# Geometry–aware Bayesian Optimization in Robotics using Riemannian Matérn Kernels

**Anonymous Author(s)**
Affiliation
Address
email

**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 and impedance control, manipulability optimization, motion planning, among others, 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 [32] is a technique of growing interest in robotics due to its ability to efficiently solve challenging optimization problems, including controller tuning [5, 21], policy adaptation [4, 11], and robot design [29]. 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. [26] using gait feature transformations to design kernels to optimize locomotion controllers. More generic kernels were introduced by Marco et al. [21], Marco et al. [22], and Wilson et al. [37] 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 can be viewed as elements of the Lie group $\mathrm{SO}(3)$ [14] or the unit-sphere manifold $\mathbb{S}^3$ [35], control gains, inertia and manipulability ellipsoids lie in the manifold of symmetric-positive-definite matrices $\mathcal{S}_{++}$, while the joint configuration of a $d$-degree-of-freedom robot with rotational joints may be

Submitted to the 5th Conference on Robot Learning (CoRL 2021). Do not distribute.

viewed as a point on a torus $\mathbb{T}^d$, i.e. the product of $d$ circles [18], (see Fig. 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 [17].

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: fortunately, mathematical theory and techniques for building the necessary kernels are beginning to emerge [7, 19]. In particular, the recent work of Borovitskiy et al. [7] 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 needed to deploy geometry–aware Bayesian optimization on practical robotics problems.

The goal of this work is to study geometry–aware Bayesian optimization for robotics by connecting emerging geometric Gaussian process tools with the key spaces of interest for robotics. This includes, among others, optimization of stiffness–matrix gains, manipulability ellipsoids, end-effector poses, and graphs for path planning. To do so, we provide practical expressions for computing kernels on a variety of Riemannian manifolds while employing an automatic–differentiation–compatible approach, yielding algorithms extending the work of Jaquier et al. [17]. We extensively benchmark these algorithms on a set of artificial functions and on a number of robotics problems of interest to practitioners. This makes geometry–aware Bayesian optimization more accessible and demonstrates the benefits of incorporating geometry into a roboticist's workflow.

## 2 Geometry-aware Bayesian Optimization

Bayesian optimization [31] is a class of sequential search algorithms for finding a global minimizer

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}) \tag{1}$$

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 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 $\mathcal{S}_{++}$. 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 [21, 27, 28]. 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 the Bayesian optimization performance.

Performing geometry–aware Bayesian optimization on Riemannian manifolds requires one to address two main challenges. Firstly, defining a valid kernel turns out to be 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

One of the key challenges in constructing Gaussian processes on non–Euclidean domains is in defining a valid kernel: seemingly obvious and innocuous–looking candidate expressions such as

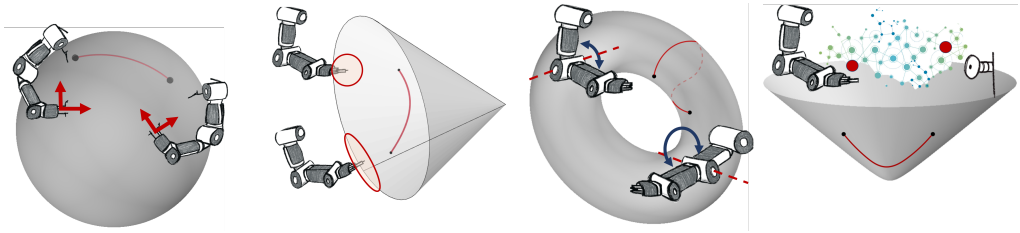

Figure 1: Illustration of different Riemannian manifolds of interest in robotics. Unit sphere $\mathbb{S}^d$, symmetric positive definite matrices $\mathcal{S}^d_{++}$, a torus manifold $\mathbb{T}^d$, and the Lorentz model of hyperbolic geometry in $\mathcal{H}^2$ (from left to right). Red solid lines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

the squared exponential geodesic distance $\exp(-d_g(\boldsymbol{x}, \boldsymbol{x}')^2/\kappa)$ are not valid kernels for all length scales $\kappa > 0$ [10]. In spite of this, they have been shown to be beneficial for practical robotics applications [17]: this motivates one to develop technically sound analogues. To tackle these issues, recent works on non–Euclidean Gaussian processes, including Lindgren et al. [19] and Borovitskiy et al. [7], 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. [7] introduces expressions for the kernels of these processes in terms of the Laplace–Beltrami eigenpairs of their respective manifolds. For many manifolds of interest in robotics, these expressions simplify further [7]. For more general settings including non-compact manifolds, appropriate kernels can often be found in the literature on a case–by–case basis by leveraging a connection between Riemannian squared exponential kernels and heat kernels in the sense of fundamental solutions of heat equations. These can be connected with Matérn kernels by viewing them as certain integrals of squared exponential kernels [34]. We further explore Riemannian Matérn kernels in Section 3. These considerations enable one to train Gaussian processes over manifolds that frequently arise in robotics.

