
This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Abu-Dakka, Fares J.; Kyrki, Ville

Geometry-aware Dynamic Movement Primitives

Published in:

Proceedings of the IEEE Conference on Robotics and Automation, ICRA 2020

DOI:

[10.1109/ICRA40945.2020.9196952](https://doi.org/10.1109/ICRA40945.2020.9196952)

Published: 01/01/2020

Document Version

Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:

Abu-Dakka, F. J., & Kyrki, V. (2020). Geometry-aware Dynamic Movement Primitives. In *Proceedings of the IEEE Conference on Robotics and Automation, ICRA 2020* (pp. 4421-4426). Article 9196952 (IEEE International Conference on Robotics and Automation). IEEE. <https://doi.org/10.1109/ICRA40945.2020.9196952>

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

© 2020 IEEE. This is the author's version of an article that has been published by IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Geometry-aware Dynamic Movement Primitives

Fares J. Abu-Dakka and Ville Kyrki

Abstract—In many robot control problems, factors such as stiffness and damping matrices and manipulability ellipsoids are naturally represented as symmetric positive definite (SPD) matrices, which capture the specific geometric characteristics of those factors. Typical learned skill models such as dynamic movement primitives (DMPs) can not, however, be directly employed with quantities expressed as SPD matrices as they are limited to data in Euclidean space. In this paper, we propose a novel and mathematically principled framework that uses Riemannian metrics to reformulate DMPs such that the resulting formulation can operate with SPD data in the SPD manifold. Evaluation of the approach demonstrates that beneficial properties of DMPs such as change of the goal during operation apply also to the proposed formulation.

I. INTRODUCTION

Day by day realistic robotic applications are bringing robots into human environments such as houses, hospitals, and museums where they are expected to assist us in our daily life tasks. Such human-inhabited environments are highly unstructured, dynamic and uncertain, making hard-coding the environments and related skills infeasible.

In this context, human expertise can be exploited to teach robots how to perform such tasks by transferring human skills to robots [1]. Learning-from-human-demonstrations (LfD) has been widely studied as a convenient way to transfer human skills to robots. This learning approach is aimed at extracting relevant motion patterns from human demonstrations and subsequently applying these patterns to different situations. In the past decades, several LfD based approaches have been developed such as: dynamic movement primitives (DMP) [2], [3], probabilistic movement primitives (ProMP) [4], Gaussian mixture models (GMM) along with Gaussian mixture regression (GMR) [5], and more recently, kernelized movement primitives (KMP) [6], [7]. DMPs have several beneficial properties such as robustness against perturbations and ability to adapt to new requirements such as a new goal.

However, many tasks in those environments require variable impedance [8], [9], [10] or high manipulability [11], [12], [13], [14]), the parameters of which are encapsulated in symmetric positive definite (SPD) matrices. Because of the structure of the manifold of SPD matrices, standard LfD approaches such as DMPs can not be directly used as they rely on Euclidean parametrization of the space.

In this paper, we propose a novel formulation for DMPs using Riemannian metrics such that the resulting formulation can operate with SPD data. This allows variable SPD quantities

to be modeled while retaining the useful properties of standard DMPs. The main contributions are

- 1) A novel and mathematically principled framework for reformulating DMPs using Riemannian metrics, in order to learn and reproduce SPD-matrices-based robot skills.
- 2) Reformulating standard DMP goal switching to be able to handle SPD-matrix-based robot skills.

This extension of DMPs to Riemannian manifolds allows the generation of smooth trajectories for data that do not belong to the Euclidean space.

The work is inspired by quaternion and rotation matrix based formulations of DMPs [3], [15] which target specifically the problem of parametrizing the space of orientations $SO(3)$. Compared to the tensor-based formulation of GMM and GMR on Riemannian manifold of SPD matrices [16] (demonstrated for manipulability transfer in [14]), the proposed approach allows adapting the SPD profile on-line to a new goal, similar to the difference between standard GMM/GMR and DMPs for Euclidean quantities.

This paper is organized as follows: We begin by providing background about standard DMPs (Section II-A) and Riemannian manifold of SPD matrices (Section II-B). Afterwards, we exploit Riemannian manifold to derive the new formulation of DMPs (Section III-A) followed by goal switching formulation (Section III-B). Subsequently, we evaluate our approach through several examples (Section IV). The work is concluded in Section V.

II. BACKGROUND

In this scope we introduce a brief introduction to standard DMPs and Riemannian manifold of SPD matrices.

A. Dynamic Movement Primitives

The basic idea of DMPs is to model movements by a system of differential equations that ensure some desired behavior, e.g. convergence to the specified attractor point [17], [2], [3]. A DMP for a single degree of freedom trajectory y is defined by the following set of nonlinear differential equations [2]

$$\tau \dot{z} = \alpha_z(\beta_z(g - y) - z) + f(x), \quad (1)$$

$$\tau \dot{y} = z, \quad (2)$$

$$\tau \dot{x} = -\alpha_x x, \quad (3)$$

where z is the scaled velocity, x is the phase variable to avoid explicit time dependency and $x(0) = 1$, α_z and β_z define the behavior of the 2nd-order system, g is the goal of the movement, and $f(x)$ is a nonlinear forcing term that provides

