Geometry-aware Bayesian Optimization in Robotics using Riemannian Matérn Kernels

Noémie Jaquier[*1] **Viacheslav Borovitskiy**[*2,6] **Andrei Smolensky**[2]
Alexander Terenin[3,4] **Tamim Asfour**[1] **Leonel Rozo**[5]
[1]Karlsruhe Institute of Technology [2]St. Petersburg State University
[3]University of Cambridge [4]Imperial College London [5]Bosch Center for Artificial Intelligence
[6]St. Petersburg Department of Steklov Mathematical Institute of Russian Academy of Sciences

Abstract: Bayesian optimization is a data-efficient technique which can be used for control parameter tuning, parametric policy adaptation, and structure design in robotics. Many of these problems require optimization of functions defined on non-Euclidean domains like spheres, rotation groups, or spaces of positive-definite matrices. To do so, one must place a Gaussian process prior, or equivalently define a kernel, on the space of interest. Effective kernels typically reflect the geometry of the spaces they are defined on, but designing them is generally non-trivial. Recent work on the Riemannian Matérn kernels, based on stochastic partial differential equations and spectral theory of the Laplace–Beltrami operator, offers promising avenues towards constructing such geometry-aware kernels. In this paper, we study techniques for implementing these kernels on manifolds of interest in robotics, demonstrate their performance on a set of artificial benchmark functions, and illustrate geometry-aware Bayesian optimization for a variety of robotic applications, covering orientation control, manipulability optimization, and motion planning, while showing its improved performance.

Keywords: Bayesian optimization, Matérn kernels, Riemannian manifolds

1 Introduction

Fast and data-efficient adaptation is a key challenge in a large range of robotics applications, owing to the need to handle sensor noise, model uncertainty, and generalize to unforeseen settings. In this context, Bayesian optimization [46] is a technique of growing interest in robotics due to its ability to efficiently solve challenging optimization problems, including controller tuning [5, 33], policy adaptation [4, 15], and robot design [42]. This makes it an attractive tool for settings where data-efficiency is of key interest, and large-scale methods such as deep reinforcement learning are not presently viable. Bayesian optimization algorithms are generally powered by probabilistic models based on Gaussian processes that rely on inductive biases encoded by kernels. Improving performance and scalability of such models via a number of different strategies has therefore been a central goal pursued by the Bayesian optimization and Gaussian process literatures.

Inductive bias in robotics can be introduced by exploiting domain knowledge about the task at hand. For example, task-specific kernels were proposed by Antonova et al. [5] by learning a distance metric via simulated bipedal locomotion patterns to optimize gait controllers, and by Rai et al. [38] using gait feature transformations to design kernels to optimize locomotion controllers. More generic kernels were introduced by Marco et al. [33], Marco et al. [34], and Wilson et al. [53] for policy optimization by incorporating linear-quadratic control structures and trajectory information into the kernel design. While these kernels can be exploited for robot control or policy search with many different tasks and systems, their use is specific to the problems they were designed for.

A more general approach is to carefully select the *domain* of the function being optimized, thus incorporating information about the geometry of the search space into the optimization algorithm. Many quantities of interest in robotics carry such geometric information: three-dimensional rotations

[*]Equal contribution. Correspondence to: noemie.jaquier@kit.edu, viacheslav.borovitskiy@gmail.com.
Code at https://github.com/NoemieJaquier/MaternGaBO and video at https://youtu.be/6awfFRqP7wA.

5th Conference on Robot Learning (CoRL 2021), London, UK.

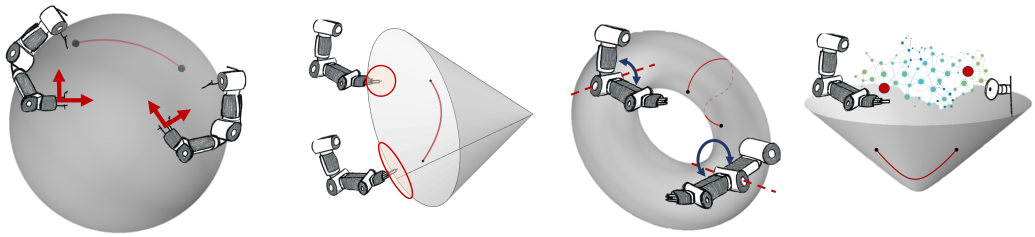

Figure 1: Illustration of different Riemannian manifolds of interest in robotics: a unit sphere $\mathbb{S}^d$, symmetric positive definite matrices $\mathcal{S}^d_{++}$, a torus $\mathbb{T}^d$, and the Lorentz model of hyperbolic geometry in $\mathcal{H}^2$ (from left to right). Red solid lines depict geodesics between two (black) points on the manifold.

can be viewed as elements of the Lie group $\mathrm{SO}(3)$ [20] or the unit sphere $\mathbb{S}^3$ [51], control gains, inertia and manipulability ellipsoids lie in the manifold of symmetric-positive-definite matrices $\mathcal{S}^d_{++}$, while the joint configuration of a d-degree-of-freedom robot with revolute joints may be viewed as a point on a torus $\mathbb{T}^d$, i.e. the product of d circles [28], see Figure 1 for illustrated examples. Incorporating the geometric structure of these spaces into Bayesian optimization has recently been shown to improve its performance on a number of important tasks [27], despite the use of a naïve geometrical approximation in the kernel design.

Geometry-aware Bayesian optimization is therefore a promising emerging tool for general use in robotics. One of the main difficulties in developing this tool is that defining geometry-aware Gaussian process models is technically non-trivial: classic Euclidean methods ignore the parameter space geometry, and naïve geometric approximations do not capture the manifold's geometric structure. Fortunately, mathematical theory and techniques for building the necessary kernels are beginning to emerge [8, 30]. In particular, the work of Borovitskiy et al. [8] on Riemannian Matérn kernels provides mathematically-sound tools for calculating geometry-aware kernels by building on stochastic partial differential equations and spectral theory of the Laplace–Beltrami operator. This provides the key ingredients to deploy geometry-aware Bayesian optimization on practical robotics problems.

Building on the performance improvements in Jaquier et al. [27] and the theoretically-grounded Matérn kernels proposed in Borovitskiy et al. [8], we present a more general kernel formulation for a wide range of Riemannian manifolds of interest in robotics. Our approach uses the fact that a Matérn kernel can be formulated as the integral of a squared-exponential kernel: this idea enables us to define Riemannian Matérn kernel on non-compact manifolds, generalizing the work of Borovitskiy et al. [8] which applies only to compact manifolds. We introduce new Riemanian Matérn kernels for the special orthogonal group, manifold of symmetric positive definite matrices, and hyperbolic space, and study their use in robotic applications. This more general formulation avoids the need to determine valid length scales as in prior work [27], provides theoretical guarantees of positive-definiteness, and makes kernel smoothness optimization possible. We benchmark these kernels on a set of artificial functions and on a number of robotics problems of interest to practitioners.

