Kinetic Energy Fields: A Solution of Riemannian Eikonal Equation on Configuration Space Manifold

Yiming Li[1,2,*], Jiacheng Qiu[1,*] and Sylvain Calinon[1,2]
[1]Idiap Research Institute [2]EPFL * Equal Contribution

***Abstract*—This paper presents a geometric representation for energy-efficient motion generation based on kinetic energy fields. Minimal energy paths correspond to geodesics in the configuration space manifold that satisfy equations of motion. Specifically, the kinetic energy corresponds to the constant-velocity curves on the Riemannian manifold endowed with a symmetric positive-definite kinetic energy metric. We introduce a wave propagation model for energy fields and geodesic flows by solving a first-order partial differentiable equation (PDE): the eikonal equation on the Riemannian manifold. A neural Riemannian Eikonal solver is proposed to handle high-dimensional spaces, leading to a compact and grid-free representation. Given a specific kinematics chain, the kinetic energy field is trained offline and allows efficient reactive motion generation to be computed online, enabling the rapid generation of paths from arbitrary start and goal configurations. We present preliminary results on generating energy-efficient motions on planar robot examples and a 7-axis Franka robot.**

I. Introduction

Motion generation is a long-standing problem in robotics. Various approaches based on inverse kinematics [3, 4] and planning [20, 11, 9] techniques have been proposed to generate feasible trajectories to satisfy different constraints such as goal reaching, obstacle avoidance, and minimum path lengths. These methods plan motions at the kinematics level and subsequently develop controllers to track these planned trajectories. However, the underlying dynamics are often ignored, which may lead to instability and robustness problems, potentially also increasing energy consumption. Due to the high nonlinearity, dynamics-aware motion generation remains a challenging problem.

Studies in differential geometry have opened up new perspectives on this problem by viewing motions as geodesics in the configuration space manifold whose geometric properties are entirely defined by the underlying dynamics [2]. For instance, the kinetic energy we focused on, corresponds to geodesic lengths on the Riemannian manifold defined by kinetic energy metric [10]. Several approaches have been proposed to compute geodesics on the manifold, by solving the geodesic equations or using optimization techniques to find the approximated geodesics [10, 1, 13]. However, these methods either suffer from computational complexity issues that are hard to scale to high-dimensional systems [1, 10], or can easily converge to poor local minima and result in distorted geodesics [13].

In this paper, we use a kinetic energy field to represent the energy landscape in the configuration space manifold. This is inspired by the use of a distance field [15, 14], which is popular in motion planning because it can provide distance and gradient information. The kinetic energy field inherits these advantages and also encodes dynamics properties by solving the eikonal equation on the Riemannian manifold. Minimal geodesics are backtracked through gradient flows of the energy field.

Eikonal equation is a first-order nonlinear partial differential equation (PDE) that describes the evolution of wave propagation, whose solution corresponds to the shortest arrival time from a source point to any other goal point under a given velocity field [22]. Here, we consider the Riemannian eikonal equation, where the velocity is anisotropic and defined by the kinetic energy metric. The kinetic energy field is the solution of this differential equation.

We present two approaches for the Riemannian eikonal equation: a numerical and a neural eikonal solver. The numerical solver relies on discrete differential geometry, offering accurate and interpretable results but poor scaling properties [5, 16]. In contrast, neural PDE solvers provide grid-free solutions and show greater flexibility and scalability in handling complex geometries and high dimensions [18, 12, 19].