a modeling of complex trajectories. Choosing a time constant $\tau > 0$ along with $\alpha_z = 4\beta_z$ and $\alpha_x > 0$ will make the linear part of (1) and (2) critically damped, which insures the convergence of y and z to a unique attractor point at $y = g$ and $z = 0$ [2]. $f(x)$ is defined as a linear combination of N nonlinear radial basis functions, which enables the robot to follow any smooth trajectory from the initial position y_0 to the final configuration g

$$f(x) = \frac{\sum_{i=1}^N w_i \Psi_i(x)}{\sum_{i=1}^N \Psi_i(x)}, \quad (4)$$

$$\Psi_i(x) = \exp\left(-h_i(x - c_i)^2\right), \quad (5)$$

where c_i are the centers of Gaussians distributed along the phase of the movement and h_i their widths. For a given N and setting τ equal to the total duration of the desired movement, we can define $c_i = \exp\left(-\alpha_x \frac{i-1}{N-1}\right)$, $h_i = \frac{1}{(c_{i+1} - c_i)^2}$ and $h_N = h_{N-1}$ where $i = 1, \dots, N$. For each DoF, the weights w_i should be adjusted from the measured data so that the desired behavior is achieved. For controlling a robotic system with more than one DoF, we represent the movement of every DoF with its own equation system (1)–(2), but with the common phase (3) to synchronize them.

B. Riemannian manifold of SPD matrices

Let us define \mathcal{S}_{++}^m as the set of $m \times m$ SPD matrices which cannot be considered as a vector space since it is not closed under addition and scalar product [18], and thus the use of classical Euclidean space methods for treating and analyzing these matrices is inadequate. A compelling solution is to incorporate these matrices with a Riemannian metric, which allows the set of SPD matrices to form a Riemannian manifold [18]. Such a metric defines the geodesics, that is, the minimum length curves between two points on the manifold.

A Riemannian manifold \mathcal{M} is a topological space, each point of which locally resembles a Euclidean space. For each point $\mathbf{p} \in \mathcal{M}$, there exists a tangent space $\mathcal{T}_{\mathbf{p}}\mathcal{M}$ which corresponds to the space of symmetric matrices for the case of the SPD manifold. The metric in the tangent space is flat, which allows the use of classical arithmetic tools. Note that the space of \mathcal{S}_{++}^m can be represented as the interior of a convex cone embedded in its tangent space of symmetric $m \times m$ matrices \mathbf{Sym}^m .

To operate on the tangent spaces, a mapping system is required to switch between $\mathcal{T}_{\mathbf{p}}\mathcal{M}$ and \mathcal{M} . The two mapping operators are known as exponential and logarithmic maps: *The logarithmic map* $\text{Log}_{\mathbf{Q}}(\mathbf{Q}): \mathcal{M} \mapsto \mathcal{T}_{\mathbf{Q}}\mathcal{M}$ is a function that maps a point in the manifold $\mathbf{Q} \in \mathcal{M}$ to a point in the tangent space $\mathbf{\Delta} \in \mathcal{T}_{\mathbf{Q}}\mathcal{M}$.

$$\text{Log}_{\mathbf{Q}}(\mathbf{Q}) = \mathbf{\Gamma}^{\frac{1}{2}} \text{logm}\left(\mathbf{\Gamma}^{-\frac{1}{2}} \mathbf{Q} \mathbf{\Gamma}^{-\frac{1}{2}}\right) \mathbf{\Gamma}^{\frac{1}{2}}, \quad (6)$$

The exponential map $\text{Exp}_{\mathbf{Q}}(\mathbf{\Delta}): \mathcal{T}_{\mathbf{Q}}\mathcal{M} \mapsto \mathcal{M}$ is a function that maps a point $\mathbf{\Delta} \in \mathcal{T}_{\mathbf{Q}}\mathcal{M}$ to a point $\mathbf{Q} \in \mathcal{M}$, so that it

lies on the geodesic starting from $\mathbf{Q} \in \mathcal{S}_{++}^m$ in the direction of $\mathbf{\Delta}$.

$$\text{Exp}_{\mathbf{Q}}(\mathbf{\Delta}) = \mathbf{\Gamma}^{\frac{1}{2}} \text{expm}\left(\mathbf{\Gamma}^{-\frac{1}{2}} \mathbf{\Delta} \mathbf{\Gamma}^{-\frac{1}{2}}\right) \mathbf{\Gamma}^{\frac{1}{2}}, \quad (7)$$

where $\text{logm}(\cdot)$ and $\text{expm}(\cdot)$ are the matrix logarithm and exponential functions.

Moving elements between different tangent spaces is performed by the parallel transport operator [19], [20]. The parallel transport $\mathbb{B}_{\mathbf{Q} \mapsto \mathbf{Q}'}(\mathbf{V}): \mathcal{T}_{\mathbf{Q}}\mathcal{M} \mapsto \mathcal{T}_{\mathbf{Q}'}\mathcal{M}$ is a function that transports $\mathbf{V} \in \mathcal{T}_{\mathbf{Q}}\mathcal{M}$ to $\mathcal{T}_{\mathbf{Q}'}\mathcal{M}$ over the geodesic from \mathbf{Q} to \mathbf{Q}' is given by

$$\mathbb{B}_{\mathbf{Q} \mapsto \mathbf{Q}'}(\mathbf{V}) = \mathbf{C}_{\mathbf{Q} \mapsto \mathbf{Q}'} \mathbf{V} \mathbf{C}_{\mathbf{Q} \mapsto \mathbf{Q}'}^T, \quad (8)$$

where $\mathbf{C}_{\mathbf{Q} \mapsto \mathbf{Q}'} = (\mathbf{Q}' \mathbf{\Gamma}^{-1})^{\frac{1}{2}}$. Equation (8) has been proved to be computationally efficient [20]. This transporter is exploited whenever it is required to transport SPD matrices along geodesics in a nonlinear manifold.