2 Geometry-aware Bayesian Optimization

Bayesian optimization [45] is a class of sequential search algorithms for finding a global minimizer $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x})$ of an unknown objective function f defined over a domain of interest $\mathcal{X}$. The function f is not observed directly: instead, at each step the algorithm must select a query point $\boldsymbol{x} \in \mathcal{X}$ at which to evaluate f. Moreover, it is usually assumed that the algorithm only observes a noisy value $y \in \mathbb{R}$ whose expectation $\mathbb{E}(y) = f(\boldsymbol{x})$ is the true function value at the chosen point. Based on the full set of observed evaluations, the algorithm must select the next query point $\boldsymbol{x}$.

The vast majority of Bayesian optimization algorithms employ Gaussian process models, which represent the known information about the objective function using a Gaussian process prior $f \sim \mathrm{GP}(\mu, k)$, with mean function $\mu : \mathcal{X} \to \mathbb{R}$ and symmetric-positive-definite kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. The Gaussian process is updated by conditioning on observed data $(\boldsymbol{x}_i, y_i)_{i=1}^n$ using Bayes' Rule. The next query point is selected by optimizing a carefully-constructed acquisition function defined with respect to the resulting posterior distribution.

In robotics, the domain of interest $\mathcal{X}$ often possesses important non-Euclidean geometric structure. For example, when tuning robot controllers, the variables of interest include end-effector poses

lying on $\mathbb{R}^3 \times \mathbb{S}^3$ where $\mathbb{S}^d$ denotes the d-dimensional unit sphere, robot joint postures defined on a torus $\mathbb{T}^d = \mathbb{S}^1 \times \ldots \times \mathbb{S}^1$, as well as gain matrices and manipulability ellipsoids belonging to the space of symmetric-positive-definite matrices $\mathcal{S}^d_{++}$. In addition, when adapting parametric robot policies, some of their parameters such as mean vectors or covariance matrices lie in Riemannian manifolds. This inherent geometry is often overlooked in robotic applications [33, 40, 41]. In this paper, we demonstrate that incorporating the correct geometric structure of $\mathcal{X}$ into the Gaussian process is an important potential avenue for improving performance of Bayesian optimization.

Performing geometry-aware Bayesian optimization on Riemannian manifolds requires one to address two main challenges. Firstly, defining a valid Gaussian process prior is more difficult than in the Euclidean case. Secondly, numerical techniques such as training and optimizing the acquisition functions may require special considerations. We address these issues in the remainder of this section.

2.1 Gaussian Processes on Riemannian Manifolds

The key challenge in constructing Gaussian processes on non-Euclidean domains is in defining a valid kernel: seemingly obvious and innocuous-looking candidate expressions such as the squared exponential geodesic distance $\exp(-d_g(\boldsymbol{x}, \boldsymbol{x}')^2/\kappa)$ are not valid kernels for all length scales $\kappa > 0$ simultaneously [14]. In spite of this, they have been shown to be beneficial for practical robotics applications [27]: this motivates one to develop technically sound analogues. To tackle these issues, recent works on non-Euclidean Gaussian processes, including Lindgren et al. [30] and Borovitskiy et al. [7, 8], have developed provably well-posed techniques based on stochastic partial differential equations and spectral theory of the Laplace–Beltrami operator. We focus on the latter work as it does not require one to solve stochastic partial differential equations numerically.

The general theory of Matérn Gaussian processes on compact Riemannian manifolds from Borovitskiy et al. [8] introduces expressions for the kernels of these processes in terms of the Laplace–Beltrami eigenpairs of their respective manifolds. For more general settings including non-compact manifolds, such as the manifold of positive definite matrices and hyperbolic space, kernels can often be built by exploiting the connection between Riemannian squared exponential and heat kernels as fundamental solutions of heat equations. These can be connected with Matérn kernels by viewing them as certain integrals of squared exponential kernels [50]. We use this connection to introduce new Riemannian kernels on non-compact manifolds, and to further explore existing Riemannian kernels in Section 3.

2.2 Optimization of acquisition functions on manifolds

To use these Gaussian processes in a Bayesian optimization system, one must further be able to determine where to evaluate the unknown function and collect additional data. To do so, Bayesian optimization minimizes an acquisition function, which resolves the *explore-exploit tradeoff* [45] inherent in the problem due to lack of complete information about f. This is done, for instance, by prioritizing points which are known to perform well due to having high posterior mean, while also considering points whose performance is unknown due to high posterior variance. In this way, the probabilistic uncertainty given by the posterior Gaussian process is used to resolve the tradeoff.

In the geometric setting, Bayesian optimization therefore requires minimization of real-valued functions on manifold domains as an intermediate step. To do so, we leverage optimization algorithms on Riemannian manifolds [1] to maximize the acquisition function in a geometry-aware manner. These algorithms work with the Euclidean tangent space $\mathcal{T}_{\boldsymbol{x}}\mathcal{X}$ linked to each point $\boldsymbol{x}$ on the manifold $\mathcal{X}$, utilize gradients intrinsically defined as elements of the tangent space, and project the results of their computations back onto the manifold with the exponential map $\mathrm{Exp}_{\boldsymbol{x}} : \mathcal{T}_{\boldsymbol{x}}\mathcal{X} \to \mathcal{X}$. These different operations are determined based on the Riemannian metric with which the manifold is endowed.

In this paper, we employ trust region methods on Riemannian manifolds, as introduced by Absil et al. [2], to optimize the acquisition function at each iteration of Bayesian optimization. The recursive process used in trust region methods on Riemannian manifolds involves the same steps as its Euclidean counterpart, namely: (i) optimization of a quadratic subproblem m_k trusted locally in a region around the iterate, (ii) updating of the trust region parameters, typically the trust-region radius Δ_k, (iii) updating the iterates, where a candidate is accepted or rejected as function of the quality of the model m_k. For the cases where the space of interest is restricted to a subset of a Riemannian manifold, we employ a constrained trust region algorithm proposed by Jaquier and Rozo [25] that handles linear constraints. This recipe applies to almost all acquisition functions, including stochastic

ones via pathwise conditioning [54, 55], and allows us to introduce physical or safety constraints that naturally arise when interacting with physical systems and are crucial for robotics applications.

3 Matérn Kernels on Riemannian Manifolds

Training Gaussian processes, a crucial component of Bayesian optimization, requires us to be able to compute the kernel point-wise. This kernel encapsulates similarity information in the domain of interest: for domains with geometric structure, such as Riemannian manifolds, the kernel should reflect this structure in order to perform effectively. We now introduce geometric kernels that capture the structure of a number of spaces of interest in robotics, and study techniques for computing them.