### 2.2 Optimization of acquisition functions on manifolds

To use these Gaussian processes in a Bayesian optimization system, one must further be able 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* [31] 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 optimize the acquisition function in a geometry–aware manner. These algorithms take advantage of the Euclidean tangent space $\mathcal{T}_{\boldsymbol{x}}\mathcal{X}$ linked to each point $\boldsymbol{x}$ on the manifold $\mathcal{X}$. The optimization process typically operates on function gradients, which are intrinsically defined on the tangent space, and projects the result of these computations back onto the manifold with the exponential map $\mathrm{Exp}_{\boldsymbol{x}} : \mathcal{T}_{\boldsymbol{x}}\mathcal{X} \to \mathcal{X}$. Notice that 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*) the optimization of a quadratic subproblem $m_k$ trusted locally in a region around the iterate; (*ii*) the update of the trust region parameters, typically the trust–region radius $\Delta_k$; (*iii*) the iterate update, where a candidate is accepted or rejected in 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 [16] that handles linear constraints. This 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 [36]. These equations naturally generalize to the Riemannian setting, and yield kernels which are well-defined for all Riemannian manifolds and hyperparamter values. 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. [7] prove general expressions for Matérn kernels of the form

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

where $(\lambda_n, f_n)$ are Laplace–Beltrami eigenpairs, $\kappa$ is the length scale, $\sigma^2$ is the variance, $\nu$ is the smoothness and the constants $C_\nu$, $C_\nu$ are chosen so that the resulting kernels have average variance across the domain equal to $\sigma^2$. 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 (2) 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 with the finite element method. 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: examples of this can be seen in the sequel. Although we only present the formulas for squared exponential kernels, respective formulas for Matérn kernels hold as well.

When the manifold of interest is noncompact, the expression (2) does not apply, 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

$$k_{\nu,\kappa,\sigma}(\boldsymbol{x},\boldsymbol{x}') = \int_0^\infty u^{\nu-1}e^{-\frac{2\nu}{\kappa^2}u}k_{\infty,\sqrt{2u},\sigma}(\boldsymbol{x},\boldsymbol{x}')\,\mathrm{d}u. \tag{3}$$

This can be obtained by writing the spectral measure of the Matérn kernel, the T distribution, as a gamma mixture of Gaussians. This 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, which proves to be 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 (3). We thus focus on the heat kernel henceforth.[1]

By leveraging these general expressions for Riemannian Matérn kernels, we now turn our attention to a set of manifolds of key 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 the eigenfunctions of the Euclidean Laplacian,

[1] Note that it is usually beneficial to exploit explicit expressions for the Matérn kernels when they are available, e.g., (2), to alleviate additional numerical errors introduced by approximately computing the integral (3).

which are 1-periodic. Specifically, they correspond to the sines and cosines whose frequency is an integer multiple of $2\pi$. Substituting this into the general formula yields

$$k_{\infty,\kappa,\sigma^2}(\boldsymbol{x},\boldsymbol{x}') = \frac{\sigma^2}{C_\infty} \sum_{\boldsymbol{\tau}\in\mathbb{Z}^d} e^{-2\kappa^2\pi^2|\boldsymbol{\tau}|^2} \cos(2\pi\langle \boldsymbol{\tau}, \boldsymbol{x}-\boldsymbol{x}'\rangle), \tag{4}$$

where the sum is over vectors of integers $\boldsymbol{\tau}$, $\boldsymbol{x},\boldsymbol{x}' \in [0,1)^d$ are the representation of elements of $\mathbb{T}^d$ as vectors of angles divided by $2\pi$, 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, $C_\infty$ is again the normalizing constant chosen so that $k(x,x)=\sigma^2$. The key difference with (2) 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

$$k_{\infty,\kappa,\sigma^2}(\boldsymbol{x},\boldsymbol{x}') = \frac{\sigma^2}{C_\infty} \sum_{n=0}^{\infty} c_{n,d}\, e^{-\frac{\kappa^2}{2}n(n+d-1)}\, \mathcal{C}_n^{(d-1)/2}\big(\cos(d_g(\boldsymbol{x},\boldsymbol{x}'))\big), \tag{5}$$

where $C_\infty$ is a normalizing constant which ensures $k_{\infty,\kappa,\sigma^2}(x,x)=\sigma^2$, $c_{n,d}$ are explicit constants, $\mathcal{C}_n^{(\cdot)}$ are the Gegenbauer polynomials, and $d_g$ is the geodesic distance on the sphere [7]. This kernel is obtained by noting that the eigenfunctions of the spherical Laplace–Beltrami operator are the spherical harmonics, the 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 $\mathbb{R}^d_{\geq 0}$ or $\mathbb{R}^d_{>0}$, we may obtain a kernel over positive (semi)definite matrices.