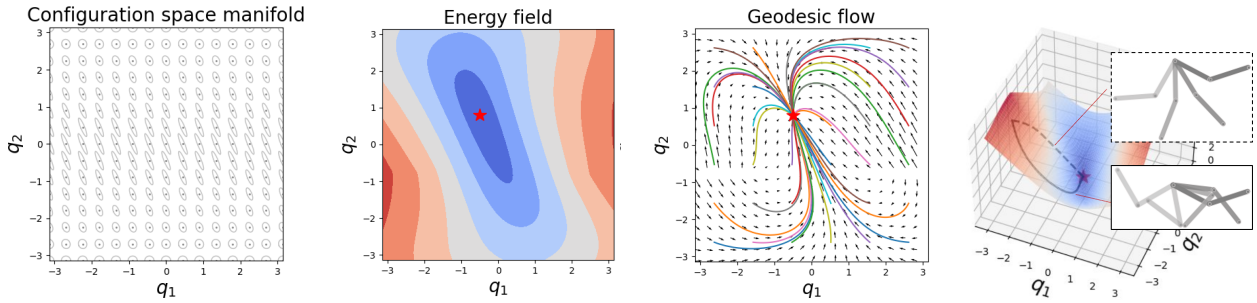
The neural network parameterization for kinetic energy fields and geodesic flows introduces additional benefits. The network model only depends on the intrinsic inertia-mass matrix given by a robot manipulator. Once the network is trained, it can be used online for the generation of energy-efficient paths for arbitrary source and target joint configurations. In addition, it provides a compact structure for continuous-time fields. Finally, the gradient can be computed through network backpropagation, accounting for analytical geodesic flows with high efficiency. The geodesic flow of the energy field can also be viewed as a local policy to integrate to other motion planning frameworks to satisfy constraints such as joint limits and collision avoidance. We demonstrate our preliminary results on planar robot examples and on a 7-axis Franka robot.

II. Background

We briefly introduce here the mathematical background of the Riemannian manifold, geodesics, kinetic energy metric, and eikonal equation (see [2] for details).

A. Riemannian Metrics and Geodesics

A d-dimensional Riemannian manifold $\mathcal{M}$ is a topological space with a smooth metric tensor $\boldsymbol{A}(\boldsymbol{x})$ defined at each point $\boldsymbol{x} \in \mathcal{M}$. The metric tensor $\boldsymbol{A}(\boldsymbol{x})$ is a positive definite



(a) (b) (c) (d)

Fig. 1: An illustration of our approach. (a) Configuration space manifold endowed with a Riemannian metric equal to the inverse of the inertial matrix (visualized as ellipsoids). The geodesics on this manifold correspond to the minimal kinetic energy paths. We solve the eikonal equation on this manifold, accounting for the kinetic energy field (b) and gradient flow (c) that can be used to backtrack geodesics (the source point is fixed for visualization). (d) Geodesic path (solid line) and Euclidean path (dashed line) on this manifold with corresponding robot motions.

symmetric matrix, defining a Riemannian metric. For each point $\boldsymbol{x} \in \mathcal{M}$, there is a tangent space $T_{\boldsymbol{x}}\mathcal{M}$, that locally linearizes the manifold. The inner product of two velocity vectors $\boldsymbol{u}, \boldsymbol{v}$ in tangent space $T_{\boldsymbol{x}}\mathcal{M}$ at $\boldsymbol{x} \in \mathcal{M}$ is defined as $\langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\boldsymbol{A}} = \boldsymbol{u}^T \boldsymbol{A}(\boldsymbol{x}) \boldsymbol{v}$ and the Riemannian norm $\|\boldsymbol{u}\|_{\boldsymbol{A}}$ of $\boldsymbol{u}$ is defined as $\|\boldsymbol{u}\|_{\boldsymbol{A}} = \sqrt{\langle \boldsymbol{u}, \boldsymbol{u} \rangle_{\boldsymbol{A}}}$. The definition of inner product and Riemannian norm enables the measurement of vector lengths and angles in tangent space $T_{\boldsymbol{x}}\mathcal{M}$ so that we can define the Riemannian distance between two points $\boldsymbol{x}_1, \boldsymbol{x}_2$ as

$$\mathcal{L}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \min_{\boldsymbol{x}(t)} \int_{t_0}^{t_1} \|\dot{\boldsymbol{x}}(t)\|_{\boldsymbol{A}(\boldsymbol{x})} \, dt, \tag{1}$$

where $\boldsymbol{x}(t)$ is a smooth curve connecting $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ with $\boldsymbol{x}(t_0) = \boldsymbol{x}_1$ and $\boldsymbol{x}(t_1) = \boldsymbol{x}_2$. Geodesics are the shortest curves with constant velocity connecting $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$.