In this paper, we exploit the Riemannian manifold to reformulate DMPs to be capable of encoding and reproducing SPD-matrices-based robot skills.

III. PROPOSED APPROACH

In this section, we provide a complete formulation for DMPs in order to learn and reproduce SPD-matrices-based robot skills. For the sake of simplicity let us first recall the re-interpretation of basic standard operations in a Riemannian manifold (Table I). Define $\mathbf{A}, \mathbf{B} \in \mathcal{M}$ and $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$.

TABLE I
RE-INTERPRETATION OF BASIC STANDARD OPERATIONS IN A
RIEMANNIAN MANIFOLD [18].

	Euclidean space	Riemannian manifold
Subtraction	$\mathbf{a} - \mathbf{b} = \mathbf{a} - \mathbf{b}$	$\mathbf{A} - \mathbf{B} = \text{Log}_{\mathbf{A}}(\mathbf{B})$
Addition	$\mathbf{a} + \mathbf{b} = \mathbf{a} + \mathbf{b}$	$\mathbf{A} + \mathbf{B} = \text{Exp}_{\mathbf{A}}(\mathbf{A} - \mathbf{B})$
Distance	$\text{dist}(\mathbf{a}, \mathbf{b}) = \ \mathbf{a} - \mathbf{b}\ $	$\text{dist}(\mathbf{A}, \mathbf{B}) = \ \mathbf{A} - \mathbf{B}\ _{\mathbf{A}}$
Interpolation	$\mathbf{a}(t) = \mathbf{a}_1 + t(\mathbf{a}_2 - \mathbf{a}_1)$	$\mathbf{A}(t) = \text{Exp}_{\mathbf{A}_1}(t(\mathbf{A}_2 - \mathbf{A}_1))$

A. Geometry-aware DMPs Formulation

Define a variable $\mathbf{X} \in \mathcal{S}_{++}^m$ as an arbitrary SPD matrix and $\mathbf{\Xi} = \{t_i, \mathbf{X}_i\}_{i=1}^T$ as the set of SPD matrices in one demonstration. In order to prepare the demonstration data for DMP, its 1st- and 2nd-time derivatives are needed. The 1st-time derivative is computed as follows

$$\dot{\mathbf{\Xi}} = (\text{Log}_{\mathbf{X}_{i-1}}(\mathbf{X}_i))/dt, \quad (9)$$

where each $d\mathbf{\Xi}_i$ belongs to the corresponding tangent space $\mathcal{T}_{\mathbf{X}_{i-1}}\mathcal{M}$. Afterwards, we use (8) to move all $d\mathbf{\Xi}_i$ to a common/shared arbitrary tangent space, e.g. the tangent space of the first SPD data $\mathcal{T}_{\mathbf{X}_1}\mathcal{M}$. The tangent space $\mathcal{T}_{\mathbf{X}_1}\mathcal{M}$ corresponds to \mathbf{Sym}^m , which allows the use of classical arithmetic tools as mentioned in section II-B. To avoid replicating information due to symmetry, we propose to reduce the space dimensionality of the data in the tangent space to $n = m + m(m-1)/2$ using Mandel's representation.

$$\text{vec}(\mathbf{\Sigma}) = \boldsymbol{\sigma}, \quad \boldsymbol{\sigma} \in \mathbb{R}^n, \quad (10)$$

where $vec(\cdot)$ is a function that transforms a symmetric matrix into a vector using Mandel's notation. E.g. a vectorization of a 2×2 symmetric matrix is

$$vec\left(\begin{pmatrix} a & b \\ b & d \end{pmatrix}\right) = \begin{pmatrix} a \\ d \\ \sqrt{2}b \end{pmatrix}. \quad (11)$$

Now, the 2nd-derivatives $\ddot{\Sigma}$ can be computed straight forward using standard Euclidean tools and its vectorization is denoted as $\dot{\sigma}$. Having all necessary data $\{t_l, \mathbf{X}_l, \sigma_l, \dot{\sigma}_l\}_{l=1}^T$, we transform the standard DMP system (1)–(2) into a geometry-aware form as follows

$$\tau \dot{\sigma} = \alpha_z (\beta_z vec(\mathbb{B}_{\mathbf{X}_l \mapsto \mathbf{X}_1}(\text{Log}_{\mathbf{X}_l}(\mathbf{X}_g))) - \sigma) + \mathcal{F}(x), \quad (12)$$

$$\tau \dot{\xi} = \sigma, \quad (13)$$

where ξ is the vectorization of Ξ . $\mathbf{X}_g \in \mathcal{S}_{++}^m$ represents the goal SPD matrix. $vec(\mathbb{B}_{\mathbf{X}_l \mapsto \mathbf{X}_1}(\text{Log}_{\mathbf{X}_l}(\mathbf{X}_g)))$ is the vectorization of the transported symmetric matrix $\text{Log}_{\mathbf{X}_l}(\mathbf{X}_g)$ over the geodesic from \mathbf{X}_l to \mathbf{X}_1 . The forcing term $\mathcal{F}(x)$ can be recalculated as

