2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)
Beijing, China, November 6-9, 2018

Force-based Learning of Variable Impedance Skills for Robotic Manipulation

Fares J. Abu-Dakka, Leonel Rozo, and Darwin G. Caldwell

***Abstract*— Numerous robotics tasks involve complex physical interactions with the environment, where the role of variable impedance skills and the information of contact forces are crucial for successful performance. The dynamicity of our environments demands robots to adapt their manipulation skills to a large variety of situations, where learning capabilities are necessary. In this context, we propose a framework to teach a robot to perform manipulation tasks by integrating force sensing and variable impedance control. This framework endows robots with force-based variable stiffness skills that become relevant when vision information is unavailable or uninformative. Such skills are built on stiffness estimations that are computed from human demonstrations, which are then used along with sensed forces, to encode a probabilistic model of the robot skill. The resulting model is later used to retrieve time-varying stiffness profiles. We study two different stiffness representations based on *(i)* Cholesky decomposition, and *(ii)* Riemannian manifolds. For validation, we use a simulation of a 2D mass-spring-damper system subject to external forces, and a real experiment where a 7-DoF robot learns to perform a valve-turning task by varying its Cartesian stiffness.**

I. INTRODUCTION

Day by day robotic applications are bringing robots into unstructured environments (e.g., houses, hospitals, museums, etc.) where they are expected to perform complex manipulation tasks that are hard to program. This difficulty arises mainly because the environments are dynamic, uncertain, and therefore unpredictable, making hard-coding an unfeasible approach. In such environments, the robot may also need to physically interact with tools, humans, or other robots, where compliant motions are imperative for safe interactions, further complicating robot programming. In contrast, humans can efficiently perform different types of tasks in such environments, even those requiring compliance control, because of their outstanding adaptation capabilities and ability to change the arm stiffness according to the task requirements.

Therefore, human expertise may be exploited in a robot learning approach, where a robot learns manipulation tasks that require variable impedance skills from human demonstrations. In this vein, learning from demonstration (LfD) [1] is a user-friendly and intuitive methodology for a human to teach a new task to a robot, where task-relevant information is extracted from several demonstrations. Standard LfD approaches have mainly focused on trajectory-following tasks, however, there exist recent works aimed at extending such learning paradigm to force and impedance control [2], [3].

Department of Advanced Robotics, Istituto Italiano di Tecnologia, 16163 Genova, Italy (e-mail: {fares.abudakka, leonel.rozo, darwin.caldwell}@iit.it)

Fig. 1: *Left*: a human operator teaches a robot how to perform a valve-turning task. *Right*: a snapshot showing the valve-turning task reproduced by a KUKA LWR robot.

In this paper, we propose to exploit LfD to learn manipulation tasks that require stiffness levels that vary according to the state of the task. We also take advantage of the robot force-based perceptions, which usually convey relevant information in robotic manipulation. Specifically, the proposed learning framework implements kinesthetic teaching (Fig. 1-*left*) to collect demonstrations of a manipulation task, where both kinematic and dynamic data are recorded. The demonstration data along with a model of a mass-spring-damper (MSD) system are used to estimate time-varying Cartesian stiffness matrices (Sec. III-A). These are computed as the closest symmetric positive semi-definite matrix (SPSD) of an initial least-squares estimation (Sec. III-C). These estimates are then used as desired stiffness matrices for corresponding force patterns observed during the demonstrations. Both stiffness matrices and forces are later probabilistically encoded using a Gaussian mixture model (GMM). This model is exploited during reproduction to retrieve variable stiffness profiles by Gaussian mixture regression (GMR) (Sec. IV and V). The main contribution of our LfD framework lies on the exploitation of force sensing and variable impedance control (VIC) to learn and reproduce manipulation tasks requiring time-varying full stiffness matrices.[1] Beyond compliant movements and corresponding safety aspects, variable impedance skills allow the robot to perform a task using a larger range of control responses when compared to standard control methods. Moreover, we provide a different encoding perspective where stiffness matrices are directly modeled by our learning framework, which imposes new mathematical constraints due to the symmetric positive semi-definiteness of Cartesian stiffnesses.

The proposed force-based learning framework is extensively validated in simulation using the model of a 2D MSD system subjected to external forces. Moreover, a valve-

[1] We refer to variable impedance learning as the process of estimating and reproducing variable stiffness profiles. Nevertheless, the proposed approach can also be used to learn variable damping controllers.

turning task is used as a real scenario to further evaluate the performance of the proposed approach, as shown in Section VI. Conclusions and future research are given in Section VII.

II. Related Work

Traditionally, robot learning has focused on trajectory following tasks [4]. However, the new generation of torque-controlled robots has made it possible to extend learning capabilities to tasks that require variable impedance skills [3], allowing robots to exhibit compliant motions when physically interacting with the environment. This new research trend exploits impedance control [5], where the controller resembles a virtual MSD system between the environment and robot end-effector. This control approach allows robots to interact with the environment or humans more safely and in an energy-efficient way.

Previous works have been devoted to understand how impedance is modulated when humans interact with the environment [6] or to transfer human's impedance-based skills to robots [7]. However, robot learning capabilities were not developed to automatically vary impedance controller parameters to satisfactorily adapt in face of unseen situations.