As aforementioned, we focus on Riemannian Matérn kernels. These are based on the characterization of the widely used Euclidean Matérn Gaussian processes as solutions of a class of stochastic partial differential equations given by Whittle [52]. These equations naturally generalize to the Riemannian setting, and yield kernels which are well-defined for all Riemannian manifolds and hyperparameter values, unlike prior work [27] where restricting length scale range is required to guarantee positive definiteness. The main difficulty of this approach is that this definition is implicit and does not give practical expressions for computing the kernel without further considerations. For compact Riemannian manifolds, Borovitskiy et al. [8] prove expressions for Matérn kernels of the form

$$k_{\nu,\kappa,\sigma^2}(\boldsymbol{x},\boldsymbol{x}') = \frac{\sigma^2}{C_\nu}\sum_{n=0}^{\infty}\Phi_\nu(\lambda_n)f_n(\boldsymbol{x})f_n(\boldsymbol{x}') \qquad \Phi_\nu(\lambda_n) = \begin{cases}\left(\frac{2\nu}{\kappa^2}+\lambda_n\right)^{-\nu-\frac{d}{2}} & \nu<\infty,\\ e^{-\frac{\kappa^2}{2}\lambda_n} & \nu=\infty,\end{cases} \tag{1}$$

where (λ_n, f_n) are Laplace–Beltrami eigenpairs ($\lambda_n \geq 0$), κ is the length scale, σ^2 is the variance, ν is the smoothness and the constant C_ν is chosen so that the resulting kernels have average variance across the domain equal to σ^2. Note that despite notation C_ν depends on both ν and κ. The limiting kernel as $\nu \to \infty$, which we denote by $k_{\infty,\kappa,\sigma^2}$, can be viewed as the Riemannian analog of the Euclidean squared exponential kernel.

In practice, the kernel (1) and its squared exponential limit are computed by calculating the eigenpairs either analytically or numerically, and then truncating the infinite sum. Numerical computation of the eigenpairs amounts to approximating the manifold with a mesh and solving the eigenproblem using finite element methods [31]. Since this approach scales exponentially with manifold dimension, we do not focus on it in this work. Fortunately, owing to their extensive use in physics and other areas, mathematical techniques for evaluating the eigenpairs and simplifying the related expressions are available for many manifolds of interest in robotics: examples of this can be seen in the sequel. Although we only present the formulas for squared exponential (heat) kernels for the sake brevity, respective formulas for Matérn kernels hold as well.

The expression (1) does not apply for non-compact manifolds, although the implicit stochastic partial differential equation definition it is derived from is still valid. In this case, we leverage representations of Matérn kernels as integrals of squared exponential kernels, also known as heat kernels in the mathematical literature. In the Euclidean case, observe that

$$k_{\nu,\kappa,\sigma^2}(\boldsymbol{x},\boldsymbol{x}') = C\int_0^\infty u^{\nu-1}e^{-\frac{2\nu}{\kappa^2}u}k_{\infty,\sqrt{2u},\sigma^2}(\boldsymbol{x},\boldsymbol{x}')\,\mathrm{d}u, \tag{2}$$

where C is given in (20). This can be obtained by writing the spectral measure of the Matérn kernel, the T distribution, as a gamma mixture of Gaussians. A similar relationship also holds in the compact Riemannian case: see Appendix B for details. More generally, it can be considered as the *definition* of a Matérn kernel in any setting where a suitable notion of a squared exponential or heat kernel is available (in this case we usually use $C = 1$), which is particularly useful in non-compact settings. Moreover, as shown in Appendix B, this kernel is positive definite whenever the respective heat kernel is positive definite. Therefore, it suffices to obtain expressions for the squared exponential kernel: analogous expressions for the Matérn kernel follow by integrating over length scales via (2). We thus focus on the heat kernel henceforth.[1]

These general expressions for Riemannian Matérn kernels allow us to build theoretically-sound kernels for the torus and the sphere. More importantly, we can introduce Riemannian formulations

[1] Note that it is usually beneficial to exploit explicit expressions for the Matérn kernels when they are available, e.g., (1), to alleviate additional numerical errors introduced by approximately computing the integral (2).

for kernels on non-compact manifolds such as the manifold of symmetric positive definite matrices, and $\mathcal{H}^d$. We now turn our attention to manifolds of key importance in robotics applications.

Torus. The torus $\mathbb{T}^d$ is defined as a product of circles $\mathbb{T}^d = \mathbb{S}^1 \times \ldots \times \mathbb{S}^1$, which naturally arises as, for example, the space of joint configuration of a d-degree-of-freedom robot. Functions on a torus $f : \mathbb{T}^d \to \mathbb{R}$ can be viewed as 1-periodic functions $f : \mathbb{R}^d \to \mathbb{R}$. Using this identification, the Laplace–Beltrami eigenfunctions on $\mathbb{T}^d$ are exactly those eigenfunctions of the Euclidean Laplacian which are 1-periodic: sines and cosines whose frequency is an integer multiple of 2π. Substituting this into the general formula yields

$$k_{\infty,\kappa,\sigma^2}(\boldsymbol{x}, \boldsymbol{x}') = \frac{\sigma^2}{C_\infty} \sum_{\boldsymbol{\tau} \in \mathbb{Z}^d} e^{-2\kappa^2\pi^2\|\boldsymbol{\tau}\|^2} \cos(2\pi\langle \boldsymbol{\tau}, \boldsymbol{x} - \boldsymbol{x}'\rangle), \tag{3}$$

where the sum is over vectors of integers $\boldsymbol{\tau}$, and $\boldsymbol{x}, \boldsymbol{x}' \in [0, 1)^d$ are the representation of elements of $\mathbb{T}^d$ as vectors of angles divided by 2π, i.e., such that $\left(e^{2\pi i x_1}, \ldots, e^{2\pi i x_d}\right)$ and $\left(e^{2\pi i x'_1}, \ldots, e^{2\pi i x'_d}\right)$ are the corresponding elements of $\mathbb{T}^d$. Here, C_∞ is the normalizing constant chosen so that $k(\boldsymbol{x}, \boldsymbol{x}) = \sigma^2$. The key difference with (1) is that the eigenpairs are now parameterized by vectors of integers, slightly simplifying the expression. A simple proof of this expression is given in Appendix B.

Sphere. The sphere manifold occurs in a number of settings: in particular, unit quaternions commonly used in robotics to represent the orientation of rigid bodies are a common example of the three-dimensional sphere $\mathbb{S}^3$. On the d-dimensional sphere $\mathbb{S}^d$ the squared exponential kernel is

$$k_{\infty,\kappa,\sigma^2}(\boldsymbol{x}, \boldsymbol{x}') = \frac{\sigma^2}{C_\infty} \sum_{n=0}^{\infty} c_{n,d}\, e^{-\frac{\kappa^2}{2} n(n+d-1)}\, \mathcal{C}_n^{(d-1)/2}\big(\cos(d_g(\boldsymbol{x}, \boldsymbol{x}'))\big), \tag{4}$$

where C_∞ is a normalizing constant which ensures $k_{\infty,\kappa,\sigma^2}(\boldsymbol{x}, \boldsymbol{x}) = \sigma^2$; $c_{n,d}$ are explicit constants, $\mathcal{C}_n^{(\cdot)}$ are the Gegenbauer polynomials, and d_g is the geodesic distance on the sphere [8]. This kernel is obtained by noting that the eigenfunctions of the spherical Laplace–Beltrami operator are the spherical harmonics, sums of which may be reformulated through the Gegenbauer polynomials—the resulting algebra, along with combining repeated eigenvalues, leads to the desired expression. This expression is a significant simplification compared to the general case: note in particular that the kernel factorizes and can be written as a function only of the geodesic distance d_g.