$$\mathcal{F}(x) = \frac{\sum_{i=1}^N \mathcal{W}_i \Psi_i(x_l)}{\sum_{i=1}^N \Psi_i(x_l)} x_l = \tau \dot{\sigma} - \alpha_z (\beta_z vec(\mathbb{B}_{\mathbf{X}_l \mapsto \mathbf{X}_1}(\text{Log}_{\mathbf{X}_l}(\mathbf{X}_g))) - \sigma), \quad (14)$$

where the phase $x_l = x(t_l) = \exp(-\frac{\alpha_x}{\tau} t_l)$. Using (14), the weights $\mathcal{W}_l \in \mathbb{R}^n$ can be estimated by encoding any sampled SPD-matrices-based robot skills.

In the reproduction, equation (13) is integrated as follows

$$\hat{\mathbf{X}}(t + \delta t) = \text{Exp}_{\mathbf{X}(t)}\left(\frac{\mathbb{B}_{\mathbf{X}_l \mapsto \mathbf{X}(t)}(mat(\sigma(t)))}{\tau} \delta t\right). \quad (15)$$

where the function $mat(\cdot)$ is the inverse of $vec(\cdot)$ and denotes to the matricization using Mandel's notation. $\hat{\mathbf{X}} \in \mathcal{S}_{++}^m$ represents the new SPD-matrices-based robot skills.

B. Goal switching

In the standard DMP formulation, in case of sudden goal switching (e.g. based on external sensory information) during the execution, Ijspeert *et al.* [2] proposed to add an additional equation to the dynamic system (1)–(2) in order to smoothly change the goal g in (1) to a new goal g_{new} as

$$\tau \dot{g} = \alpha_g (g_{new} - g), \quad (16)$$

where α_g is a constant. This equation transforms g from being a constant to a continuous variable. Analogously, SPD-based DMP can switch the goal using

$$\tau \dot{\mathbf{g}} = \alpha_g \text{Log}_{\mathbf{g}}(\mathbf{g}_{new}). \quad (17)$$

so \mathbf{g} now is updated continually.

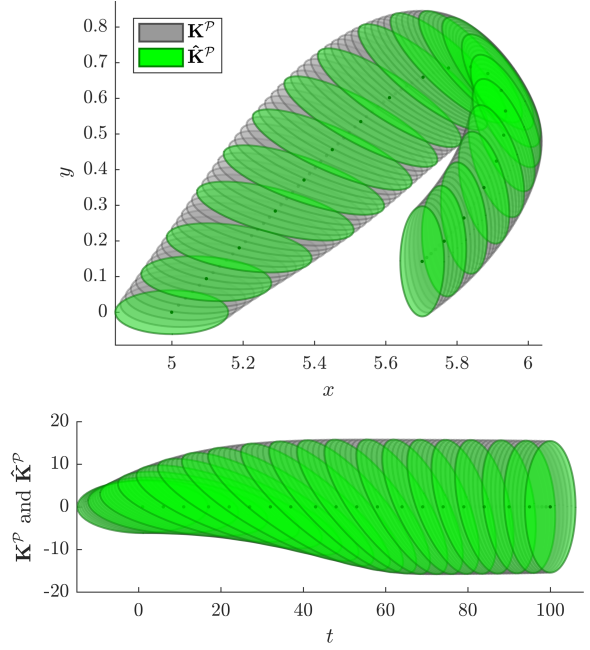


Fig. 1. Learning and reproduction of variable stiffness profile for MSD system. *Top*: The reproduced stiffness ellipsoids in different time steps (in green) coinciding the demonstration (in gray) over the Cartesian trajectory of the MSD. *Bottom*: The reproduced stiffness ellipsoids in different time steps (in green) coinciding the ground truth (in gray) over time.

IV. EXPERIMENTS

We evaluated the proposed imitation learning framework using simulated data. All algorithms have been implemented in MATLAB[®] using a workstation running Ubuntu 16.04 LTS with Intel Core i9-8950HK CPU @ 2.90GHz \times 12, 16 GB of RAM.

Four different experiments were carried out to evaluate the proposed framework:

- Learning and reproducing full stiffness matrix profiles with a 2-DoF virtual-mass spring-damper system (MSD).
- Goal switching applied to full stiffness matrix profiles.
- Learning and reproduction manipulability ellipsoids and transfer them from a teacher to a learner.
- Comparison of the resulting SPD profile between the proposed DMP and GMM/GMR proposed in [16].

A. Learning Variable Impedance Skills

In this scope, we propose to use a simulation of MSD to evaluate our geometry-aware DMPs for learning and reproducing variable impedance¹ skills. Note that stiffness matrices \mathbf{K}^P belong to the space of \mathcal{S}_{++}^m . The MSD system starts from an initial, horizontally-aligned, stiffness ellipsoid \mathbf{K}^P at rest position. Afterwards, external forces \mathbf{f}^e are applied to stimulate the MSD system. During the stimulation, \mathbf{K}^P is rotating through $\mathbf{R}^T \mathbf{K}^P \mathbf{R}$ (\mathbf{R} is a rotation matrix) until it ends up with a vertically-aligned ellipsoid as shown in Fig. 1 in gray. This demonstration then is encoded using

¹Here we refer to variable impedance as variable stiffness profiles. Nevertheless, the proposed approach can also be used to learn variable damping controllers as well as any SPD-matrix-based robot skills.