B. Configuration Space Manifold

In robotics, the configuration space corresponds to the set of all possible joint configurations of a robot. It can be thought of as a manifold that locally resembles Euclidean space but can have a more complex global structure that geometrically reflects the robot's dynamics. In terms of kinetic energy, this manifold incorporates the mass-inertia matrix that curves the space and accounts for nonlinear properties. It is given by

$$U = \frac{1}{2} \dot{\boldsymbol{q}}^T \boldsymbol{M}(\boldsymbol{q}) \dot{\boldsymbol{q}}, \tag{2}$$

where $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$ are joint position and velocity vectors. $\boldsymbol{M}(\boldsymbol{q})$ is the inertial matrix (kinetic metric), defining a Riemannian manifold in configuration space. The geodesic equation can be derived by applying the Euler-Lagrange equation

$$\ddot{q}_i + \sum_{j,k} \Gamma^i_{jk} \dot{q}_j \dot{q}_k = 0, \tag{3}$$

where $\Gamma^i_{jk} = \frac{1}{2} \sum_{l=1}^{n} M^{-1}_{il} \left(\frac{\partial M_{lj}}{\partial q_k} + \frac{\partial M_{lk}}{\partial q_j} - \frac{\partial M_{jk}}{\partial q_l} \right)$ are the Christoffel symbols that describe the curvature of the manifold, and M_{lj}, q_i are l, j-th and i-th element of $\boldsymbol{M}$ and $\boldsymbol{q}$. This equation is also equivalent to the standard equation of motion $\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}$ in the absence of gravity term $\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{0}$ and external forces $\boldsymbol{\tau}$. The joint velocity $\|\dot{\boldsymbol{q}}\|_{\boldsymbol{M}}$ is constant along the geodesic. The solution of this geodesic equation corresponds to the kinetic energy optimal path.

C. Eikonal Equation

The eikonal equation is a nonlinear first-order PDE $\|\dot{\boldsymbol{q}}\| = c(\boldsymbol{q})$ that describes the propagation of wavefronts. Given a speed model, its solution is the arrival time from a source point to the front, corresponding to the continuous shortest path. The signed distance field (SDF) is a special case when $c(\boldsymbol{q}) = 1$. A typical numerical approach to solve this PDE is the Fast Marching Method (FMM) [22], requiring the discretization of the domain into a mesh.

The Riemannian eikonal equation is a variant of the standard eikonal equation $\|\dot{\boldsymbol{q}}\|_{\boldsymbol{M}} = c(\boldsymbol{q})$. It describes the propagation of wavefronts in a Riemannian manifold, corresponding to the minimal distance traveled along the curved surface of the manifold. We consider the kinetic energy in configuration space in the absence of obstacles with unit velocity $\|\dot{\boldsymbol{q}}\|_{\boldsymbol{M}} = 1$ to build the energy map.

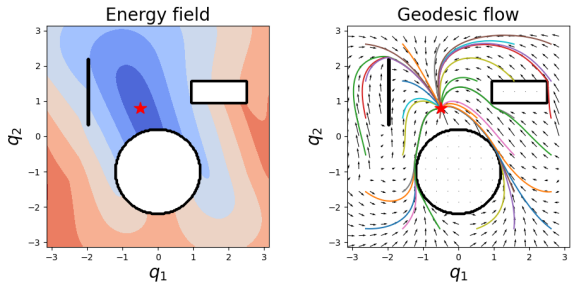
III. Riemannian Eikonal Solver

We present two approaches to solve the Riemannian eikonal equation for the kinetic energy field: a numerical method that discretizes the configuration space to a Cartesian grid, generalizing the fast marching method to Riemannian manifold, and a neural Riemannian eikonal solver (NES). Riemannian fast marching provides accurate solutions. However, it leads to significant computational and memory efficiency in high dimensions and does not scale well to high-resolution grids [16]. Neural PDE solvers provide a grid-free structure that computes gradients through network propagation, showing better scalability and efficiency. However, the training process can be complex and the convergence is not guaranteed.

A. Riemannian Fast Marching

Starting from a set of initial points, the fast marching method updates the travel times of its neighbors iteratively based on the eikonal equation and the Riemannian metric, until all points on the grid have been processed, accounting for travel times corresponding to the energy field. The accuracy of the algorithm depends on the resolution of the grid and the anisotropy of the Riemannian metric.