Special orthogonal group. The special orthogonal group $\mathrm{SO}(d)$ is a compact Riemannian manifold[2] of rotations of d-dimensional Euclidean space, most often represented as the set of orthogonal $d \times d$ matrices of determinant 1. $\mathrm{SO}(3)$ is widely used to represent orientations in the 3-dimensional physical space. This manifold may likewise be useful as a building block for representing other geometric objects. For instance, the product $\mathbb{R}^d \times \mathrm{SO}(d)$ may be used as a proxy to represent symmetric matrices via a non-unique decomposition $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{U}^\top$, where $\mathbf{D} = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$, $\lambda_i \in \mathbb{R}$ and $\mathbf{U} \in \mathrm{SO}(d)$. A kernel on $\mathbb{R}^d \times \mathrm{SO}(d)$ may be designed as a product of kernels on $\mathbb{R}^d$ and on $\mathrm{SO}(d)$. By restricting the Euclidean part of the product kernel to $\mathbb{R}^d_{>0}$ or $\mathbb{R}^d_{>0}$, we may obtain a kernel over positive (resp. semi)definite matrices, albeit not a canonical Matérn kernel.

By leveraging the group structure of $\mathrm{SO}(d)$ it is possible to express the corresponding heat kernel in terms of *group characters*. Specifically, we have that

$$k_{\infty,\kappa,\sigma^2}(\mathbf{X}, \mathbf{Y}) = \frac{\sigma^2}{C_\infty} \sum_{\pi} e^{-\frac{\kappa^2}{2}\lambda_\pi} d_\pi \chi_\pi(\mathbf{X}\mathbf{Y}^{-1}), \tag{5}$$

where the sum is over a certain set of tuples of nonnegative integers π called the *highest weights* and C_∞ is the normalizing constant ensuring that $k_{\infty,\kappa,\sigma^2}(\mathbf{X}, \mathbf{X}) = \sigma^2$: see Appendix B. Here, λ_π are Laplacian eigenvalues, χ_π are certain functions called *characters* that depend only on eigenvalues of their argument, and $d_\pi = \chi_\pi(\mathbf{I})$. Partial sums of (5) corresponding to the most significant eigenvalues may be computed in closed form by means of the Weyl character formula and Freudenthal's formula for the eigenvalues of the Casimir operator. We provide technical details concerning this formula and its practical implementation in Appendix B and in the supplemented code respectively, while deferring a thorough discussion to future work.

[2] $\mathrm{SO}(d)$ is a Lie group, meaning that it is a group endowed with a structure of differentiable manifold: this manifold admits a canonical Riemannian structure induced by the Killing form.

Hyperbolic space. The hyperbolic space $\mathcal{H}^d$ is the unique simply-connected complete d-dimensional Riemannian manifold with a constant negative sectional curvature -1. An important property of this space is the exponential rate of growth of the volume of a ball with respect to its radius. Because of this, it is often used to hold embeddings of hierarchical data, such as trees or graphs [24, 37]. Although its potential to embed discrete data structures into a continuous space is well-known in the machine learning community, its application in robotics is presently scarce. In Section 4, we provide a toy example featuring a simple path planning on a 2D grid to illustrate how the hyperbolic manifold can be exploited in robotics.

Hyperbolic space is non-compact, thus the technique of Borovitskiy et al. [8] is unable to represent this kernel. Fortunately, explicit formulas for the corresponding heat kernel in arbitrary dimension d are present in the literature. Take $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{H}^d$ and denote by $\rho = \operatorname{dist}_{\mathcal{H}^d}(\boldsymbol{x}, \boldsymbol{y})$ the geodesic distance between $\boldsymbol{x}$ and $\boldsymbol{y}$ in $\mathcal{H}(d)$. Then we have, as introduced in Grigoryan and Noguchi [19],

$$k^{\mathcal{H}(2)}_{\infty,\kappa,\sigma^2}(\boldsymbol{x}, \boldsymbol{y}) = \frac{\sigma^2}{C_\infty} \int_\rho^\infty \frac{s \exp\left(-s^2/(2\kappa^2)\right)}{(\cosh(s) - \cosh(\rho))^{1/2}} \,\mathrm{d}s, \quad k^{\mathcal{H}(3)}_{\infty,\kappa,\sigma^2}(\boldsymbol{x}, \boldsymbol{y}) = \frac{\sigma^2}{C_\infty} \frac{\rho}{\sinh \rho} e^{-\frac{\rho^2}{2\kappa^2}} \tag{6}$$

for $d = 2$ and $d = 3$, and the following recurrence relation, known as Millson's formula, for $d > 3$:

$$k^{\mathcal{H}(d)}_{\infty,\kappa,\sigma^2}(\boldsymbol{x}, \boldsymbol{y}) = -\frac{\sigma^2}{C_\infty \sinh \rho} \frac{\partial}{\partial \rho} k^{\mathcal{H}(d-2)}_{\infty,\kappa,\sigma^2}(\boldsymbol{x}, \boldsymbol{y}). \tag{7}$$

Manifold of positive definite matrices. Positive definite matrices occur naturally in many physics related problems. In robotics, they are used to represent the stiffness and manipulability ellipsoids [3, 57] or as control gain matrices [33]. These matrices strongly affect the performance of robot motion and force controllers, and they are variables of interest for task-compatible trajectory optimization.

Symmetric positive definite (SPD) matrices may be endowed with different Riemannian structures. For instance, $\mathbf{A} \mapsto \exp(\mathbf{A})$ and $\mathbf{A} \mapsto \mathbf{A}\mathbf{A}^\top$ define one-to-one maps between symmetric matrices or lower triangular matrices with positive elements on the diagonal and SPD matrices, allowing one to transfer Euclidean structure and kernels to SPD matrices. Though simple and natural, these Riemannian structures are often too simplistic and poorly capture the geometry of SPD matrices [13].

Another approach is to consider the map $\mathbf{A} \mapsto \mathbf{A}\mathbf{A}^\top$ as a function from the space $\mathrm{GL}(d)$ of invertible matrices to $\mathcal{S}^d_{++}$. This map is not one-to-one: specifically, $\mathbf{A}\mathbf{A}^\top = \mathbf{B}\mathbf{B}^\top$ if and only if $\mathbf{A} = \mathbf{B}\mathbf{Q}$ where $\mathbf{Q}$ is an orthogonal matrix: see Appendix B for details. To make this map one-to-one, we need to switch the domain to the quotient space of $\mathrm{GL}(d)$ by the equivalence relation of being equal up to an orthogonal matrix. The resulting space $\mathrm{GL}(d)/\,\mathrm{O}(d)$ is a Riemannian manifold. The Riemannian structure defined this way is not Euclidean and the resulting manifold is non-compact, hence the method of Borovitskiy et al. [8] does not apply here either. Still, explicit formulas for the corresponding heat kernels can be found in literature for cases of $d = 2$ and $d = 3$, which are common dimensions in robotics problems.