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

$$k_{\infty,\kappa,\sigma^2}(\mathbf{X},\mathbf{Y}) = \frac{\sigma^2}{C_\kappa}\sum_{\pi} e^{-\frac{\kappa^2}{2}\lambda_\pi} d_\pi \chi_\pi(\mathbf{X}\mathbf{Y}^{-1}), \tag{6}$$

where the sum is over a certain set of tuples of nonnegative integers $\pi$ called the *highest weights* and $C_\kappa$ is the normalizing constant ensuring that $k_{\infty,\kappa,\sigma^2}(\mathbf{X},\mathbf{X})=\sigma^2$ (see Appendix B). Here, $\lambda_\pi$ are Laplacian eigenvalues, $\chi_\pi$ are certain functions called *characters* that depend only on eigenvalues of their argument, and $d_\pi=\chi_\pi(\mathbf{I})$. Partial sums of (6) 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 thorough discussion on its detailed properties to future work.

**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

[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.

graphs [15, 25]. 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 applied in robotics.

Hyperbolic space is non-compact, thus the technique of Borovitskiy et al. [7] 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

$$k^{\mathcal{H}(2)}_{\infty,\kappa,\sigma^2}(\boldsymbol{x}, \boldsymbol{y}) = \frac{\sigma^2}{C_\kappa} \int_\rho^\infty \frac{s \exp\left(-s^2/(2\kappa^2)\right)}{(\cosh(s) - \cosh(\rho))^{1/2}} \, \mathrm{d}s, \quad k^{\mathcal{H}(3)}_{\infty,\kappa,\sigma^2}(\boldsymbol{x}, \boldsymbol{y}) = \frac{\sigma^2}{C_\kappa} \frac{\rho}{\sinh \rho} e^{-\frac{\rho^2}{2\kappa^2}} \tag{7}$$

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

$$k^{\mathcal{H}(d)}_{\infty,\kappa,\sigma^2}(\boldsymbol{x}, \boldsymbol{y}) = -\frac{\sigma^2}{C_\kappa \sinh \rho} \frac{\partial}{\partial \rho} k^{\mathcal{H}(d-2)}_{\infty,\kappa,\sigma^2}(\boldsymbol{x}, \boldsymbol{y}). \tag{8}$$

These expressions are presented in Grigoryan and Noguchi [13].

**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, 38] or as control gain matrices [21]. 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 SPD manifold's geometry [9].

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—a proof is given in Appendix B. To make this relation unique, 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 noncompact, hence the method of Borovitskiy et al. [7] 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 [30], we have

$$k_{\infty,\kappa,\sigma^2}(\mathbf{X}, \mathbf{Y}) = \frac{\sigma^2}{C_\kappa} \exp\left(-\frac{H_1^2 + H_2^2}{2\kappa^2}\right) \int_0^\infty \frac{(2s + \alpha) \exp\left(-s(s+\alpha)/\kappa^2\right)}{(\sinh(s)\sinh(s+\alpha))^{1/2}} \, \mathrm{d}s, \tag{9}$$

where the one-dimensional integral on the right-hand side may be evaluated numerically and $C_\kappa$ 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 $k^{\mathcal{M}_j}_{\nu,\kappa,s\sigma^2}$, then we can introduce the following product Matérn kernel as a kernel on the product manifold $\mathcal{M}$. This is done by writing

$$k^{\mathcal{M}}_{\nu,\boldsymbol{\kappa},\sigma^2}(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 k^{\mathcal{M}_1}_{\nu,\kappa_1,1}(x_1, x_1') \cdot \ldots \cdot k^{\mathcal{M}_d}_{\nu,\kappa_d,1}(x_d, x_d'), \tag{10}$$

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, for example, a torus or other product manifold. 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 $\mathbb{R}^3 \times \mathbb{S}^3$.