Robot learning approaches have recently gained great interest for modeling variable impedance skills. Peternel *et al.* [8] proposed a single muscle impedance control interface for a compliance learning framework. Such an approach required prior knowledge on human anatomy and long calibration time. The authors extended their work by introducing a 3-DoF force feedback at the operator's end-effector [9]. Peternel *et al.* used human impedance for learning, while our approach estimates the robot stiffness from the task dynamics observed across multiple demonstrations.

Variable impedance also plays an important role in human-robot collaboration. In [10], a VIC is introduced based on the human arm stiffness estimation. They varied the virtual damping coefficient of the robot as a function of the estimated stiffness of the end-point human arm, and differential changes in position and force. Rozo *et al.* [11] proposed a framework to learn stiffness in a collaborative assembly task based on visual and haptic information. The estimated stiffness relied on a task-parameterized GMM, where each Gaussian component was assigned an independent stiffness matrix. The estimations were obtained via weighted least-squares (WLS) and the Frobenius norm. Recently, the authors reformulated their stiffness estimation method as a convex optimization problem [3]. Moreover, the robot movements were governed by time-driven attractor trajectories, which demanded to manually specify an initial stiffness to compute the attractor dynamics. Instead, our work focuses on force-driven skills learned by a probabilistic model that directly encodes full stiffness matrices estimated from the task dynamics, without requiring an initial approximation of the robot stiffness nor computation of attractor trajectories.

Manipulation tasks involve contact with the environment, where force sensing plays a crucial role for successful robot performance. In this context, force-based skills have been recently addressed from a robot learning perspective in [11], [12], [13], [14]. Kormushev *et al.* [13] encoded position and force data into a time-driven GMM to later retrieve a set of attractors in Cartesian space through least-squares regression. Stiffness matrices were estimated using the residuals terms of the regression process. In [12], kinesthetic demonstrations were used to teach haptic-based stiffness variations to a robot. Their approach retrieved full stiffness matrices via GMR, which used a GMM that encoded the robot Cartesian position and the Cholesky vector of the stiffness matrix. During reproduction, the teacher wiggled or applied strong pressures to produce haptic data aimed at varying the robot stiffness. This on-the-fly intervention may not be suitable in tasks that demand (delicate) physical contact with the environment. Also, the stiffness estimates required a manually-designed initial matrix. On the contrary, we estimate the stiffness directly from the task dynamics and interaction forces observed across demonstrations without interfering the task execution nor predefining an initial stiffness. In addition, driving the learning and reproduction phases by force perceptions, endows the robot with the ability to adapt its stiffness profile to different situations described by unseen sensed forces (e.g. unseen friction forces in the valve-turning task). Note that the demonstrated task dynamics was exploited in grasping [14] to estimate object-level stiffness matrices as a constrained optimization problem.

Rey *et al.* [15] used Path Integrals (PI^2) policy search for time-invariant policies to learn variable stiffness and motion simultaneously. Their work was built on [16] where PI^2 was used with dynamic movement primitives to learn stiffness matrices through a set of basis functions that linearly depended on a time-driven policy parameter optimized by PI^2. Rey *et al.* regressed the policy parameters via GMR, which allowed for state-dependent policies. However, iterative learning methods can be time consuming in some applications and impractical in others. Therefore, we focus our work on a learning approach where a model of the task is learned from human examples, which may also be combined with the aforementioned iterative approaches if safe exploration strategies are considered. Moreover, in contrast to our approach, the foregoing work estimated a diagonal stiffness matrix profile and did not exploit force-based perceptions to reproduce the task.

III. Proposed Approach

A successful LfD should include both a learning framework that encodes the raw data as well as an appropriate model that encapsulates the dynamics of the desired behavior. In this line, we exploit a linear dynamical model to estimate local full stiffness matrices, which are then encoded along with force data by a probabilistic model. More specifically, during the demonstration of the task, we record the robot Cartesian position $\mathbf{x}$, its velocity $\dot{\mathbf{x}}$, acceleration $\ddot{\mathbf{x}}$, and the associated sensed Cartesian forces $\mathbf{f}^e$ with respect to the robot frame. Using the collected data, we first estimate the stiffness required to locally fit the dynamics of a virtual MSD system as explained next. After, a GMM is trained to encode both force and stiffness data. The model is later used by

TABLE I: Phases of the proposed approach

1. Task demonstrations
- Performing $N \times k$ recordings for a task.
  k demonstrations in N different situations.

2. Stiffness estimation
- Estimate $\tilde{\mathbf{K}}^{\mathcal{P}}$ using (3) and (4) (Section III-B).
- Compose dataset $\{\bar{\mathbf{f}}^{e}_{t,n}, \tilde{\mathbf{K}}^{\mathcal{P}}_{t,n}\}_{n=1}^{N}$ for learning (Section III-C).

2. Model fitting (Sections IV and V)
- Learn model parameters $\boldsymbol{\Theta}$ using (5) or (12).

4. Reproduction (Sections IV and V)
- Set the input $\mathcal{I}$ and output $\mathcal{O}$ elements.

for $t \leftarrow 1$ **to** T (for each reproduction time step)
- Given an input $\mathcal{I}$, estimate the output $\mathcal{O}$ via GMR using (7) or (18).
- Compute Cartesian force command $\mathbf{F}$ at the robot end-effector using (2) under the *unitary-mass* assumption.
- Compute desired joint torques using (24)

end

GMR to retrieve continuous stiffness profiles given unseen force trajectories during the reproduction of the task.