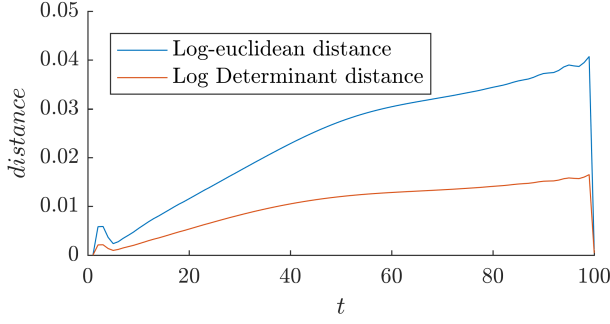


Fig. 2. The distance between the resulting stiffness profile from the proposed DMPs and the ground truth using different metrics.

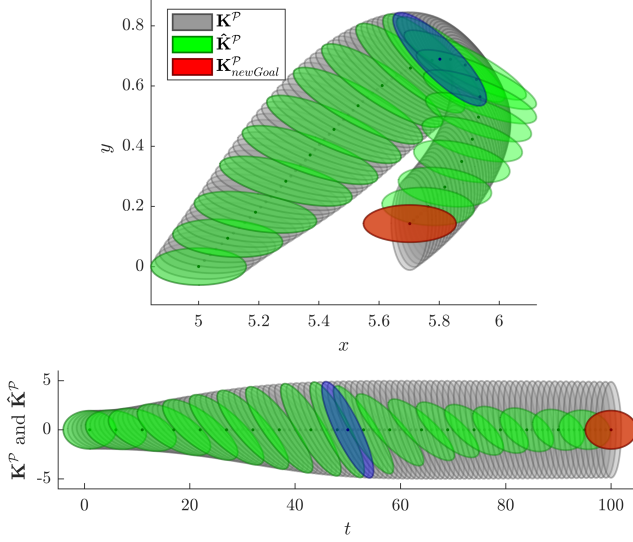


Fig. 3. Response of the proposed DMPs to the switching of stiffness goal during the movement over Cartesian trajectory (Top) and over Time (Bottom). The stiffness goal was changed at the middle of the movement (blue ellipsoid). The new goal (in red) is the original ellipsoid but rotated by 90 degrees.

(12)–(13) to reproduce the ellipsoids in green $\hat{\mathbf{K}}^P$. From the figure, we can see the match between the results of the SPD-based DMPs and the demonstration.

Figure 2 tests the accuracy of the proposed SPD-based DMP by calculating the distance between the resulting SPD profile and the demonstration one. Here we used two distance metric systems: (i) Log-Euclidean distance [21]. (i) Jensen-Bregman Log-Determinant distance [22].

B. Goal Switching

In this simulation we used the same MSD setup introduced in section IV-A. However, here we are about to test the response of the proposed SPD-based DMP to sudden goal changing during the execution. At the middle of DMP execution we changed the goal by rotating it 90 degrees to be horizontally-aligned (red ellipsoid in Fig. 3) instead of being vertically-aligned (in gray).

Figure 3 shows the smoothness of the adaptation of the stiffness profile (in green) to the new goal (in red). The blue stiffness ellipsoid marks the instant of goal switching. It is clear from the figure that the resulting profile was following the demonstrated one until the blue ellipsoid, then started

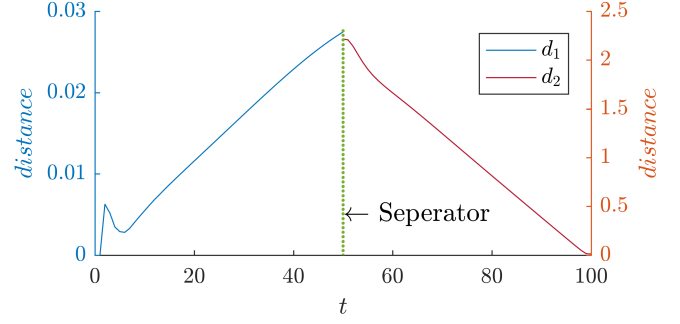


Fig. 4. d_1 Represents the distance between the demonstrated stiffness profile and the resulting DMP profile just before the goal switching occurs. d_2 is the distance between the resulting stiffness from the DMPs and the new goal. d_2 shows how the DMP is adapting and approaching to the new goal.

to adapt to the new goal. More clarification regarding the accuracy of the approach can be seen in Fig. 4. The blue part of the figure shows the distance before the occurrence of goal switching. However, the red part shows the distances between the SPD-based DMP results and the new goal. The figure illustrates that the system converges to the new goal.