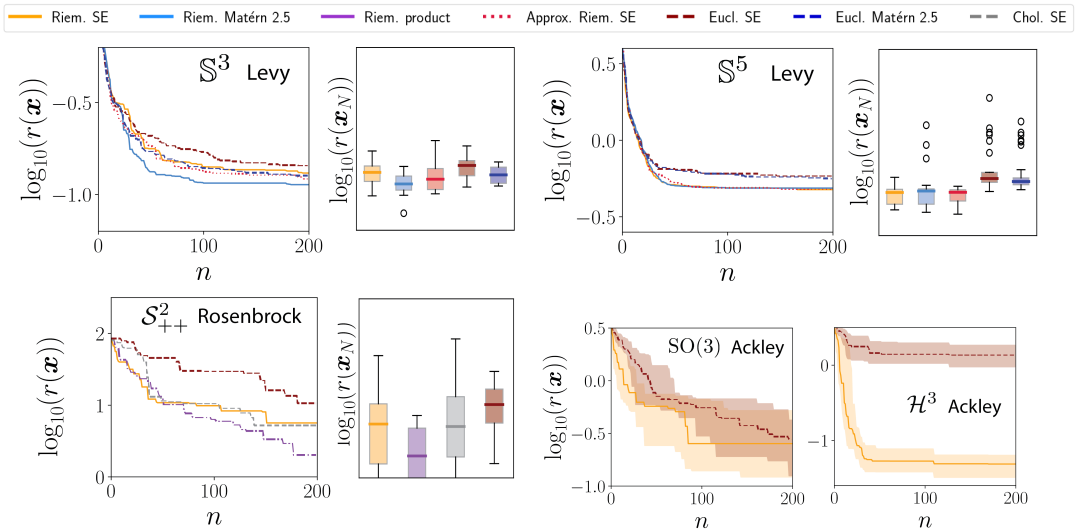


Figure 2: Logarithm of the regret for a set of benchmark test functions on the manifolds $\mathbb{S}^3$, $\mathbb{S}^5$, $\mathrm{SO}(3)$, $\mathcal{S}^2_{++}$, and $\mathcal{H}^3$, for both geometry–aware and Euclidean Bayesian optimization algorithms.

# 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 [23] as the acquisition function. Each optimization process is repeated 30 times with 5 random initial samples. Our implementations employ GPyTorch [12], BoTorch [6] and Pymanopt [33] as well as the Robotics Toolbox [8]. 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 $\mathbb{S}^3$, $\mathbb{S}^5$, $\mathrm{SO}(3)$, $\mathcal{S}^2_{++}$, 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 a constrained optimization on the Euclidean space to minimize the acquisition function. For the sphere, we additionally compare against the scheme used by Jaquier et al. [17] 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 (9), 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 $\boldsymbol{X} = \boldsymbol{L}\boldsymbol{L}^\mathsf{T}$, so that the resulting parameter is the vectorization of the lower triangular matrix $\boldsymbol{L}$.

Figure 2 shows performance results. 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. 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

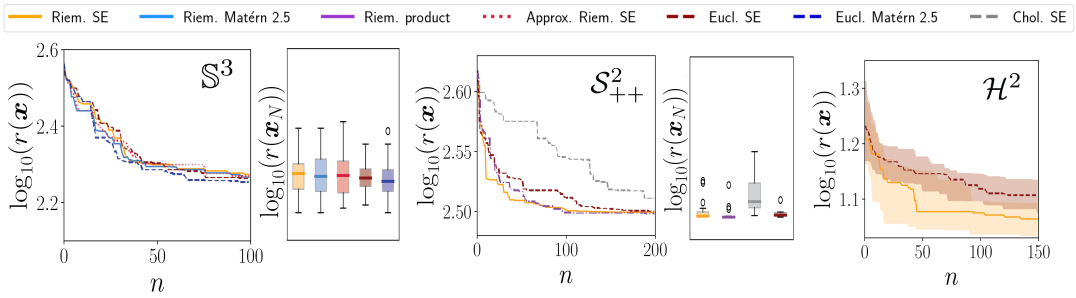


Figure 3: Logarithm of the regret for the orientation, task-compatible manipulability, and path planning optimization problems, for both geometry-aware and Euclidean Bayesian optimization algorithms.

in Jaquier et al. [17]. A velocity-controlled robot samples an orientation reference $\boldsymbol{x} \equiv \hat{\boldsymbol{q}}$ around a prior orientation $\tilde{\boldsymbol{q}}$. The objective of the robot is to minimize the error between the prior and the current end–effector orientation with low joint torques and an isotropic manipulability ellipsoid, i.e., $f(q) = w_{\boldsymbol{q}} \operatorname{dist}_{\mathcal{S}^3}(\tilde{\boldsymbol{q}}, \boldsymbol{q}) + w_{\boldsymbol{\tau}} \|\boldsymbol{\tau}\|^2 + w_{\boldsymbol{M}} \operatorname{cond}(\boldsymbol{M})$, where $\boldsymbol{q}$ is the current end–effector orientation, and $\operatorname{cond}(\boldsymbol{M})$ is the condition number of the velocity manipulability ellipsoid.