A. Interaction model

Using impedance control helps overcome environment uncertainties and avoids large impact forces, since robots are controlled to modulate their motion/compliance according to sensed forces arising from the interaction between a robot and the environment. In this work, we model the robot end-effector as a *unitary-mass* $\mathbf{I}_M$ affected by two forces: control input $\mathbf{f}^c$ and external/sensed forces $\mathbf{f}^e$, which leads to the simple dynamic model [17]

$$\mathbf{I}_M \ddot{\mathbf{x}} = \mathbf{f}^c + \mathbf{f}^e. \tag{1}$$

The dynamics governing the robot behavior is modeled as a virtual MSD system as follows

$$\mathbf{I}_M \ddot{\mathbf{x}}_t = \mathbf{K}^{\mathcal{P}}_t(\hat{\mathbf{x}} - \mathbf{x}_t) - \mathbf{K}^{\mathcal{V}}_t \dot{\mathbf{x}}_t + \mathbf{f}^e_t, \tag{2}$$

where $\mathbf{K}^{\mathcal{P}}_t$ and $\mathbf{K}^{\mathcal{V}}_t$ are full stiffness and damping matrices, while $\hat{\mathbf{x}}$ represents the equilibrium position (or desired trajectory) of the virtual system.

B. Stiffness estimation from demonstrations

The full stiffness matrix $\mathbf{K}^{\mathcal{P}}$ can be estimated from demonstrations using different approaches. For example, GMM was used in [3] to cluster the demonstrated data, and then WLS was applied to estimate a $\mathbf{K}^{\mathcal{P}}_i$ associated with each Gaussian component i using convex optimization. This approach required several iterations for convergence, which increases the computational cost of the estimation process. In [18], $\mathbf{K}^{\mathcal{P}}$ is estimated from the inverse of the observed position covariance encapsulated in a GMM. Their approach is limited to tasks displaying variability in position trajectories across demonstrations, which does not arise in scenarios where the end-effector is constrained to follow a single Cartesian path (e.g., valve-turning tasks). Unlike [3], [18], we propose to use a sliding window technique to carry out local stiffness estimations that fulfill (2) at each time step t. This method is only parametrized by the window length L. Note that the use of clustering techniques such as GMM may demand to use a high number of Gaussian components

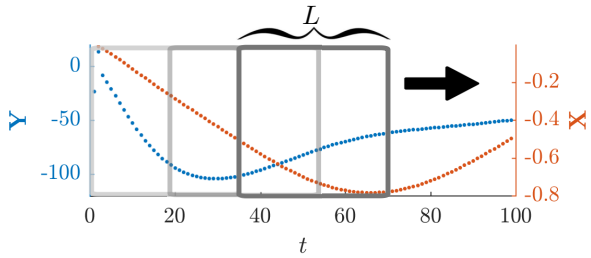

Fig. 2: Sliding window approach. A rectangular window with a fixed width L slides across the data over time. For each of these windows, $\mathbf{K}^{\mathcal{P}}$ is estimated through (3). The arrow indicates the sliding direction.

for highly-nonlinear dynamic tasks, significantly increasing the number of parameters for the estimation process.

Specifically, in our estimation method, a window of length L moves all along the demonstration data $\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}$, and $\mathbf{f}^e$, and the full stiffness matrix is estimated over the data covered by the window for each t. Formally, from (2), let us define $\{\tilde{\mathbf{x}} = \hat{\mathbf{x}} - \mathbf{x}\}_t$ and $\{\tilde{\mathbf{y}} = \mathbf{I}_M \ddot{\mathbf{x}} + \mathbf{K}^{\mathcal{V}} \dot{\mathbf{x}} - \mathbf{f}^e\}_t$ for all the datapoints in the window, for all demonstrations. Then, by concatenating the resulting data, matrices $\mathbf{X} = \tilde{\mathbf{x}}_1 || \tilde{\mathbf{x}}_2 || \dots || \tilde{\mathbf{x}}_k$ and $\mathbf{Y} = \tilde{\mathbf{y}}_1 || \tilde{\mathbf{y}}_2 || \dots || \tilde{\mathbf{y}}_k$ are defined (Fig. 2). The stiffness matrix estimation is carried out at each t by solving the following linear system using least-squares

$$\mathbf{X}\mathbf{K}^{\mathcal{P}} = \mathbf{Y}. \tag{3}$$

However, a stiffness matrix $\mathbf{K}^{\mathcal{P}} \in \boldsymbol{\mathcal{S}}^m_+$ is symmetric and positive definite (SPD), which means that the estimation computed by (3) is just a rough approximation that does not fulfill the SPD constraints. Inspired by [11], we resort to the approximation formulated in [19] to compute the nearest SPD matrix from the approximation in (3), as follows,

$$\tilde{\mathbf{K}}^{\mathcal{P}} = \frac{\mathbf{B} + \mathbf{H}}{2}, \quad \mathbf{B} = \frac{\mathbf{K}^{\mathcal{P}} + \left(\mathbf{K}^{\mathcal{P}}\right)^{\mathsf{T}}}{2}, \quad \mathbf{H} = \mathbf{V}\mathbf{S}\mathbf{V}^{\mathsf{T}}, \tag{4}$$

where $\mathbf{H}$ is the symmetric polar factor of $\mathbf{B} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathsf{T}}$.