Specifically, for $d = 2$ there is a relatively simple formula. Take $\mathbf{X}, \mathbf{Y} \in \mathcal{S}^2_{++}$, denote by $H_1 \geq H_2$ the singular values of the matrix $\mathbf{X}\mathbf{Y}^{-1}$, and $\alpha = H_1 - H_2$, then, by Sawyer [43], we have

$$k_{\infty,\kappa,\sigma^2}(\mathbf{X}, \mathbf{Y}) = \frac{\sigma^2}{C_\infty} \exp\left(-\frac{H_1^2 + H_2^2}{2\kappa^2}\right) \int_0^\infty \frac{(2s + \alpha) \exp\left(-s(s+\alpha)/\kappa^2\right)}{(\sinh(s)\sinh(s+\alpha))^{1/2}} \,\mathrm{d}s, \tag{8}$$

where the one-dimensional integral on the right-hand side may be evaluated numerically and C_∞ is a normalizing constant ensuring that $k_{\infty,\kappa,\sigma^2}(\mathbf{X}, \mathbf{X}) = \sigma^2$. Additional details are given in Appendix B.

Product kernels on manifolds. If the Riemannian manifold $\mathcal{M}$ of interest is the product of d manifolds $\mathcal{M} = \mathcal{M}_1 \times \ldots \times \mathcal{M}_d$, where each factor $\mathcal{M}_j$ is equipped with a Matérn kernel $k^{\mathcal{M}_j}_{\nu,\kappa,\sigma^2}$, then we can introduce the following product Matérn kernel as a kernel on the product manifold $\mathcal{M}$. This is done by writing

$$k^{\mathcal{M}}_{\nu,\boldsymbol{\kappa},\sigma^2}(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 k^{\mathcal{M}_1}_{\nu,\kappa_1,1}(x_1, x_1') \cdot \ldots \cdot k^{\mathcal{M}_d}_{\nu,\kappa_d,1}(x_d, x_d'), \tag{9}$$

where $\boldsymbol{\kappa} = (\kappa_1, \ldots, \kappa_d)$, $\boldsymbol{x} = (x_1, \ldots, x_d)$ and $\boldsymbol{x}' = (x_1', \ldots, x_d')$. Note that this enables automatic relevance determination in the product Riemannian setting. Because of this, it is an alternative and more expressive way to define a kernel on e.g. a torus or other product manifold. Note that the product of Matérn kernels does not generally coincide with the Matérn kernel on the product manifold, unless $\nu = \infty$ and $\kappa_1 = .. = \kappa_d$. The product of manifolds naturally arises when representing the pose of rigid bodies such as the end-effector of a robotic manipulator, which lies on $\mathbb{R}^3 \times \mathbb{S}^3$.

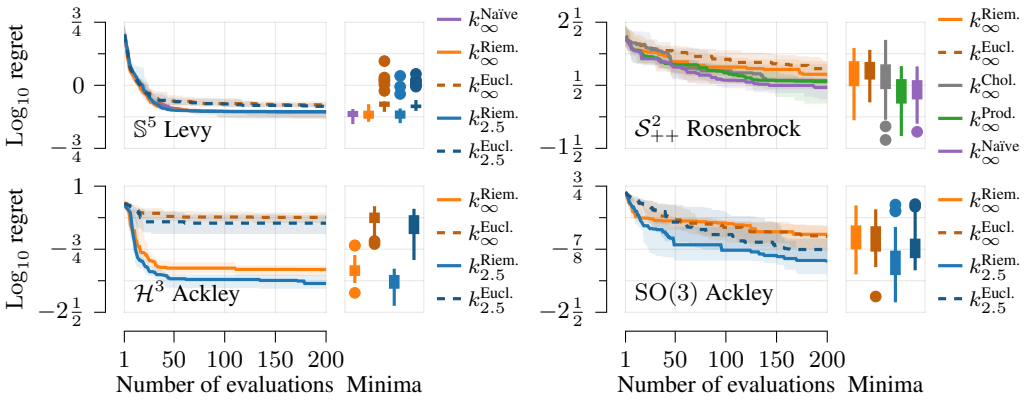

Figure 2: Logarithm of the regret for a set of benchmark test functions on $\mathbb{S}^5$, $\mathrm{SO}(3)$, $\mathcal{S}^2_{++}$, and $\mathcal{H}^3$.

4 Experiments

We evaluate the performance of geometry-aware Bayesian optimization via Riemannian Matérn kernels on a set of benchmark functions and robotics settings, using expected improvement [35] as the acquisition function. Each optimization process is repeated 30 times with 5 random initial samples. Our implementations employ GPyTorch [16], BoTorch [6] and Pymanopt [49] as well as the Robotics Toolbox [11]. Cost parameter values and additional experiments are given in Appendix A.

4.1 Benchmarks

We first study performance using benchmark test functions projected onto the manifolds $\mathbb{S}^5$, $\mathrm{SO}(3)$, $\mathcal{S}^2_{++}$, and $\mathcal{H}^3$. The search domain $\mathcal{X}$ is given by the full manifold, except for $\mathcal{S}^d_{++}$ where we restrict to matrices with eigenvalues $\lambda_i \in [0.001; 5]$. For each manifold, we compare geometry-aware Bayesian optimization against classical Bayesian optimization with Euclidean kernels using constrained optimization on the Euclidean space to minimize the acquisition function. For the sphere and SPD manifolds, we additionally compare against the scheme used by Jaquier et al. [27] featuring a naïve Riemannian squared exponential kernel, where Euclidean distance is replaced by the geodesic distance. For the SPD manifold, we consider two Riemannian kernels, namely (8), and a product of kernels $\mathbb{R}^2 \times \mathbb{S}^2$ which is applied on the eigendecomposition of SPD matrices. We also compare against an alternative implementation using the Cholesky decomposition of an SPD matrix $\mathbf{X} = \mathbf{L}\mathbf{L}^\mathsf{T}$, so that the resulting parameter is the vectorization of the lower triangular matrix $\mathbf{L}$.

Figure 2 shows performance results, including evolution of the median and the distribution of the logarithm of the simple regret of the final recommendation $\boldsymbol{x}_N$ after 200 iterations. We observe that geometry-aware algorithms generally match or outperform their Euclidean counterparts. These differences are more pronounced in higher dimensions, and for manifolds which more significantly depart from the Euclidean setting due to presence of curvature or other geometric features, with the hyperbolic manifold $\mathcal{H}^3$ showing the most pronounced difference. Surprisingly, the naïve Riemannian squared exponential kernel—which for certain length scales may become ill-defined—performs competitively in some settings: this may occur because small length scales keep it out of the problematic regime, and its simple form introduces less numerical error than series truncations or other approaches. Geometry-aware Matérn kernels ($\nu = 2.5$) outperform the squared exponential ones for the test functions with uneven landscapes, such as the Ackley function in Figure 2. Overall, we see that taking geometry into account generally results in faster, more data-efficient convergence and lower variance, at cost of more complex algorithmic implementation.