We test this approach by considering a 2D robot with link lengths $l_1 = l_2 = 2$ and masses $m_1 = m_2 = 1$ concentrated



(a) (b)

Fig. 2: The solution of Riemannian eikonal equation in case of obstacles (black contours) calculated using fast marching. (a) Kinetic energy field. (b) Geodesic flow.

at the articulations, with joint limits from $-\pi$ to π. Therefore, the inertia matrix $\boldsymbol{M}(\boldsymbol{q})$ is given by

$$\boldsymbol{M}(\boldsymbol{q}) = \begin{bmatrix} (m_1+m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos(q_2) & m_2 l_2^2 + m_2 l_1 l_2 \cos(q_2) \\ m_2 l_2^2 + m_2 l_1 l_2 \cos(q_2) & m_2 l_2^2 \end{bmatrix},$$

where q_1 and q_2 are two joint configurations. We solve the Riemannian Eikonal equation $\|\nabla U(\boldsymbol{q})\|_{\boldsymbol{M}^{-1}(\boldsymbol{q})} = 1$ with source point $\boldsymbol{q}_s$ as the boundary. The solution U is the minimal geodesic length on the manifold corresponding to the minimal kinetic energy path defined by $\boldsymbol{M}$. The vector field of gradient flow is written as $V(\boldsymbol{q}) = \boldsymbol{M}^{-1}(\boldsymbol{q})\nabla U(\boldsymbol{q})$, which has unit norm with respect to the kinetic energy metric

$$\begin{aligned} \|V(\boldsymbol{q})\|^2_{\boldsymbol{M}(\boldsymbol{q})} &= V^\top(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})V(\boldsymbol{q}) \\ &= \nabla U^\top(\boldsymbol{q})\boldsymbol{M}^{-1}(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})\boldsymbol{M}^{-1}(\boldsymbol{q})\nabla U(\boldsymbol{q}) \\ &= \nabla U^\top(\boldsymbol{q})\boldsymbol{M}^{-1}(\boldsymbol{q})\nabla U(\boldsymbol{q}) \\ &= \|\nabla U(\boldsymbol{q})\|^2_{\boldsymbol{M}^{-1}(\boldsymbol{q})} = 1. \end{aligned} \tag{4}$$

We use the method described in [17] for implementation that uses Voronoi reduction to handle Riemannian metrics with strong anisotropy. Fig. 1 (b) and (c) are the solution of the energy field and geodesic flow, with source point $\boldsymbol{q}_s = -(0.5, 0.8)$. It shows that minimal kinetic energy trajectories follow geodesics on the manifold instead of straight lines. (d) visualizes the manifold in 3D and shows different robot motions between a geodesic path and an Euclidean path. A comparison of geodesic length is shown in Tab I.

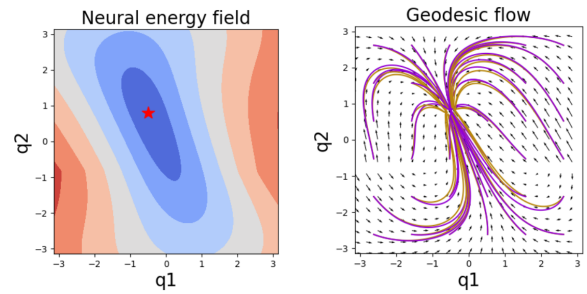
It is worth noting that the formulation of the eikonal equation can be naturally used to handle motion planning problems in terms of collision avoidance, by assigning zero velocity for points inside the obstacle, corresponding to infinite travel time. We showcase an example in Fig. 2.

B. Neural Riemannian Eikonal Solver

The geodesic length $U(\boldsymbol{q}, \boldsymbol{q}_s)$ from source joint configuration $\boldsymbol{q}_s$ to goal $\boldsymbol{q}$ on the Riemannian manifold can be approximated through a neural network. The idea is to calculate

	Euclidean Path	Geodesic Path	
		FMM	NES
2D planar robot	7.35	**6.38**	6.44
7D Franka robot	3.76	-	**3.30**

TABLE I: Path lengths on the Riemannian manifold defined by kinetic energy metric.