C. Learning Manipulability Ellipsoids

In general terms, manipulability can be defined as the capacity of change in position and orientation of the end-effector of a robot given a joint configuration, and mathematically, it is represented as an ellipsoid in a b -dimensional Euclidean space. The manipulability of a robotic arm is derived from the relation between joint and task velocities ($\dot{\mathbf{q}}$ and $\dot{\mathbf{s}}$ respectively),

$$\dot{\mathbf{q}} = \mathbf{J}^+ \dot{\mathbf{s}}, \quad (18)$$

where $\mathbf{q} \in \mathbb{R}^d$ and $\mathbf{J} \in \mathbb{R}^{6 \times d}$ are the joint position and Jacobian of d -DoF arm robot. The superscript $+$ denotes the pseudo-inverse of a matrix. In addition, Consider the set of joint velocities with constant unit norm $\|\dot{\mathbf{q}}\| = \dot{\mathbf{q}}^T \dot{\mathbf{q}} = 1$ which defines a point on the surface of an ellipsoid in the Cartesian velocity space. By substituting in (18),

$$\dot{\mathbf{q}}^T \dot{\mathbf{q}} = \dot{\mathbf{s}}^T (\mathbf{J}^+)^T \mathbf{J}^+ \dot{\mathbf{s}} = \dot{\mathbf{s}}^T (\mathbf{J} \mathbf{J}^T)^+ \dot{\mathbf{s}}, \quad (19)$$

The manipulability ellipsoid $\Upsilon = (\mathbf{J} \mathbf{J}^T)^+$, where $\Upsilon \in \mathcal{S}_{++}^m$ provides an intuitive indicator of the direction of possible movement of a configuration.

In this simulation we propose to learn manipulability ellipsoids, using the proposed SPD-based DMP, from one demonstration (from a teaching 3-DoF robot) and transfer it to a learner 5-DoF robot.

Here we propose to use the toy example proposed by [14] to evaluate our SPD-based DMP approach. One demonstration $\Xi = \{t_k, \Upsilon_k\}_{k=1}^T$ is collected by executing a tracking task by 3-DoF robot. Define \mathbf{s} as the Cartesian position of the robot end-effector. During reproduction, a 5-DoF robot tracks the desired Cartesian position $\hat{\mathbf{s}}$. The robot estimates its desired force \mathbf{F} at the end-effector by obeying the following controller [14],

$$\tau_d = \mathbf{J}^T \mathbf{F} - (\mathbf{I} - \mathbf{J}^T \mathbf{J}^T) \alpha \nabla g_t(\mathbf{q}); \quad \alpha > 0 \quad (20)$$

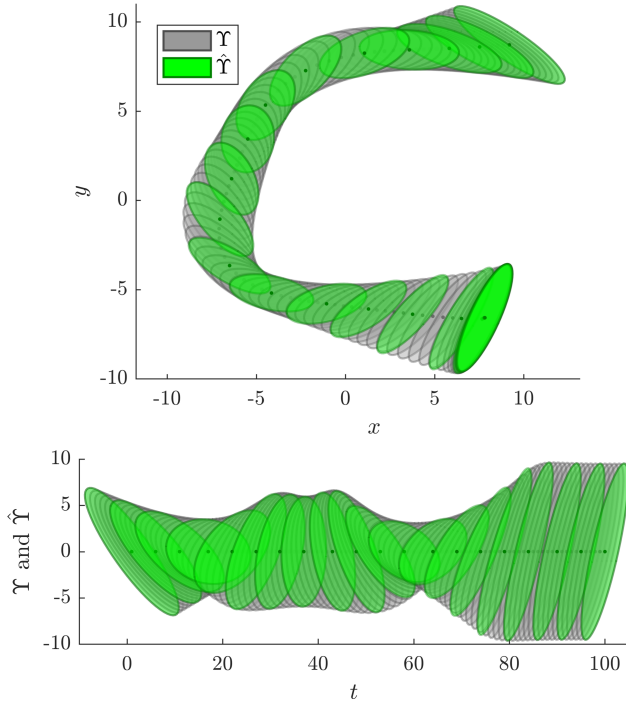


Fig. 5. *Top*: Executed Cartesian trajectory (black dots) given in centimeters, the demonstration (in gray) and the reproduced (in green) manipulability ellipsoids by the proposed DMPs, at different time steps of the task. *Bottom*: Both demonstration and resulting manipulability ellipsoids over time.

where $\bar{\mathbf{J}}$ is the inertia-weighted pseudo-inverse of \mathbf{J} , τ_d the desired joint torque, and the cost function $g_t(\mathbf{q})$ is defined as in [14],

$$g_t(\mathbf{q}) = \log \left(\det \left(\frac{\hat{\mathbf{\Upsilon}}_t + \mathbf{\Upsilon}_{a,t}(\mathbf{q})}{2} \right) \right) - \frac{1}{2} \log \left(\det(\hat{\mathbf{\Upsilon}}_t \mathbf{\Upsilon}_{a,t}(\mathbf{q})) \right) \quad (21)$$

where $\hat{\mathbf{\Upsilon}}$ are the desired manipulability ellipsoids obtained from our SPD-based DMP approach. $\mathbf{\Upsilon}_a(\mathbf{q})$ are the actual manipulability ellipsoids of the current robot configuration.

Figure 5(*top*) shows the matching between the actual and desired manipulability ellipsoids while the 5-DoF robot was following a desired C-shape Cartesian position. The figure illustrates how the desired manipulability ellipsoids $\hat{\mathbf{\Upsilon}}$ (in green) are smoothly following the demonstrated ellipsoids $\mathbf{\Upsilon}$ (in gray). The *bottom* part of the figure shows the reproduced and the demonstrated manipulability ellipsoids over time. Despite the fact that this manipulability profile is more complex than the stiffness one in the previous simulation (section IV-A), the reproduced profile still follows the demonstrated one. The accuracy of this reproduction can be seen in Fig. 6, where log-euclidean distance is used to calculate the difference between the result and the demonstration. The accuracy is high as also illustrated in Figure 7 which shows excellent matching between the demonstrated and the reproduced manipulability profiles in the cone of the SPD manifold.