4.2 Robotics Experiments

Here, we evaluate the performance of geometry-aware Bayesian optimization on several simulated robotics scenarios. In the first experiment, we use Bayesian optimization as an orientation sampler aiming at satisfying the requirements defined by a cost function, similarly to the experiment presented in Jaquier et al. [27]. A velocity-controlled robot samples an orientation reference $\boldsymbol{x} = \hat{\boldsymbol{q}}$ around a

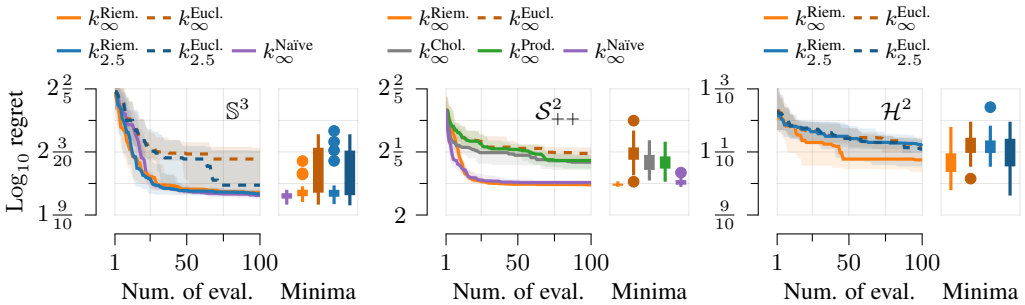

Figure 3: Logarithm of the regret for the orientation, task-compatible manipulability, and path planning problems.

prior orientation $\tilde{\boldsymbol{q}}$. The objective of the robot is to minimize the absolute value of the difference between the prior and the current end-effector orientation $\boldsymbol{q}$ with low joint torques $\boldsymbol{\tau}$ and a high-volume manipulability ellipsoid, i.e., $f(\boldsymbol{q}) = \sum w_{\boldsymbol{q}} \| \operatorname{Log}_{\boldsymbol{q}}(\tilde{\boldsymbol{q}}) \|_1 + w_{\boldsymbol{\tau}} \|\boldsymbol{\tau}\|_2^2 - w_{\mathbf{M}} \det(\mathbf{M})$, where $\det(\mathbf{M})$ is the determinant of the velocity manipulability ellipsoid, proportional to its volume.

Next, we consider a task-compatible manipulability optimization scenario, a common problem in robotics where most solutions do not exploit the geometry of manipulability ellipsoids [21, 29]. Here, an 8-degree-of-freedom planar robot is required to track a desired Cartesian velocity trajectory leading to a vertical line, while tracking a desired manipulability ellipsoid in its nullspace [26]. Optimization aims at finding the desired manipulability $\boldsymbol{x} = \widehat{\mathbf{M}} \in \mathcal{S}^2_{++}$ so that the end-effector trajectory jerk $\dddot{\boldsymbol{p}}$ is minimized and the robot tracks the desired manipulabity aligning with the end-effector movement direction as proposed by Chiacchio [10]. This leads to $f(\widehat{\mathbf{M}}) = \sum w_{\dddot{\boldsymbol{p}}} \|\dddot{\boldsymbol{p}}\|_2 + w_{\mathbf{M}} \| \operatorname{Log}_{\mathbf{M}}(\widehat{\mathbf{M}}) \|_1 + w_{\boldsymbol{t}} \boldsymbol{t}^\top \mathbf{M} \boldsymbol{t}$, with $\mathbf{M}$ the current manipulability and $\boldsymbol{t}$ the unit vector tangential to the end-effector path.

Finally, we consider a simple path planning problem on $\mathbb{R}^2$ for a robot represented by a point. Our goal is to find a collision-free path between two points in the environment. To do so, we leverage geometry-aware Bayesian optimization to estimate paths for sampling, similarly to Marchant and Ramos [32], but introducing hyperbolic geometry. Specifically, we discretize the robot environment by defining a grid on $\mathbb{R}^2$ with 4-connected nodes, represented by a connected undirected graph $\mathcal{G} = (V, E)$, of n vertices V and edges E. A circle-shaped obstacle of unknown location occupies this environment: the robot finds this by sensing collisions while testing sampled paths. The graph $\mathcal{G}$ is mapped to $\mathcal{H}^2$ by learning embeddings in the Lorentz model of hyperbolic geometry, similarly to Nickel and Kiela [36]. At each iteration, the algorithm samples a set of m points $\{\boldsymbol{x}_1, \dots \boldsymbol{x}_m\} \in \mathcal{H}^2$ representing a hyperbolic path of m nodes, which are mapped back to a grid in $\mathbb{R}^2$ in order for the robot to navigate the environment using the resulting path. An optimal path corresponds to the shortest collision-free trajectory from a fixed starting location to the goal $\boldsymbol{x}_g$. We minimize the cost $f(\boldsymbol{x}_1, \dots, \boldsymbol{x}_m) = \sum w_g \operatorname{dist}^{\boldsymbol{x}_g}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_m) + w_d \operatorname{dist}^t(\boldsymbol{x}_1, \dots, \boldsymbol{x}_m)$, where $\operatorname{dist}^{\boldsymbol{x}_g}(\cdot)$ is the distance to the goal $\boldsymbol{x}_g$ (zero in no-collision cases), and $\operatorname{dist}^t(\cdot)$ is the total travel distance.

Figure 3 shows performance results. Here, we also see a moderate improvement in performance when using geometry-aware Bayesian optimization, mirroring the results on benchmark functions. Note that the gained performance improvements may depend on both the manifold geometry and cost function landscape. For instance, the additional orientation experiment reported in Appendix A show less-pronounced improvements as the low-dimensional nature may make the Euclidean and Riemannian kernels very similar. This contrasts with the results in the manipulability and path planning problems, whose geometries depart more from the Euclidean setting.

5 Conclusion

We study geometry-aware Bayesian optimization for robotics, and propose techniques for computing geometry-aware kernels on various Riemannian manifolds occurring in robotics applications. This is achieved by studying how general expressions for Matérn kernels based on stochastic partial differential equations simplify for the manifolds of interest. This provide practitioners with explicit formulas, enabling the use of geometry-aware Bayesian optimization in novel settings. We demonstrate our techniques both on benchmarks and robotics examples to showcase improved performances and potential use cases. We hope these contributions enable greater use of geometry among roboticists.

Acknowledgments

NJ and TA were supported by the Carl Zeiss Foundation under the project JuBot (Jung Bleiben mit Robotern). VB was supported by the Ministry of Science and Higher Education of the Russian Federation, agreement № 075-15-2019-1620 and by "Native towns", a social investment program of PJSC Gazprom Neft.

References

[1] P. A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2007. URL: https://press.princeton.edu/absil. Cited on page 3.

[2] P.-A. Absil, C. G. Baker, and K. A. Gallivan. Trust-Region Methods on Riemannian Manifolds. *Foundations of Computational Mathematics*, 7:303–330, 2007. URL: https://doi.org/10.1007/s10208-005-0179-9. Cited on page 3.

[3] A. Ajoudani, N. G. Tsagarakis, and A. Bicchi. Choosing Poses for Force and Stiffness Control. *IEEE Transactions on Robotics (T-RO)*, 33(6):1483–1490, 2017. URL: https://doi.org/10.1109/TRO.2017.2708087. Cited on page 6.

[4] R. Akrour, D. Sorokin, J. Peters, and G. Neumann. Local Bayesian Optimization of Motor Skills. In *International Conference on Machine Learning (ICML)*, pages 41–50, 2017. URL: http://proceedings.mlr.press/v70/akrour17a.html. Cited on page 1.