Note that some of the approximated matrices might be on the boundary of the SPD matrices space, resulting in symmetric positive semi-definite matrices. In this case, each eigenvalue $\lambda_i \approx 0$ of $\tilde{\mathbf{K}}^{\mathcal{P}}$ is constrained to have a minimum value ϵ, and then $\tilde{\mathbf{K}}^{\mathcal{P}}$ is reconstructed using its eigendecomposition $\tilde{\mathbf{K}}^{\mathcal{P}} = \boldsymbol{\mathcal{Q}}\boldsymbol{\Lambda}\boldsymbol{\mathcal{Q}}^{-1}$. The damping term in (2) can be chosen either experimentally or using the eigendecomposition of $\tilde{\mathbf{K}}^{\mathcal{P}}$ to keep the system critically damped, e.g. $\mathbf{K}^{\mathcal{V}}_t = \boldsymbol{\mathcal{Q}}(\zeta \boldsymbol{\Lambda}^{\frac{1}{2}})\boldsymbol{\mathcal{Q}}^{\mathsf{T}}$, where $\boldsymbol{\mathcal{Q}}$ is the eigenvectors and $\boldsymbol{\Lambda}$ is the eigenvalues of $\tilde{\mathbf{K}}^{\mathcal{P}}$, and $\zeta \in \mathbb{R}^+$ is a tuning parameter.

C. Stiffness learning

Some manipulation tasks heavily depend on force perceptions, which play a relevant role when vision information is unavailable or uninformative. We propose an approach that learns and reproduces variable impedance skills based on interaction forces. In this paper, given the estimated $\tilde{\mathbf{K}}^{\mathcal{P}}$ and sensed forces $\mathbf{f}^e$, we introduce the following two methods to learn force-based variable impedance skills:

1. Learn $\tilde{\mathbf{K}}^{\mathcal{P}}$ using GMM/GMR for Euclidean data. Here, we represent $\tilde{\mathbf{K}}^{\mathcal{P}}$ by its Cholesky vector $\mathbf{v}^{\mathcal{P}}$ obtained by vectorizing the upper triangle factor $\mathcal{K}^{\mathcal{P}}$ in $\tilde{\mathbf{K}}^{\mathcal{P}} = \mathcal{K}^{\mathcal{P}\mathsf{T}}\mathcal{K}^{\mathcal{P}}$, [12]. Using this representation we reduce the data space dimensionality to $m + m(m-1)/2$ instead of vectorizing the whole matrix $\tilde{\mathbf{K}}^{\mathcal{P}}$, which replicates stiffness information due to its symmetry.
2. Use a tensor-based formulation of GMM/GMR on the Riemannian manifold $\mathcal{S}_+^m$ [20], [21] to directly learn $\tilde{\mathbf{K}}^{\mathcal{P}}$, which avoids the reparametrization of the above approach and exploits the geometry of the SPD space.

In the proposed framework, we exploit (3) and (4) to estimate the desired time-varying stiffness $\tilde{\mathbf{K}}^{\mathcal{P}}$ using the recorded data $\{\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}, \mathbf{f}^e\}$ of a subset of k demonstrations given under the same task situation T_n, with $n = 1, ..., N$, and where N represents the number of different task instances. Each task situation T_n is characterized by different task conditions, which basically affect the task dynamics and interaction forces. Therefore, for N different situations, N different time-varying stiffness profiles are estimated, leading to the training dataset $\{\bar{\mathbf{f}}^e_{t,n}, \tilde{\mathbf{K}}^{\mathcal{P}}_{t,n}\}_{n=1}^N$, where $\bar{\mathbf{f}}^e_t = \frac{1}{k}\left(\sum_{s=1}^k \mathbf{f}^e_{t,s}\right)$ is the mean sensed force across the subset of k demonstrations at each t. Table I summarizes our approach and its different stages.

IV. GMM/GMR on Cholesky Vector

First, let us denote $\mathbf{A}$ as any arbitrary SPD matrix and $\mathbf{L}$ its Cholesky factor, so that $\mathbf{A} = \mathbf{L}^{\mathsf{T}}\mathbf{L}$. This representation is here exploited to learn a joint probability distribution of the vectorization of the upper triangle matrix $\mathbf{L}$ and other data vector(s). Also, let us define the input vector $\boldsymbol{\xi}_t^{\mathcal{I}}$ and output vector $\boldsymbol{\xi}_t^{\mathcal{O}}$, which form a demonstration datapoint $\boldsymbol{\xi}_t = \{\boldsymbol{\xi}_t^{\mathcal{I}}, \boldsymbol{\xi}_t^{\mathcal{O}}\}$. The training dataset $\{\{\boldsymbol{\xi}_t\} = \{\boldsymbol{\xi}_t^{\mathcal{I}}, \boldsymbol{\xi}_t^{\mathcal{O}}\}\}_{t=1}^T$ is assumed to be normally distributed and modeled by a mixture of G Gaussians

$$\mathcal{P}\left(\boldsymbol{\xi}_t^{\mathcal{I}}, \boldsymbol{\xi}_t^{\mathcal{O}}\right) = \sum_{l=1}^{G} \boldsymbol{\pi}_l \mathcal{N}\big(\boldsymbol{\xi}_t; \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l\big), \tag{5}$$

where $\boldsymbol{\Theta} = \{\boldsymbol{\pi}_l, \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l\}_{l=1}^G$ are the prior, mean, and covariance parameters. The superindexes $\mathcal{I}$ and $\mathcal{O}$ respectively indicate input and output. Maximum-likelihood estimation of the mixture parameters is iteratively performed using the standard Expectation-Maximization (EM) algorithm [4].