The 1st-time derivative of the manipulability test is shown in Fig 8. This figure shows the convergence of the algorithm

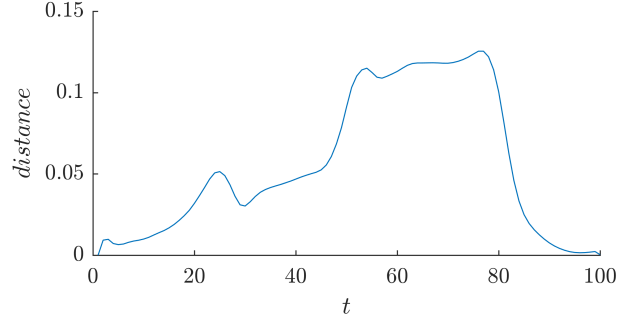


Fig. 6. The log-Euclidean distance between the resulting manipulability profile from the proposed DMPs and the ground truth using different metrics.

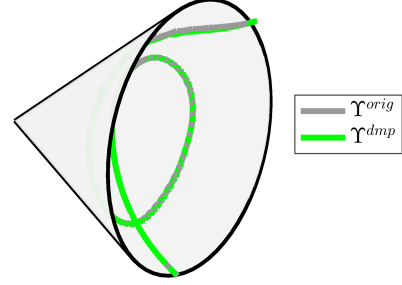


Fig. 7. Representation of the demonstration (in gray) and reproduced (in green) manipulability ellipsoids in the cone of SPD manifold.

to a unique attractor point at $\hat{\xi} = \mathbf{g}$ and $\hat{\sigma} = 0$.

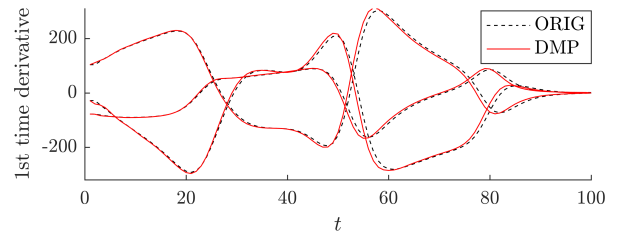


Fig. 8. The 1st-time derivative generated by the the proposed DMPs for learning manipulability ellipsoids.

D. Comparison with GMM/GMR

In this scope we compare the proposed SPD-based DMP with SPD-based GMM/GMR proposed by [16]. For fair comparison, as DMP is trained using one demonstration, we used also this same one demonstration to train GMM. The demonstration is taken from the previous manipulability simulation (section IV-C). As number of Gaussian components influence the accuracy of GMM/GMR, we trained 1-, 4-, 7-, and 10-states GMMs.

For each GMM model, we calculated the distance error between the manipulability profile obtained by GMR and the demonstration. Moreover, the distance error also has been calculated in the case of the proposed SPD-based DMP. Figure 9 shows the resulting distance error in all cases. From the figure, it is clear that the accuracy of GMM/GMR increases when the number of Gaussian components increases. However, increasing Gaussian components leads to a significant increase in the computation time as shown in Table II, while the proposed SPD-based DMP is significantly

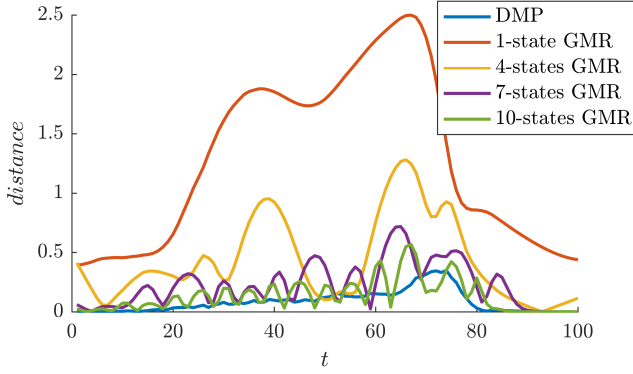


Fig. 9. The log-Euclidean distance between the resulting manipulability profile from the proposed DMPs and the ground truth using different metrics.

faster. Moreover, the GMM/GMR approach would not allow e.g. goal switching.

TABLE II

COMPARISON OF THE COMPUTATIONAL COST BETWEEN THE PROPOSED DMP AND GMM/GMR [16]

	Time in seconds	
	Learning	Reproduction
1-state GMM/GMR	2.8354	1.0788
4-states GMM/GMR	9.9459	3.9176
7-states GMM/GMR	16.3209	6.1992
10-states GMM/GMR	22.2517	8.3852
Proposed SPD-based DMP	0.0941	0.4214

V. CONCLUSIONS

In this paper we successfully exploited the Riemannian manifold of \mathcal{S}_{++}^m to derive a new formulation of DMPs capable of direct learning and reproduction of SPD-matrix-based robot skills. This formulation avoids any prior reparametrization of such skills. Moreover, we integrated a new formulation for the goal switching that can deal directly with SPD-matrix-based robot skills.