[5] R. Antonova, A. Rai, and C. Atkeson. Deep Kernels for Optimizing Locomotion Controllers. In *Conference on Robot Learning (CoRL)*, pages 47–56, 2017. URL: http://proceedings.mlr.press/v78/antonova17a.html. Cited on page 1.

[6] M. Balandat, B. Karrer, D. R. Jiang, S. Daulton, B. Letham, A. G. Wilson, and E. Bakshy. BoTorch: Programmable Bayesian Optimization in PyTorch. *arXiv preprint 1910.06403*, 2019. URL: https://botorch.org/. Cited on page 7.

[7] V. Borovitskiy, I. Azangulov, A. Terenin, P. Mostowsky, M. P. Deisenroth, and N. Durrande. Matérn Gaussian Processes on Graphs. In *Artificial Intelligence and Statistics*, pages 2593–2601, 2021. URL: http://proceedings.mlr.press/v130/borovitskiy21a.html. Cited on page 3.

[8] V. Borovitskiy, A. Terenin, P. Mostowsky, and M. Deisenroth. Matérn Gaussian Processes on Riemannian Manifolds. In *Neural Information Processing Systems (NeurIPS)*, pages 12426–12437, 2020. URL: https://proceedings.neurips.cc/paper/2020/file/92bf5e6240737e0326ea59846a83e076-Paper.pdf. Cited on pages 2–6, 14, 15, 18.

[9] T. Bröcker and T. Tom Dieck. Representations of compact Lie groups. In volume 98. Springer, 2013. URL: https://link.springer.com/book/10.1007/978-3-662-12918-0. Cited on page 18.

[10] P. Chiacchio. Exploiting Redundancy in Minimum-Time Path Following Robot Control. In *American Control Conference*, pages 2313–2318, 1990. URL: https://doi.org/10.23919/ACC.1990.4791142. Cited on page 8.

[11] P. Corke. Robotics Toolbox for Python. URL: https://github.com/petercorke/robotics-toolbox-python. Cited on page 7.

[12] E. De Vito, N. Mücke, and L. Rosasco. Reproducing kernel Hilbert spaces on manifolds: Sobolev and diffusion spaces. *Analysis and Applications*:1–34, 2020. URL: https://doi.org/10.1142/S0219530520400114. Cited on page 15.

[13] A. Feragen and A. Fuster. Geometries and Interpolations for Symmetric Positive Definite Matrices. In *Modeling, Analysis, and Visualization of Anisotropy*, pages 85–113. Springer International Publishing, 2017. URL: https://doi.org/10.1007/978-3-319-61358-1_5. Cited on page 6.

[14] A. Feragen, F. Lauze, and S. Hauberg. Geodesic exponential kernels: When curvature and linearity conflict. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3032–3042, 2015. URL: https://doi.org/10.1109/CVPR.2015.7298922. Cited on page 3.

[15] L. P. Fröhlich, M. N. Zeilinger, and E. D. Klenske. Cautious Bayesian Optimization for Efficient and Scalable Policy Search. In *Conference on Learning for Dynamics and Control (L4DC)*, pages 227–240, 2021. URL: http://proceedings.mlr.press/v144/frohlich21a.html. Cited on page 1.

[16] J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson. GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration. In *Neural Information Processing Systems (NeurIPS)*, 2018. URL: https://gpytorch.ai/. Cited on page 7.

[17] C. S. Gordon. Survey of Isospectral Manifolds. In *Handbook of Differential Geometry*. Volume 1, pages 747–778. Elsevier, 2000. URL: https://doi.org/10.1016/S1874-5741(00)80009-6. Cited on page 17.

[18] I. S. Gradshteyn and I. M. Ryzhik. *Table of Integrals, Series, and Products*. Academic Press, 7th edition, 2014. URL: https://doi.org/10.1016/C2010-0-64839-5. Cited on page 15.

[19] A. Grigoryan and M. Noguchi. The heat kernel on hyperbolic space. *Bulletin of the London Mathematical Society*, 30(6):643–650, 1998. URL: https://doi.org/10.1112/S0024609398004780. Cited on page 6.

[20] Y. Gu. Analysis of orientation representations by Lie algebra in robotics. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 874–879, 1988. URL: https://doi.org/10.1109/ROBOT.1988.12170. Cited on page 2.

[21] L. Guilamo, J. Kuffner, K. Nishiwaki, and S. Kagami. Manipulability optimization for trajectory generation. In *International Conference on Robotics and Automation (ICRA)*, pages 2017–2022, 2006. URL: https://doi.org/10.1109/ROBOT.2006.1642001. Cited on page 8.

[22] C. S. Herz. Les espaces symétriques pour piétons. *Publications mathématiques d'Orsay, Séminaire d'analyse harmonique 79-07*, 1979. URL: https://www.imo.universite-paris-saclay.fr/~biblio/numerisation/docs/31_SEMINAIRE/pdf/31_SEMINAIRE.pdf. Cited on page 19.

[23] J. Humphreys. *Introduction to Lie Algebras and Representation Theory*, volume 9. Springer, 1994. URL: https://link.springer.com/book/10.1007/978-1-4612-6398-2. Cited on page 18.

[24] V. C. Ines Chami Albert Gu and C. Ré. From Trees to Continuous Embeddings and Back: Hyperbolic Hierarchical Clustering. In *Neural Information Processing Systems (NeurIPS)*, pages 15065–15076, 2020. URL: https://proceedings.neurips.cc/paper/2020/file/ac10ec1ace51b2d973cd87973a98d3ab-Paper.pdf. Cited on page 6.

[25] N. Jaquier and L. Rozo. High-Dimensional Bayesian Optimization via Nested Riemannian Manifolds. In *Neural Information Processing Systems (NeurIPS)*, pages 20939–20951, 2020. URL: https://proceedings.neurips.cc/paper/2020/file/f05da679342107f92111ad9d65959cd3-Paper.pdf. Cited on page 3.

[26] N. Jaquier, L. Rozo, D. G. Caldwell, and S. Calinon. Geometry-aware Manipulability Learning, Tracking and Transfer. *International Journal of Robotics Research (IJRR)*, 20(2-3):624–650, 2021. URL: https://journals.sagepub.com/doi/full/10.1177/0278364920946815. Cited on pages 8, 13.

[27] N. Jaquier, L. Rozo, S. Calinon, and M. Bürger. Bayesian Optimization meets Riemannian Manifolds in Robot Learning. In *Conference on Robot Learning (CoRL)*, pages 233–246, 2019. URL: http://proceedings.mlr.press/v100/jaquier20a.html. Cited on pages 2–4, 7.

[28] L. E. Kavraki and S. M. LaValle. Motion Planning. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, pages 139–162. 2016. URL: https://doi.org/10.1007/978-3-319-32552-1_7. Cited on page 2.

[29] I. Lee and J.-H. Oh. Humanoid Posture Selection for Reaching Motion and a Cooperative Balancing Controller. *Journal of Intelligent & Robotic Systems*, 81:301–316, 2016. URL: https://doi.org/10.1007/s10846-015-0225-z. Cited on page 8.

[30] F. Lindgren, H. Rue, and J. Lindström. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498, 2011. URL: https://doi.org/10.1111/j.1467-9868.2011.00777.x. Cited on pages 2, 3.