(a) (b)

Fig. 3: Solution of neural Riemannian eikonal solver. (a) shows the energy field and geodesic flow at the same source point as 1(b)(c). (b) highlights the difference in the geodesic flows between the solution from NES (Gold) and FMM (Violet).

the gradient $\nabla_{\boldsymbol{q}} U(\boldsymbol{q}, \boldsymbol{q}_s)$ and make it satisfy the Riemannian eikonal equation $\|\nabla U(\boldsymbol{q}, \boldsymbol{q}_s)\|_{\boldsymbol{M}^{-1}(\boldsymbol{q})} = 1$. Following previous work described in [12], we reformulate the geodesic as

$$U_{\boldsymbol{\theta}}(\boldsymbol{q}, \boldsymbol{q}_s) = \|\boldsymbol{q} - \boldsymbol{q}_s\|(\sigma \circ u_{\boldsymbol{\theta}}(\boldsymbol{q}, \boldsymbol{q}_s)), \tag{5}$$

where $u_{\boldsymbol{\theta}}(\boldsymbol{q}, \boldsymbol{q}_s)$ is the output from the neural network parameterized by $\boldsymbol{\theta}$. σ is a non-negative strictly increasing function and $\circ$ is function composition. This equation ensures the energy at the source point $U_{\boldsymbol{\theta}}(\boldsymbol{q}_s, \boldsymbol{q}_s) = 0$, mitigating singularity issues for points near the source point [23, 12, 8]. We used a Softplus function $\sigma(\boldsymbol{x}) = \frac{1}{\beta}(1 + \exp(\beta \boldsymbol{x}))$ after the neural network output.

Unlike FMM, which fixes the source position, the neural network solver can handle any point pairs within the domain. To guarantee the symmetry of the geodesic distance for permuted pairs, we require $U_{\boldsymbol{\theta}}(\boldsymbol{q}_1, \boldsymbol{q}_2) = U_{\boldsymbol{\theta}}(\boldsymbol{q}_2, \boldsymbol{q}_1)$ and to maintain the equity of the gradient with respect to the same configuration, we require $\nabla_{\boldsymbol{q}_1} U_{\boldsymbol{\theta}}(\boldsymbol{q}_1, \boldsymbol{q}_2) = \nabla_{\boldsymbol{q}_1} U_{\boldsymbol{\theta}}(\boldsymbol{q}_2, \boldsymbol{q}_1)$. The factorized eikonal equation 5 is further extended for the global estimation as suggested in [8], namely

$$U_{\boldsymbol{\theta}}(\boldsymbol{q}_1, \boldsymbol{q}_2) = \|\boldsymbol{q}_1 - \boldsymbol{q}_2\| \frac{\sigma \circ u_{\boldsymbol{\theta}}(\boldsymbol{q}_1, \boldsymbol{q}_2) + \sigma \circ u_{\boldsymbol{\theta}}(\boldsymbol{q}_2, \boldsymbol{q}_1)}{2}. \tag{6}$$

We employ automatic differentiation for calculating the gradient of the parameterized network output with respect to the input $\nabla_{\boldsymbol{q}_1} U_{\boldsymbol{\theta}}(\boldsymbol{q}_1, \boldsymbol{q}_2)$. This gradient is utilized to construct the loss function, ensuring the restriction of the gradient as shown in (4), namely

$$L(\boldsymbol{q}_1, \boldsymbol{q}_2) = \sqrt{\sum_{i=1}^{n} \left(\|\nabla_{\boldsymbol{q}_1^i} U_{\boldsymbol{\theta}}(\boldsymbol{q}_1^i, \boldsymbol{q}_2^i)\|_{\boldsymbol{M}^{-1}(\boldsymbol{q}_1^i)} - 1 \right)^2}, \tag{7}$$

where i is the i-th data and n is the batchsize. To improve the performance of the neural eikonal solver, we exploit a geometry-aware method to sample training data on the Riemannian manifold.

C. Geometry-Aware Sampling

Sampling points in Euclidean space randomly or uniformly does not exploit the local geometric structure of a manifold. Consequently, the solution from the eikonal equation can deviate from the true geodesics [12, 6]. The challenge lies

in incorporating knowledge of the geometric structure during the sampling process.