In literature, GMM is usually combined with GMR, providing an elegant solution for encoding and synthesizing motor skills in robotics [22]. GMR is used here to retrieve the vectorized versions of $\hat{\mathbf{L}}$, which are subsequently used to reconstruct a desired SPD matrix $\hat{\mathbf{A}}$. Formally, let us decompose the GMM parameters $\boldsymbol{\mu}_l$ and $\boldsymbol{\Sigma}_l$ as

$$\boldsymbol{\mu}_l = \begin{bmatrix} \boldsymbol{\mu}_l^{\mathcal{I}} \\ \boldsymbol{\mu}_l^{\mathcal{O}} \end{bmatrix}, \quad \boldsymbol{\Sigma}_l = \begin{bmatrix} \boldsymbol{\Sigma}_l^{\mathcal{I}} & \boldsymbol{\Sigma}_l^{\mathcal{I}\mathcal{O}} \\ \boldsymbol{\Sigma}_l^{\mathcal{O}\mathcal{I}} & \boldsymbol{\Sigma}_l^{\mathcal{O}} \end{bmatrix}. \tag{6}$$

During the reproduction of the task, the conditional distribution $\mathcal{P}(\boldsymbol{\xi}_t^{\mathcal{O}}|\boldsymbol{\xi}_t^{\mathcal{I}})$ is estimated as follows,

$$\mathcal{P}\left(\boldsymbol{\xi}_t^{\mathcal{O}}|\boldsymbol{\xi}_t^{\mathcal{I}}\right) = \mathcal{N}\Big(\boldsymbol{\xi}_t^{\mathcal{O}}; \hat{\boldsymbol{\mu}}_t^{\mathcal{O}}, \hat{\boldsymbol{\Sigma}}_t^{\mathcal{O}}\Big), \tag{7}$$

where

$$\begin{aligned}
\hat{\boldsymbol{\mu}}_t^{\mathcal{O}} &= \sum_{l=1}^{G} h_l(\boldsymbol{\xi}_t^{\mathcal{I}})\hat{\boldsymbol{\mu}}_l^{\mathcal{O}}(\boldsymbol{\xi}_t^{\mathcal{I}}), \qquad (8)\\
\hat{\boldsymbol{\mu}}_l^{\mathcal{O}}(\boldsymbol{\xi}_t^{\mathcal{I}}) &= \boldsymbol{\mu}_l^{\mathcal{O}} + \boldsymbol{\Sigma}_l^{\mathcal{O}\mathcal{I}}{\boldsymbol{\Sigma}_{t,l}^{\mathcal{I}}}^{-1}(\boldsymbol{\xi}_t^{\mathcal{I}} - \boldsymbol{\mu}_l^{\mathcal{I}}),\\
\hat{\boldsymbol{\Sigma}}_t^{\mathcal{O}} &= \sum_{l=1}^{G} h_l(\boldsymbol{\xi}_t^{\mathcal{I}})\Big(\hat{\boldsymbol{\Sigma}}_l^{\mathcal{O}} + \hat{\boldsymbol{\mu}}_l^{\mathcal{O}}(\boldsymbol{\xi}_t^{\mathcal{I}})\hat{\boldsymbol{\mu}}_l^{\mathcal{O}}(\boldsymbol{\xi}_t^{\mathcal{I}})^{\mathsf{T}}\Big) \qquad (9)\\
&\quad - \hat{\boldsymbol{\mu}}_t^{\mathcal{O}}\hat{\boldsymbol{\mu}}_t^{\mathcal{O}\mathsf{T}},\\
\hat{\boldsymbol{\Sigma}}_l^{\mathcal{O}} &= \boldsymbol{\Sigma}_l^{\mathcal{O}} - \boldsymbol{\Sigma}_l^{\mathcal{O}\mathcal{I}}{\boldsymbol{\Sigma}_l^{\mathcal{I}}}^{-1}\boldsymbol{\Sigma}_l^{\mathcal{I}\mathcal{O}}.
\end{aligned}$$

The upper triangle matrix $\hat{\mathbf{L}}$ of the Cholesky decomposition is reconstructed from the mean $\hat{\boldsymbol{\mu}}^{\mathcal{O}}$ and so the desired SPD matrix becomes $\hat{\mathbf{A}} = \hat{\mathbf{L}}^{\mathsf{T}}\hat{\mathbf{L}}$.

V. GMM/GMR on SPD Manifolds

As $\mathbf{K}^{\mathcal{P}} \in \mathcal{S}_+^m$, a geometry-aware approach that allows us to encode and retrieve SPD matrices is an important tool to learn variable impedance skills. In this context, the model introduced in [20] deals with the geometry of the SPD manifold and allows us to build joint and conditional probability distributions over it. This model is mainly built on the incorporation of a Riemannian metric, which allows the set of SPD matrices to form a Riemannian manifold [23]. Such metric defines the geodesics (a generalization of straight lines in Euclidean space), which is the minimum length curves between two points on the manifold.