[31] G. J. Lord, C. E. Powell, and T. Shardlow. *An introduction to computational stochastic PDEs*. Cambridge University Press, 2014. URL: https://doi.org/10.1017/CBO9781139017329. Cited on page 4.

[32] R. Marchant and F. Ramos. Bayesian Optimisation for informative continuous path planning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 6136–6143, 2014. URL: https://doi.org/10.1109/ICRA.2014.6907763. Cited on page 8.

[33] A. Marco, P. Hennig, J. Bohg, and S. S. and S. Trimpe. Automatic LQR Tuning Based on Gaussian Process Global Optimization. In pages 270–277, 2016. URL: https://doi.org/10.1109/ICRA.2016.7487144. Cited on pages 1, 3, 6.

[34] A. Marco, P. Hennig, S. Schaal, and S. Trimpe. On the Design of LQR Kernels for Efficient Controller Learning. In *Conference on Decision and Control (CDC)*, pages 5193–5200, 2017. URL: https://doi.org/10.1109/CDC.2017.8264429. Cited on page 1.

[35] J. Močkus. On Bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference*, pages 400–404, 1975. URL: https://link.springer.com/chapter/10.1007/3-540-07165-2_55. Cited on page 7.

[36] M. Nickel and D. Kiela. Learning Continuous Hierarchies in the Lorentz Model of Hyperbolic Geometry. In *International Conference on Machine Learning (ICML)*, pages 3779–3788, 2018. URL: http://proceedings.mlr.press/v80/nickel18a.html. Cited on page 8.

[37] M. Nickel and D. Kiela. Poincaré Embeddings for Learning Hierarchical Representations. In *Neural Information Processing Systems (NeurIPS)*, 2017. URL: https://proceedings.neurips.cc/paper/2017/file/59dfa2df42d9e3d41f5b02bfc32229dd-Paper.pdf. Cited on page 6.

[38] A. Rai, R. Antonova, S. Song, W. Martin, H. Geyer, and C. Atkeson. Bayesian Optimization Using Domain Knowledge on the ATRIAS Biped. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1771–1778, 2018. URL: https://doi.org/10.1109/ICRA.2018.8461237. Cited on page 1.

[39] C. E. Rasmussen and C. K. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. URL: http://www.gaussianprocess.org/gpml/. Cited on page 15.

[40] L. Roveda, M. Magni, M. Cantoni, D. Piga, and G. Bucca. Human–robot collaboration in sensorless assembly task learning enhanced by uncertainties adaptation via Bayesian Optimization. *Robotics and Autonomous Systems*, 136:103711, 2021. URL: https://doi.org/10.1016/j.robot.2020.103711. Cited on page 3.

[41] L. Rozo. Interactive Trajectory Adaptation through Force-guided Bayesian Optimization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7596–7603, 2019. URL: https://doi.org/10.1109/IROS40897.2019.8968571. Cited on page 3.

[42] K. A. Saar, F. Giardina, and F. Iida. Model-Free Design Optimization of a Hopping Robot and Its Comparison With a Human Designer. *IEEE Robotics and Automation Letters (RA-L)*, 3(2):1245–1251, 2018. URL: https://doi.org/10.1109/LRA.2018.2795646. Cited on page 1.

[43] P. Sawyer. The heat equation on the spaces of positive definite matrices. *Canadian journal of mathematics*, 44(3):624–651, 1992. URL: https://doi.org/10.4153/CJM-1992-038-7. Cited on pages 6, 19.

[44] P. Sawyer. *The Heat Equation on the Symmetric Space Associated with* $\mathrm{SL}(n, \mathbb{R})$. PhD thesis, McGill University Libraries, 1989. URL: https://central.bac-lac.gc.ca/.item?id=TC-QMM-74269&op=pdf&app=Library&oclc_number=897817061. Cited on page 19.

[45] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016. URL: https://doi.org/10.1109/JPROC.2015.2494218. Cited on pages 2, 3.

[46] J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian Optimization of Machine Learning Algorithms. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2012. URL: https://proceedings.neurips.cc/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf. Cited on page 1.

[47] E. M. Stein. *Topics in Harmonic Analysis, Related to the Littlewood-Paley Theory*. Princeton University Press, 1970. URL: https://doi.org/10.1515/9781400881871. Cited on page 18.

[48] Y. Thanwerdas and X. Pennec. Is affine-invariance well defined on SPD matrices? A principled continuum of metrics. In *International Conference on Geometric Science of Information*, pages 502–510. Springer, 2019. URL: https://doi.org/10.1007/978-3-030-26980-7_52. Cited on page 19.

[49] J. Townsend, N. Koep, and S. Weichwald. Pymanopt: A Python Toolbox for Optimization on Manifolds using Automatic Differentiation. *Journal of Machine Learning Research*, 17(137):1–5, 2016. URL: http://jmlr.org/papers/v17/16-177.html. Cited on page 7.

[50] F. Tronarp, T. Karvonen, and S. Särkkä. Mixture Representation of the Matérn Class with Applications in State Space Approximations and Bayesian Quadrature. In *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2018. URL: https://doi.org/10.1109/MLSP.2018.8516992. Cited on page 3.

[51] J.-Y. Wen and K. Kreutz-Delgado. The attitude control problem. *IEEE Transactions on Automatic Control*, 36(10):1148–1162, 1991. URL: https://doi.org/10.1109/9.90228. Cited on page 2.

[52] P. Whittle. Stochastic processes in several dimensions. *Bulletin of the International Statistical Institute*, 40(2):974–994, 1963. Cited on page 4.

[53] A. Wilson, A. Fern, and P. Tadepalli. Using Trajectory Data to Improve Bayesian Optimization for Reinforcement Learning. *Journal of Machine Learning Research*, 15(8):253–282, 2014. URL: http://jmlr.org/papers/v15/wilson14a.html. Cited on page 1.

[54] J. T. Wilson, V. Borovitskiy, A. Terenin, P. Mostowsky, and M. P. Deisenroth. Efficiently Sampling Functions from Gaussian Process Posteriors. In *International Conference on Machine Learning*, pages 10292–10302, 2020. URL: http://proceedings.mlr.press/v119/wilson20a.html. Cited on page 4.

[55] J. T. Wilson, V. Borovitskiy, A. Terenin, P. Mostowsky, and M. P. Deisenroth. Pathwise Conditioning of Gaussian Processes. *Journal of Machine Learning Research*, 22(105):1–47, 2021. URL: https://www.jmlr.org/papers/volume22/20-1260/20-1260.pdf. Cited on page 4.

[56] M. Wong. Weyl Transforms, Heat Kernels, Green Functions and Riemann Zeta Functions on Compact Lie Groups. In *Modern Trends in Pseudo-Differential Operators*, pages 67–85. Springer, 2006. URL: https://doi.org/10.1007/978-3-7643-8116-5_4. Cited on page 18.

[57] T. Yoshikawa. Manipulability of Robotic Mechanisms. *The International Journal of Robotics Research (IJRR)*, 4(2):3–9, 1985. URL: https://doi.org/10.1177/027836498500400201. Cited on page 6.