the above for the geometric fabric designed in Section VII subject to the constraint that the end-effector must lie on a plane. We repeatedly solved the program with $\Delta t = 0.04s$ and a Gauss-Newton optimizer with four iterations per time step to forward integrate the system. The result is shown in Fig. 5. The robot can maintain the constraint at the micrometer level and the end-effector can settle to within 1 mm of the closest point on the plane to the target.

APPENDIX G

END-EFFECTOR POSE CONTROL

There are many different ways geometric fabrics can be used to control a robot, including end-effector pose control. This is a straight-forward extension to design outlined in Section VII that attracted a point on the end-effector towards a target point. In this case, we calculate three additional task spaces that map to each axis of a orthonormal coordinate system placed at the end-effector. In each of these axis spaces, we place the very same attractor components discussed in Section VII-A3. With this extension to the original fabric, we can exhibit full pose control over the end-effector.

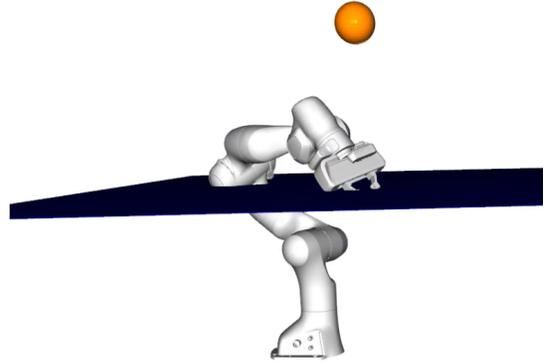


Fig. 5. Constrained geometric fabric moving the robot end-effector to orange target points while the end-effector is constrained to the blue plane.

APPENDIX H

ADDITIONAL EXPERIMENTS: CUBBY NAVIGATION HEURISTICS

An important use of geometric fabrics is to heuristically bias the system without affecting convergence to local minima of the potential. This feature is especially useful for designing global navigation heuristics. A geometric fabric system as described in Subsection VII is subject to local minima since the target attractor is greedy. In many environments, though, simple heuristics can coarsely shape the behavior en route so that these local avoidance behaviors are sufficient for global navigation as well (similar to the heuristics described in [21] Section V-D).

Section XI of [28] describes such an experimental setup in detail. In those experiments, geometric fabrics are used to navigate between a wall of cubbies and a set of floor boxes in a standard pick and place industrial setting. The heuristic geometries encoded simple geometric fields that push the system away from the interior of any cubby and box except for the one it wanted to enter into. The final system successfully navigated between any pair of cubby and surrounding boxes even with perturbations to the environment.

The simplicity and success of these heuristic terms suggest that learning such heuristics conditioned on environmental features over the top of an existing obstacle navigation fabric system would be straightforward, especially given automatic differentiation tools such as [22] and leveraging imitation learning methods such as [29]. Fabrics constitute a compact encoding of behavioral components that span multiple tasks which could make it a powerful medium for learning of strongly generalizing policies. It is an area of future work to explore these learning applications in detail, especially in the context of meta learning shared nominal fabric layers.