1) Target probabilistic density function: At each tangent space, an infinitesimal space is induced by the Riemannian metric $d\mathcal{M}(\boldsymbol{q}) = \sqrt{|\boldsymbol{M}(\boldsymbol{q})|}d\boldsymbol{q}$, bridging the target probabilistic density function (pdf) $\rho(\boldsymbol{q})$ with respect the Lesbergue measurement $d\boldsymbol{q}$, to the pdf $p(\boldsymbol{q})$ with respect to $d\mathcal{M}(\boldsymbol{q})$ by

$$\rho(\boldsymbol{q}) = p(\boldsymbol{q})\sqrt{|\boldsymbol{M}(\boldsymbol{q})|}. \tag{8}$$

2) Sampling on the Riemannian manifold: The objective is to sample variables on the Riemannian manifold from the pdf $\rho(\boldsymbol{q})$, while taking into account the local geometric structure. Given the Riemannian metric tensor, we adopt the Metropolis adjusted Langevin Monte Carlo algorithm on the Riemannian Manifold [7]. The algorithm describes the Langevin diffusion process on the Riemannian manifold in a stochastic differential equation (SDE)

$$d\boldsymbol{q}(t) = \frac{1}{2}\tilde{\nabla}_{\boldsymbol{q}}\mathcal{L}(\boldsymbol{q}(t)) + d\tilde{\mathbf{b}}(t), \tag{9}$$

with $\tilde{\nabla}_{\boldsymbol{q}}\mathcal{L}(\boldsymbol{q}(t))$ representing the natural gradient equipped by the Riemannian metric tensor $\tilde{\nabla}_{\boldsymbol{q}}\mathcal{L}(\boldsymbol{q}) = \boldsymbol{M}^{-1}(\boldsymbol{q})\ \nabla_{\boldsymbol{q}}\mathcal{L}(\boldsymbol{q})$, where $\mathcal{L}(\boldsymbol{q})$ is the logarithm of the desired pdf $\rho(\boldsymbol{q})$. Assuming $p(\boldsymbol{q})$ is a constant, the natural gradient is expressed using (8):

$$\tilde{\nabla}_{\boldsymbol{q}}\mathcal{L}(\boldsymbol{q}) = \boldsymbol{M}^{-1}(\boldsymbol{q})\nabla_{\boldsymbol{q}}\sqrt{|\boldsymbol{M}(\boldsymbol{q})|}. \tag{10}$$

In (9), $d\tilde{\mathbf{b}}(t)$ defines the Brownian motion on the Riemannian manifold as

$$\begin{aligned} d\tilde{\mathbf{b}}_i(t) = &\frac{1}{\sqrt{|\boldsymbol{M}(\boldsymbol{q}(t))|}}\sum_{j=1}^{D}\frac{\partial}{\partial \boldsymbol{q}_j}\left((\boldsymbol{M}^{-1}(\boldsymbol{q}(t)))_{ij}\sqrt{|\boldsymbol{M}(\boldsymbol{q}(t))|}\right)dt \\ &+\left(\sqrt{\boldsymbol{M}^{-1}(\boldsymbol{q}(t))}d\mathbf{b}(t)\right)_i. \end{aligned} \tag{11}$$

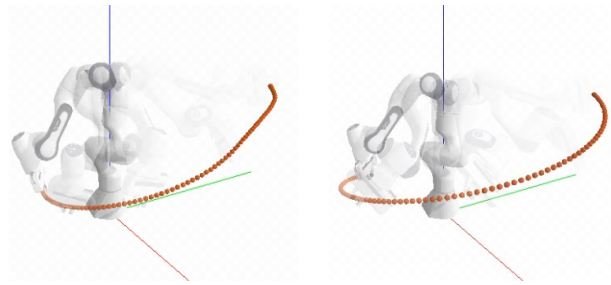
After applying the first-order Euler integration with the fixed step size ϵ to the SDE (9) as

$$\begin{aligned} \boldsymbol{q}_i(t+1) = &\boldsymbol{q}_i(t) + \frac{\epsilon^2}{2}\left((\boldsymbol{M}^{-1}(\boldsymbol{q}(t))\nabla_{\boldsymbol{q}}\mathcal{L}(\boldsymbol{q}(t)))_i\right) \\ &- \epsilon^2\sum_{j=1}^{D}\left((\boldsymbol{M}^{-1}(\boldsymbol{q}(t)))_{ij}\frac{\partial \boldsymbol{M}(\boldsymbol{q}(t))}{\partial \boldsymbol{q}_j}\boldsymbol{M}^{-1}(\boldsymbol{q}(t))\right)_{ij} \\ &+ \frac{\epsilon^2}{2}\sum_{j=1}^{D}(\boldsymbol{M}^{-1}(\boldsymbol{q}(t)))_{ij}\,\mathrm{Tr}\left((\boldsymbol{M}^{-1}(\boldsymbol{q}(t)))\frac{\partial G(\boldsymbol{q}(t))}{\partial \boldsymbol{q}_j}\right) \\ &+ (\epsilon\sqrt{\boldsymbol{M}^{-1}(\boldsymbol{q}(t))}\boldsymbol{z}(t))_i \\ = &\boldsymbol{\mu}(\boldsymbol{q}(t),\epsilon)_i + \left(\epsilon\sqrt{\boldsymbol{M}^{-1}(\boldsymbol{q}(t))}\boldsymbol{z}(t)\right)_i, \end{aligned} \tag{12}$$

