CMAES on Riemannian Manifolds for Optimizing Robotic Manipulation Tasks

Kira Ruschmeier[1], Philipp Schillinger[2], Leonel Rozo[2]

I. Introduction

One practical problem that is typically faced in robotic applications is the optimal parameterization of *skills*, for example, finding the parameters of a controller. Also, skill parameters may need to be adapted for new situations. The aim is to execute a specific robotic manipulation task safely but also efficiently.

Problem Statement. Find a parameterization $\theta^* \in \Theta$ that maximizes the performance objective $f\colon \Theta \to \mathbb{R}$ as

$$\theta^* = \underset{\theta \in \Theta}{\texttt{argmax}}\, f(\theta). \tag{1}$$

Such a problem is usually referred to as *black-box optimization* and can be solved for Euclidean parameter spaces [3]. In this work, however, we assume that Θ is given more generally by a *Riemannian manifold* and propose an extension of the *Covariance Matrix Adaptation Evolution Strategy (CMAES)* algorithm to solve (1).

II. Background

CMAES: A common approach for parameter optimization is *Covariance Matrix Adaptation Evolution Strategy* (CMAES) [3]. This method is a stochastic gradient-free approach which solves black-box optimization problems.

In CMAES, the idea is to sample parameter choices θ from the domain by a Gaussian distribution $\theta \sim \mathcal{N}(m, \Sigma)$ and scaled by a so called step-size σ which denotes the amount of exploration. The parameters are then evaluated by the objective function $f(\theta)$ and the distribution is updated towards the best performing ones. This process is iterated so that over time, parameter choices which promise a good performance are sampled with high probability.

Riemannian Manifold: The optimization domains in robotic manipulation often involve geometric properties that cannot be represented in the Euclidean space. This includes orientation represented as unit quaternions and stiffness matrices. However, the generalization to Riemannian manifolds enables us to mathematically consider such geometric optimization domains.

For each point in the Riemannian manifold $\widehat{x} \in \mathcal{M}$ there exists a Euclidean tangent space $\mathcal{T}_{\widehat{x}}\mathcal{M}$. The so-called *exponential map* $\texttt{Exp}_{\widehat{x}}\colon \mathcal{T}_{\widehat{x}}\mathcal{M} \xrightarrow{\texttt{Exp}_{\widehat{x}}} \mathcal{M}$ transforms points from this tangent space into the manifold and the *logarithmic map* $\texttt{Log}_{\widehat{x}}\colon \mathcal{M} \xrightarrow{\texttt{Log}_{\widehat{x}}} \mathcal{T}_{\widehat{x}}\mathcal{M}$ vice versa.

A third operation, called *parallel transport*, is used to transform vectors from one tangent space to another and is given by $\mathbb{T}\colon \mathcal{T}_{\widehat{x}}\mathcal{M} \xrightarrow{\mathbb{T}} \mathcal{T}_{\widehat{y}}\mathcal{M}$. Applying parallel transport $\mathbb{T}$ to vectors defined in $\mathcal{T}_{\widehat{x}}\mathcal{M}$ results in vectors in $\mathcal{T}_{\widehat{y}}\mathcal{M}$ while keeping their inner product constant.

III. Approach

The original CMAES procedure [3] includes four steps that are summarized in the following. As described for each of them, some operations are only valid for Euclidean spaces and need to be adjusted to operate on Riemannian manifolds.

1) Sampling: Parameter samples s are generated by a Gaussian distribution $\mathcal{N}(\widehat{m}, \Sigma)$, where the mean $\widehat{m} \in \mathcal{M}$ is a point in the Riemannian manifold instead of a Euclidean space. The covariance matrix $\Sigma \in \mathcal{T}_{\widehat{m}}\mathcal{M}$ is defined in the tangent space of $\widehat{m}$ so that points around $\widehat{m}$ are sampled by the distribution. A scaling based on the step-size σ remains as in the original CMAES.

In order to evaluate the performance of a sample s, we use the exponential map to transform it into the manifold $\widehat{s} = \texttt{Exp}_{\widehat{m}}(s)$. Then, a function evaluation $f(\widehat{s})$ determines the value of the objective function for this sample.

2) Mean Update: In order to compute the updated mean $\widehat{m}'$ for the next iteration, all samples are sorted according to their objective function value. Only a certain number of best samples is considered in the following to calculate the mean update.

More specifically, the Fréchet mean of the selected samples is computed here, i.e., the total distance to all samples $\widehat{s}$ is minimized by

$$\widehat{m}' = \underset{\widehat{m} \in \mathcal{M}}{\texttt{argmin}} \sum_{\widehat{s}} d(\widehat{m}, \widehat{s}) \tag{2}$$

where $d(\widehat{m}, \widehat{s})$ denotes the distance in $\mathcal{M}$ between the mean and each of the samples.

An iterative algorithm is used to compute the Fréchet mean in Riemannian manifold where $\widehat{m}'$ is approximated over multiple iterations. Note that this is in contrast to computing the weighted sum of all samples as sufficient in the Euclidean space and done in the original CMAES.

3) Covariance Update: The adaptation of Σ follows the procedure of the original CMAES except for the need to perform a parallel transport operation of the eigenvectors $e^{\widehat{m}}$

[1]Maastricht University, Department of Data Science and Knowledge Engineering, Maastricht, Netherlands. Work conducted while at [2].
k.ruschmeier@student.maastrichtuniversity.nl
[2]Bosch Center for Artificial Intelligence, Renningen, Germany.
<first name>.<last name>@de.bosch.com

of Σ and the current path variable $p_c^{\widehat{m}}$ of a so called rank-1-update $r_1 = p_c^{\widehat{m}'} {p_c^{\widehat{m}'}}^T$. This r_1 update ensures a correlation between generations. In other words, $e^{\widehat{m}}$ and $p_c^{\widehat{m}}$ are computed in $\mathcal{T}_{\widehat{m}}\mathcal{M}$ with respect to the previous mean $\widehat{m}$. Then, parallel transport is used to convert those parameters from $\mathcal{T}_{\widehat{m}}\mathcal{M}$ to $\mathcal{T}_{\widehat{m}'}\mathcal{M}$. Consequently, each variable is defined in the current tangent space.