Next, we consider a task–compatible manipulability optimization scenario. Here, a 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. Optimization aims at finding the desired manipulability $\boldsymbol{x} \equiv \hat{\boldsymbol{M}} \in \mathcal{S}^2_{++}$ so that the end–effector acceleration $\ddot{\boldsymbol{p}}$ is minimized and the robot tracks the desired manipulabity with a high precision, i.e., $f(\hat{\boldsymbol{M}}) = w_{\ddot{\boldsymbol{p}}} \|\ddot{\boldsymbol{p}}\| + w_{\boldsymbol{M}} \operatorname{dist}_{\mathcal{S}^2_{++}}(\boldsymbol{M}, \hat{\boldsymbol{M}})$ with $\boldsymbol{M}$ the current manipulability.

Finally, we consider a simple path planning problem on $\mathbb{R}^2$ for a robot represented by a point particle. 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, following Marchant and Ramos [20]. 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 with unknown location occupies this environment: the robot finds this by sensing collisions while testing the sampled path. 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 [24]. 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. These "hyperbolic" nodes are then mapped back to the 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 robot location to the goal $\boldsymbol{x}_g$. We minimize the cost $f(\boldsymbol{x}_{1,\dots,m}) = w_g \operatorname{dist}^{\boldsymbol{x}_g}(\boldsymbol{x}_{1,\dots,m}) + w_d \operatorname{dist}^t(\boldsymbol{x}_{1,\dots,m})$, 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. These are least pronounced for the orientation experiment, where the low-dimensional nature may make the Euclidean and Riemannian kernels very similar, limiting the effect of geometry, and most pronounced in the manipulability and path planning problems, which depart more from the Euclidean setting.

## 5 Conclusion

In this work, we study geometry–aware Bayesian optimization for robotics problems, and propose techniques for computing geometry-aware kernels on a number of Riemannian manifolds that occur in robotics applications. This is achieved by studying how general expressions for Matérn kernels based on stochastic partial differential equations simplify for the given manifolds of interest. This provide practitioners with explicit formulas, enabling geometry–aware Bayesian optimization to be implemented in novel settings. We demonstrate our techniques both in benchmark settings to showcase improved performance, and on robotics examples to showcase potential use cases. We hope these contributions enable greater use of geometry among robotics practitioners.

# References


[1] P. A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2007. 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. 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. DOI: 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, Nov. 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. Cited on page 7.

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

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

[9] 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. ISBN: 978-3-319-61358-1. Cited on page 6.

[10] 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.

[11] 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.

[12] 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 2018. Cited on page 7.

[13] 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.

[14] Y. Gu. Analysis of orientation representations by Lie algebra in robotics. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 874–879, 1988. DOI: https://doi.org/10.1109/ROBOT.1988.12170. Cited on page 1.

[15] 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.

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

[17] 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, Osaka, Japan, Oct. 2019. URL: http://proceedings.mlr.press/v100/jaquier20a.html. Cited on pages 2, 3, 7, 8.

[18] 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%5C_7. Cited on page 2.

[19] 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. Cited on pages 2, 3.

[20] 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. DOI: https://doi.org/10.1109/ICRA.2014.6907763. Cited on page 8.

[21] 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, Stockholm, Sweden, May 2016. URL: https://doi.org/10.1109/ICRA.2016.7487144. Cited on pages 1, 2, 6.

[22] 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, Melbourne, Australia, Dec. 2017. URL: https://doi.org/10.1109/CDC.2017.8264429. Cited on page 1.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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. ISSN: 0921–8890. DOI: https://doi.org/10.1016/j.robot.2020.103711. Cited on page 2.

[28] 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. DOI: 10.1109/IROS40897.2019.8968571. Cited on page 2.

[29] 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. DOI: https://doi.org/10.1109/LRA.2018.2795646. Cited on page 1.

[30] 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 page 6.

[31] 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.

[32] 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.

[33] 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. Cited on page 7.

[34] F. Tronarp, T. Karvonen, and S. Särkkä. Mixture Representation of the Matérn Class with Applications in State Space Approximations and Bayesian Quadrature. In *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2018. DOI: 10.1109/MLSP.2018.8516992. Cited on page 3.

[35] 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 1.

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

[37] 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.

[38] 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.