the random variable can be sampled from the Gaussian distribution

$$p\Big(\boldsymbol{q}(t+1)|\boldsymbol{q}(t)\Big) = \mathcal{N}\Big(\boldsymbol{q}(t+1)|\boldsymbol{\mu}(\boldsymbol{q}(t),\epsilon),\ \epsilon^2\boldsymbol{M}^{-1}\big(\boldsymbol{q}(t)\big)\Big). \tag{13}$$

The acceptance of the sampled variable is finally calculated with $\min\left\{1, \frac{p(\boldsymbol{q}(t+1))p(\boldsymbol{q}(t)|\boldsymbol{q}(t+1))}{p(\boldsymbol{q}(t))p(\boldsymbol{q}(t+1)|\boldsymbol{q}(t))}\right\}$.



(a) (b)

Fig. 4: Geodesic (a) and Euclidean (b) motions of 7D Franka robot.

D. Results

The network is parameterized using a fully connected multilayer perception (MLP) trained with the Adam optimizer. We first compare the result of the neural eikonal solver with FMM on the 2D robot, demonstrated in Fig. 3. Quantitive results are reported in Table I. Geodesic motions lead to a short path length on the manifold. We show the geodesic and Euclidean motions of a 7-axis Franka robot in Fig. 4.

IV. Discussion & Future Work

In this paper, we propose to solve the Riemannian eikonal equation for energy-aware motion generation on a configuration space manifold. Unlike other geometric methods that optimize geodesic paths on the manifold [10, 13], our approach is physics-informed, where a unit norm constrains joint velocities on the manifold. The proposed neural eikonal solver is promising due to its efficiency in terms of online inference, as well as its generalizability to any start-goal pairs in the configuration space. As a local policy, the gradient flow can be integrated into other frameworks such as Riemannian motion policies [21] to handle energy efficiency. There are several possible extensions for future work:

Energy-aware inverse kinematics. In this work, we considered both source and goal points in configuration space. However, the energy term could be pulled back to task space by using the pseudoinverse of Jacobian matrix $\dot{\boldsymbol{q}}^\top\boldsymbol{M}\dot{\boldsymbol{q}} = \dot{\boldsymbol{x}}^\top J^\dagger\boldsymbol{M}(J^\dagger)^\top\dot{\boldsymbol{x}}$. Therefore, geodesics on the manifold defined by the metric $J^\dagger\boldsymbol{M}(J^\dagger)^\top$ corresponds to the minimal-energy inverse kinematics solution.

General dynamics-aware motion generation and manipulation tasks. The kinetic energy metric on Riemannian manifold is a special case of equations of motion. Other metrics and geometries could also be considered to achieve more general dynamics-aware motion planning. Besides, the eikonal equation is also a special case of the Hamilton-Jacobi-Bellman (HJB) equation and the neural PDE solvers could be extended to general manipulation tasks exploiting differentiable equations.

Introducing geometric features into data. Fourier features have been demonstrated to enhance the solver accuracy [24]. However, Fourier features are the eigenfunctions of Euclidean space and do not adequately represent the eigenfunctions for a general manifold. Future work could involve learning the eigenfunctions on the high-dimensional manifold and integrating them with the neural solver framework.

References

[1] Alin Albu-Schäffer and Arne Sachtler. What can algebraic topology and differential geometry teach us about intrinsic dynamics and global behavior of robots? In *The International Symposium of Robotics Research*, pages 468–484. Springer, 2022.