4) Step-size Update: Finally, also the update of the step-size σ depends on a path variable $p_\sigma^{\widehat{m}}$. Similar to $p_c^{\widehat{m}}$, calculated for the covariance adaptation, $p_\sigma^{\widehat{m}}$ needs to be parallel transported to $\mathcal{T}_{\widehat{m}'}\mathcal{M}$ for performing the update.

Summarizing, CMAES on Riemannian manifold works by first sampling parameter choices s by a Gaussian distribution $s \sim \mathcal{N}(\widehat{m}, \Sigma)$, whereby $\widehat{m}$ is represented in the manifold. The samples s in the tangent space are then transformed to samples $\widehat{s}$ in the manifold by the *exponential map*. Afterwards, the objective value is determined and sorted accordingly from best to worst. A certain number of best samples is used to update the mean by an iterative Fréchet mean. Σ as well as σ are updated as in the original CMAES except for those elements which are computed in the old tangent space. These elements need to be parallel transported to the new tangent space.

IV. Discussion

In order to evaluate CMAES on Riemannian manifold two different experiments are conducted, see Fig. 1. The aim of the first experiment is to insert a peg into a hole by optimizing the pose of an end-effector such that the contact force is minimized. The second experiment adapts the orientation of an end-effector while moving to different poses such that the joint velocity is minimized. In both experiments, CMAES on Riemannian manifolds performs better than the original CMAES suitable for Euclidean parameter spaces which can be seen in the respective right image in Fig. 1.

The motion profile experiment finds the optimum faster when our approach is applied instead of the original CMAES. In the peg insertion experiment, the number of successfully inserted samples lies at around two third when using this recent approach whereas applying the original CMAES to this problem obtains a success rate of less than half of the samples.

There are other methods which are suitable for black-box optimization when parameters belong to a Riemannian manifold. The work of Jaquier et al. [4] extends *Bayesian Optimization* (BO) [5], another commonly used black-box optimization method, to Riemannian manifolds by using the Riemannian distance as a kernel distance for measuring geometry-aware similarities in the parameter space. Furthermore, the minimization of the acquisition function takes place on the manifold. The search direction for the next query point is modified by parallel transport.

[1] adapts the original CMAES to Riemannian manifolds to be exploited on 3D image reconstruction. This work also parallel transports all necessary elements from the old

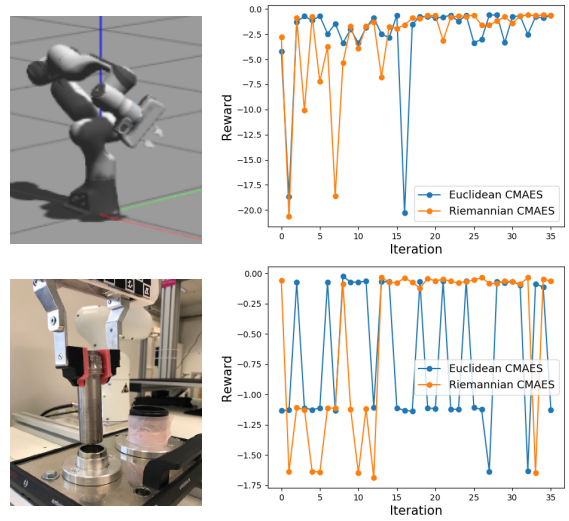

Fig. 1. The proposed method has been evaluated for robotic manipulation tasks in simulation (top) and on a real robot (bottom).

tangent space to the new tangent space as our approach proposes. Instead of using an iterative algorithm to compute a Fréchet mean, however, only a weighted sum in the tangent space is calculated for updating the mean. Another difference lies in the application. Our approach considers robotic applications whereas Colutto et al.'s [1] work is applied on images.

Another recent work is proposed in the thesis by Fong [2]. He proposes a framework which extends any Stochastic Derivative-Free Optimization algorithm to Riemannian manifolds and also tested the framework on CMAES. Here, a difference to this approach can be discovered. Instead of computing the mean by an iterative approach, the mean is updated by the weighted sum.

V. Conclusion

In this work, CMAES is extended from the Euclidean space to Riemannian manifolds such that robotic manipulation tasks can be optimized properly. For evaluation, two experiments are conducted and compared to the original CMAES. As a result, the proposed CMAES on Riemannian manifold outperforms CMAES on an approximation in the Euclidean space in both experiments.

References

[1] Sebastian Colutto, Florian Fruhauf, Matthias Fuchs, and Otmar Scherzer. The CMA-ES on Riemannian manifolds to reconstruct shapes in 3-D voxel images. *IEEE Transactions on Evolutionary Computation*, 14(2):227–245, 2009.

[2] Robert Simon Fong. *Stochastic optimization on Riemannian manifolds*. PhD thesis, University of Birmingham, 2020.

[3] Nikolaus Hansen. The CMA evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*, 2016.

[4] Noémie Jaquier, Leonel Rozo, Sylvain Calinon, and Mathias Bürger. Bayesian optimization meets Riemannian manifolds in robot learning. In *Conference on Robot Learning*, pages 233–246. PMLR, 2020.

[5] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.