A. Riemannian Manifold of SPD matrices

A Riemannian manifold $\mathcal{M}$ is a topological space in which each point locally resembles a Euclidean space. For each point $\mathbf{p} \in \mathcal{M}$, there exists a tangent space $\mathcal{T}_{\mathbf{p}}\mathcal{M}$ (endowed with a positive definite inner product), which corresponds to the space of symmetric matrices for the case of the SPD manifold. To switch between $\mathcal{T}_{\mathbf{p}}\mathcal{M}$ and $\mathcal{M}$, which is needed to compute statistics over Riemannian manifolds, three operations are required: exponential and logarithm maps, and the parallel transport.

The exponential map $\mathrm{Exp}_{\boldsymbol{\Gamma}}(\boldsymbol{\Delta}) : \mathcal{T}_{\boldsymbol{\Gamma}}\mathcal{M} \mapsto \mathcal{M}$ is a function that maps a point $\boldsymbol{\Delta}$ in the tangent space to a point $\mathbf{Q} \in \mathcal{M}$, so that it lies on the geodesic starting from $\boldsymbol{\Gamma} \in \mathcal{S}_+^m$ in the direction of $\boldsymbol{\Delta}$. On the other hand, the logarithm map $\mathrm{Log}_{\boldsymbol{\Gamma}}(\mathbf{Q}) : \mathcal{M} \mapsto \mathcal{T}_{\boldsymbol{\Gamma}}\mathcal{M}$ is the inverse of the exponential map. The aforementioned mappings are computed as [20]

$$\mathrm{Exp}_{\boldsymbol{\Gamma}}(\boldsymbol{\Delta}) = \boldsymbol{\Gamma}^{\frac{1}{2}} \exp(\boldsymbol{\Gamma}^{-\frac{1}{2}}\boldsymbol{\Delta}\boldsymbol{\Gamma}^{-\frac{1}{2}})\boldsymbol{\Gamma}^{\frac{1}{2}}, \tag{10}$$

$$\mathrm{Log}_{\boldsymbol{\Gamma}}(\mathbf{Q}) = \boldsymbol{\Gamma}^{\frac{1}{2}} \log(\boldsymbol{\Gamma}^{-\frac{1}{2}}\mathbf{Q}\boldsymbol{\Gamma}^{-\frac{1}{2}})\boldsymbol{\Gamma}^{\frac{1}{2}}. \tag{11}$$

Moreover, in order to move elements between tangent spaces while maintaining the angle between them constant, we need to use parallel transport $\mathbb{B}_{\boldsymbol{\Gamma}\mapsto\mathbf{Q}} : \mathcal{T}_{\boldsymbol{\Gamma}}\mathcal{M} \mapsto \mathcal{T}_{\mathbf{Q}}\mathcal{M}$. This is computed for a matrix $\mathbf{V}$ as $\mathbb{B}_{\boldsymbol{\Gamma}\mapsto\mathbf{Q}}(\mathbf{V}) = \mathbf{C}_{\boldsymbol{\Gamma}\mapsto\mathbf{Q}}\ \boldsymbol{\Gamma}\ \mathbf{C}_{\boldsymbol{\Gamma}\mapsto\mathbf{Q}}^{\mathsf{T}}$, where $\mathbf{C}_{\boldsymbol{\Gamma}\mapsto\mathbf{Q}} = \boldsymbol{\Gamma}^{\frac{1}{2}} \exp(\frac{1}{2}\boldsymbol{\Gamma}^{-\frac{1}{2}}\mathbf{V}\boldsymbol{\Gamma}^{-\frac{1}{2}})\boldsymbol{\Gamma}^{-\frac{1}{2}}$ [20].

B. GMM on SPD

In general terms, for a given random variable $\boldsymbol{\Xi} \in \mathcal{M}$, a GMM on the SPD manifold is defined by

$$\mathcal{P}(\boldsymbol{\Xi}) = \sum_{l=1}^{G} \pi_l \mathcal{N}_{\mathcal{M}}(\boldsymbol{\Xi}; \mathbf{M}_l, \boldsymbol{\mathcal{Y}}_l), \tag{12}$$

where G is the number of Gaussian components, $\mathbf{M} \in \mathcal{M}$ is the mean (and center of the tangent space $\mathcal{T}_{\mathbf{M}}$), and $\boldsymbol{\mathcal{Y}} \in \mathcal{T}_{\mathbf{M}}\mathcal{M}$ is the covariance tensor. The GMM parameters $\boldsymbol{\Theta} = \{\boldsymbol{\pi}_l, \mathbf{M}_l, \boldsymbol{\Xi}_l\}_{l=1}^{G}$ on the manifold are estimated using EM algorithm as follows,
E-step:

$$\mathcal{P}(l|\boldsymbol{\Xi}_i) = \frac{\pi_l \mathcal{N}_{\mathcal{M}}(\boldsymbol{\Xi}_i; \mathbf{M}_l, \boldsymbol{\mathcal{Y}}_l)}{\sum_{j=1}^{G} \pi_j \mathcal{N}_{\mathcal{M}}(\boldsymbol{\Xi}_i; \mathbf{M}_j, \boldsymbol{\mathcal{Y}}_j)}, \quad N_l = \sum_{i=1}^{N} \mathcal{P}(l|\boldsymbol{\Xi}_i). \tag{13}$$