The algorithm has been extensively validated through multiple simulation examples. Moreover, a comparison with GMM/GMR demonstrates that the proposed approach provides at least similar accuracy with a significantly lower computation cost.

In the future we propose to integrate our approach with other algorithms, e.g. iterative learning control, in order to not just reproduce SPD-matrix-based skills, but also to adapt to different situations and perform more complex tasks (e.g. forced-based variable impedance control). Moreover, we will work on exploration-based learning methods, which will prove to be crucial when a robot needs to significantly adapt to a new situation, e.g. adapt its stiffness, in order to perform successfully in a large diversity of task situations.

ACKNOWLEDGMENT

This work is supported by CHIST-ERA project IPALM (Academy of Finland decision 326304).

REFERENCES

[1] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in Cognitive Sciences*, vol. 3, no. 6, pp. 233–242, 1999.

[2] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural Computation*, vol. 25, no. 2, pp. 328–373, 2013.

[3] F. J. Abu-Dakka, B. Nemec, J. A. Jørgensen, T. R. Savarimuthu, N. Krüger, and A. Ude, "Adaptation of manipulation skills in physical contact with the environment to reference force profiles," *Autonomous Robots*, vol. 39, no. 2, pp. 199–217, 2015.

[4] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann, "Probabilistic movement primitives," in *Advances in Neural Information Processing Systems*, 2013, pp. 2616–2624.

[5] S. Calinon, D. Bruno, and D. G. Caldwell, "A task-parameterized probabilistic model with minimal intervention control," in *IEEE International Conference on Robotics and Automation*, 2014, pp. 3339–3344.

[6] Y. Huang, L. Rozo, J. Silvério, and D. G. Caldwell, "Kernelized movement primitives," *The International Journal of Robotics Research*, vol. 38, no. 7, pp. 833–852, 2019.

[7] Y. Huang, F. J. Abu-Dakka, J. Silvério, and D. G. Caldwell, "Generalized orientation learning in robot task space," in *IEEE International Conference on Robotics and Automation*, 2019, pp. 2531–2537.

[8] R. Ikeura and H. Inooka, "Variable impedance control of a robot for cooperation with a human," in *IEEE International Conference on Robotics and Automation*, vol. 3, Nagoya, Japan, 1995, pp. 3097–3102.

[9] T. Tsumugiwa, R. Yokogawa, and K. Hara, "Variable impedance control based on estimation of human arm stiffness for human-robot cooperative calligraphic task," in *IEEE International Conference on Robotics and Automation*, vol. 1, Washington, DC, USA, 2002, pp. 644–650.

[10] F. J. Abu-Dakka, L. Rozo, and D. G. Caldwell, "Force-based variable impedance learning for robotic manipulation," *Robotics and Autonomous Systems*, vol. 109, pp. 156–167, 2018.

[11] L. Guilamo, J. Kuffner, K. Nishiwaki, and S. Kagami, "Manipulability optimization for trajectory generation," in *IEEE International Conference on Robotics and Automation*, 2006, pp. 2017–2022.

[12] N. Vahrenkamp, T. Asfour, G. Metta, G. Sandini, and R. Dillmann, "Manipulability analysis," in *12th IEEE/RAS international conference on humanoid robots (humanoids)*, 2012, pp. 568–573.

[13] I. Lee and J.-H. Oh, "Humanoid posture selection for reaching motion and a cooperative balancing controller," *Journal of Intelligent & Robotic Systems*, vol. 81, no. 3-4, pp. 301–316, 2016.

[14] L. Rozo, N. Jaquier, S. Calinon, and D. G. Caldwell, "Learning manipulability ellipsoids for task compatibility in robot manipulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vancouver, Canada, 2017, pp. 3183–3189.

[15] A. Ude, B. Nemec, T. Petrić, and J. Morimoto, "Orientation in cartesian space dynamic movement primitives," in *2014 IEEE International Conference on Robotics and Automation*. IEEE, 2014, pp. 2997–3004.

[16] N. Jaquier and S. Calinon, "Gaussian mixture regression on symmetric positive definite matrices manifolds: Application to wrist motion estimation with semg," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vancouver, Canada, 2017, pp. 59–64.

[17] S. Schaal, P. Mohajerian, and A. Ijspeert, "Dynamics systems vs. optimal control—a unifying view," *Progress in brain research*, vol. 165, pp. 425–445, 2007.

[18] X. Pennec, P. Fillard, and N. Ayache, "A riemannian framework for tensor computing," *International Journal of Computer Vision*, vol. 66, no. 1, pp. 41–66, 2006.

[19] S. Sra and R. Hosseini, "Conic geometric optimization on the manifold of positive definite matrices," *SIAM Journal on Optimization*, vol. 25, no. 1, pp. 713–739, 2015.

[20] O. Yair, M. Ben-Chen, and R. Talmon, "Parallel transport on the cone manifold of spd matrices for domain adaptation," *IEEE Transactions on Signal Processing*, vol. 67, no. 7, pp. 1797–1811, 2019.

[21] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Log-euclidean metrics for fast and simple calculus on diffusion tensors," *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, vol. 56, no. 2, pp. 411–421, 2006.

[22] A. Cherian, S. Sra, A. Banerjee, and N. Papanikolopoulos, "Efficient similarity search for covariance matrices via the jensen-bregman logdet divergence," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 2399–2406.