[2] Francesco Bullo and Andrew D Lewis. *Geometric Control of Mechanical Systems: Modeling, Analysis, and Design for Simple Mechanical Control Systems*, volume 49. Springer Science & Business Media, 2004.

[3] Tan Fung Chan and Rajiv V Dubey. A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators. *IEEE transactions on Robotics and Automation*, 11(2):286–292, 1995.

[4] Pasquale Chiacchio, Stefano Chiaverini, Lorenzo Sciavicco, and Bruno Siciliano. Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy. *The International Journal of Robotics Research*, 10(4):410–425, 1991.

[5] Keenan Crane, Clarisse Weischedel, and Max Wardetzky. Geodesics in heat: A new approach to computing distance based on heat flow. *ACM Transactions on Graphics (TOG)*, 32(5):1–11, 2013.

[6] Manfredo Perdigao Do Carmo and J Flaherty Francis. *Riemannian geometry*, volume 2. Springer, 1992.

[7] Mark Girolami and Ben Calderhead. Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 73(2):123–214, 2011.

[8] Serafim Grubas, Anton Duchkov, and Georgy Loginov. Neural eikonal solver: Improving accuracy of physics-informed neural networks for solving eikonal equation in case of caustics. *Journal of Computational Physics*, 474:111789, 2023.

[9] Julius Jankowski, Lara Brudermüller, Nick Hawes, and Sylvain Calinon. VP-STO: Via-point-based stochastic trajectory optimization for reactive robot behavior. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10125–10131. IEEE, 2023.

[10] Noémie Jaquier and Tamim Asfour. Riemannian geometry as a unifying theory for robot motion learning and control. In *The International Symposium of Robotics Research*, pages 395–403. Springer, 2022.

[11] Mrinal Kalakrishnan, Sachin Chitta, Evangelos Theodorou, Peter Pastor, and Stefan Schaal. Stomp: Stochastic trajectory optimization for motion planning. In *2011 IEEE international conference on robotics and automation*, pages 4569–4574. IEEE, 2011.

[12] Daniel Kelshaw and Luca Magri. Computing distances and means on manifolds with a metric-constrained eikonal approach. *arXiv preprint arXiv:2404.08754*, 2024.

[13] Holger Klein, Noémie Jaquier, Andre Meixner, and Tamim Asfour. On the design of region-avoiding metrics for collision-safe motion generation on riemannian manifolds. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2346–2353. IEEE, 2023.

[14] Yiming Li, Xuemin Chi, Amirreza Razmjoo, and Sylvain Calinon. Configuration space distance fields for manipulation planning. In *Proc. Robotics: Science and Systems (RSS)*, 2024.

[15] Yiming Li, Yan Zhang, Amirreza Razmjoo, and Sylvain Calinon. Representing robot geometry as distance fields: Applications to whole-body manipulation. In *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, pages 15351–15357, 2024.

[16] Jean-Marie Mirebeau. Anisotropic fast-marching on cartesian grids using voronoi's first reduction of quadratic forms. *HAL (Preprint), April*, 63, 2017.

[17] Jean-Marie Mirebeau and Jorg Portegies. Hamiltonian fast marching: a numerical solver for anisotropic and non-holonomic eikonal pdes. *Image Processing On Line*, 9:47–93, 2019.

[18] Ruiqi Ni and Ahmed H Qureshi. Ntfields: Neural time fields for physics-informed robot motion planning. In *The Eleventh International Conference on Learning Representations*, 2022.

[19] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[20] Nathan Ratliff, Matt Zucker, J Andrew Bagnell, and Siddhartha Srinivasa. Chomp: Gradient optimization techniques for efficient motion planning. In *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, pages 489–494. IEEE, 2009.

[21] Nathan D Ratliff, Jan Issac, Daniel Kappler, Stan Birchfield, and Dieter Fox. Riemannian motion policies. *arXiv preprint arXiv:1801.02854*, 2018.

[22] James A Sethian. A fast marching level set method for monotonically advancing fronts. *proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

[23] Jonathan D Smith, Kamyar Azizzadenesheli, and Zachary E Ross. Eikonet: Solving the eikonal equation with deep neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 59(12):10685–10696, 2020.

[24] Sifan Wang, Hanwen Wang, and Paris Perdikaris. On the eigenvector bias of fourier feature networks: From regression to solving multi-scale pdes with physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 384:113938, 2021.