M-step:

$$\mathbf{M}_l \leftarrow \frac{1}{N_l} \mathrm{Exp}_{\mathbf{M}_l}\Big(\sum_{i=1}^{N} \mathcal{P}(l|\boldsymbol{\Xi}_i) \mathrm{Log}_{\mathbf{M}_l}(\boldsymbol{\Xi}_i) \Big), \tag{14}$$

$$\boldsymbol{\mathcal{Y}}_l \leftarrow \frac{1}{N_l} \sum_{i=1}^{N} \mathcal{P}(l|\boldsymbol{\Xi}_i) \mathrm{Log}_{\mathbf{M}_l}(\boldsymbol{\Xi}_i) \otimes \mathrm{Log}_{\mathbf{M}_l}(\boldsymbol{\Xi}_i), \tag{15}$$

$$\pi_l \leftarrow \frac{N_l}{N}. \tag{16}$$

C. GMR on SPD

During the reproduction, the conditional distribution $\mathcal{P}(\boldsymbol{\Xi}_{\mathcal{OO}}|\boldsymbol{\Xi}_{\mathcal{II}})$ is computed via GMR. Formally, by defining inputs $\boldsymbol{\Xi}_{\mathcal{II}}$ and outputs $\boldsymbol{\Xi}_{\mathcal{OO}}$, a block decomposition of $\boldsymbol{\Xi}$, $\mathbf{M}$, and $\boldsymbol{\mathcal{Y}}$ is given as follows,

$$\boldsymbol{\Xi} = \begin{bmatrix} \boldsymbol{\Xi}_{\mathcal{II}} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Xi}_{\mathcal{OO}} \end{bmatrix}, \quad \mathbf{M} = \begin{bmatrix} \mathbf{M}_{\mathcal{II}} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{\mathcal{OO}} \end{bmatrix},$$
$$\boldsymbol{\mathcal{Y}} = \left[\begin{array}{cc|cc} \boldsymbol{\mathcal{Y}}^{\mathcal{II}}_{\mathcal{II}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mathcal{Y}}^{\mathcal{OO}}_{\mathcal{II}} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \boldsymbol{\mathcal{Y}}^{\mathcal{II}}_{\mathcal{OO}} & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\mathcal{Y}}^{\mathcal{OO}}_{\mathcal{OO}} \end{array}\right], \tag{17}$$

where the $4^{\text{th}}-$order tensor $\boldsymbol{\mathcal{Y}}$ is represented by separating the different fibers with bars. With this decomposition, manifold functions can be applied individually on input and output parts, e.g.

$$\mathrm{Exp}_{\mathbf{M}_l}(\boldsymbol{\Xi}) = \begin{bmatrix} \mathrm{Exp}_{\mathbf{M}_{\mathcal{II}}}(\boldsymbol{\Xi}_{\mathcal{II}}) & \mathbf{0} \\ \mathbf{0} & \mathrm{Exp}_{\mathbf{M}_{\mathcal{OO}}}(\boldsymbol{\Xi}_{\mathcal{OO}}) \end{bmatrix}.$$

GMR on an SPD manifold approximates the conditional distribution by a single Gaussian

$$\mathcal{P}(\boldsymbol{\Xi}_{\mathcal{OO}}|\boldsymbol{\Xi}_{\mathcal{II}}) \sim \mathcal{N}\left(\hat{\mathbf{M}}_{\mathcal{OO}}, \hat{\boldsymbol{\mathcal{Y}}}^{\mathcal{OO}}_{\mathcal{OO}}\right), \tag{18}$$

where $\hat{\mathbf{M}}_{\mathcal{OO}}$ is calculated iteratively until convergence with

$$\boldsymbol{\Upsilon}_l = \mathrm{Log}_{\hat{\mathbf{M}}_{\mathcal{OO}}}(\mathbf{M}_{\mathcal{OO},l}) - \tilde{\boldsymbol{\mathcal{Y}}}^{\mathcal{II}}_{\mathcal{OO},l} \tilde{\boldsymbol{\mathcal{Y}}}^{\mathcal{II}^{-1}}_{\mathcal{II},l} \mathrm{Log}_{\boldsymbol{\Xi}_{\mathcal{II}}}(\mathbf{M}_{\mathcal{II},l}), \tag{19}$$

$$\hat{\mathbf{M}}_{\mathcal{OO}} \leftarrow \mathrm{Exp}_{\hat{\mathbf{M}}_{\mathcal{OO}}}\Big(\sum_l h_l \boldsymbol{\Upsilon}_l \Big), \tag{20}$$

where

$$h_l = \frac{\pi_l \mathcal{N}(\boldsymbol{\Xi}_{\mathcal{II}}; \mathbf{M}_{\mathcal{II},l}, \boldsymbol{\mathcal{Y}}^{\mathcal{II}}_{\mathcal{II},l})}{\sum_{j=1}^{G} \pi_j \mathcal{N}(\boldsymbol{\Xi}_{\mathcal{II}}; \mathbf{M}_{\mathcal{II},j}, \boldsymbol{\mathcal{Y}}^{\mathcal{II}}_{\mathcal{II},j})} \tag{21}$$

describe the responsibilities of the GMM component l in the regression. The 4^{th}–order tensor covariance $\hat{\boldsymbol{\mathcal{Y}}}^{\mathcal{OO}}_{\mathcal{OO}}$ is calculated in the tangent space of the estimated mean as

$$\hat{\boldsymbol{\mathcal{Y}}}^{\mathcal{OO}}_{\mathcal{OO}} = \sum_l h_l \Big(\tilde{\boldsymbol{\mathcal{Y}}}^{\mathcal{OO}}_{\mathcal{OO},l} - \tilde{\boldsymbol{\mathcal{Y}}}^{\mathcal{II}}_{\mathcal{OO},l} \tilde{\boldsymbol{\mathcal{Y}}}^{\mathcal{II}^{-1}}_{\mathcal{II},l} \tilde{\boldsymbol{\mathcal{Y}}}^{\mathcal{OO}}_{\mathcal{II},l} + \boldsymbol{\Upsilon}_l \otimes \boldsymbol{\Upsilon}_l \Big), \tag{22}$$

where $\tilde{\boldsymbol{\mathcal{Y}}} = \mathbb{B}_{\mathbf{M} \mapsto \hat{\boldsymbol{\Xi}}}(\boldsymbol{\mathcal{Y}})$ and $\hat{\boldsymbol{\Xi}} = \begin{bmatrix} \boldsymbol{\Xi}_{\mathcal{II}} & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{M}}_{\mathcal{OO}} \end{bmatrix}$.

VI. Experiments

Our framework was evaluated in both simulation and a real scenario. The approaches were implemented in MATLAB® for simulation, and in object oriented C++ for real experiments. We used a computer running Ubuntu 16.04 LTS with Intel Core i7 – 6700 CPU 3.40 GHz, 16 GB of RAM. The experiments used in the evaluation are summarized below:

- Comparison between SPD approximation (4) and convex optimization [3] to estimate $\tilde{\mathbf{K}}^{\mathcal{P}}$
- Stiffness learning and reproduction with a 2-DoF MSD
- Comparison between Cholesky and Riemannian manifold representations.
- Real experiment on a 7-DoF KUKA-LWR robot equipped with a gripper to perform a valve-turning task

For the first two simulated experiments, manually-designed $\mathbf{f}^e$ profiles were used to compute the MSD dynamics $[\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}]$ at each t for specific stiffness and damping values. This stiffness is considered as our ground truth for comparison purposes. Different springs were distinguished by their stiffness values. For each spring, the stiffness estimation $\tilde{\mathbf{K}}^{\mathcal{P}}$ was obtained from five demonstrations (differing in the applied external forces) using a window length $L = 3$ and damping coefficient $\mathbf{K}^{\mathcal{V}}_t = \mathrm{diag}(50)$ $[N.s/m]$.[2]

A. Stiffness estimation

As described in Section III-B, $\tilde{\mathbf{K}}^{\mathcal{P}}$ can be estimated either using (3) and (4), or alternatively formulating the estimation problem as a convex optimization [3]. In the latter approach, the objective is to minimize the Euclidean norm of the residuals, subject to the matrix inequality constraint introduced by the positive semi-definiteness of $\mathbf{K}^{\mathcal{P}}$,

$$\text{minimize} \|\mathbf{X}\mathbf{K}^{\mathcal{P}} - \mathbf{Y}\|_2 \quad \text{subject to } \mathbf{K}^{\mathcal{P}} \succeq \mathbf{0}. \tag{23}$$

Figures 3 and 4 show a comparison between the ground truth stiffness ellipsoids and the estimations obtained from the nearest-SPD approximation (4) and convex optimization (23). We used ten MSD systems of different time-varying stiffness. During the MSD simulation, the stiffness matrices were initialized from an horizontally-aligned ellipsoid, which was then continuously rotated clockwise until reaching a 45

[2]Parameters such as L, G, ϵ, and number of demonstrations were experimentally chosen.

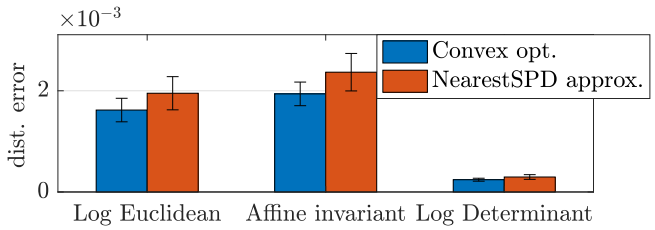

Fig. 3: Different distance-based errors of the estimated stiffness using nearestSPD approximation (4) and convex optimization (23).

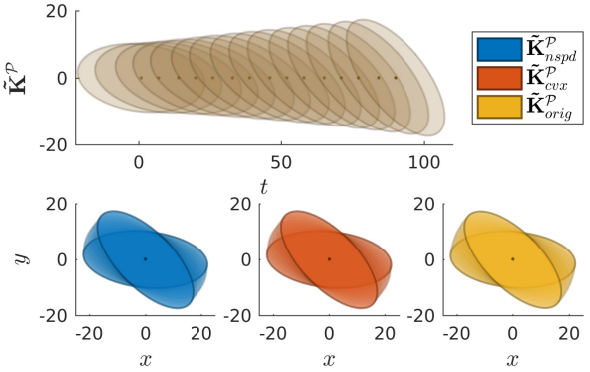

Fig. 4: *Top*: stiffness ellipsoids of the ground truth at different time steps matching the nearestSPD and convex optimization estimations. *Bottom*: 2D plot showing the hand-coded 45 degrees rotation of the stiffness ellipsoid. Stiffness ellipsoids are centered at 0 for visualization purposes. Negative values in the vertical axis do not imply that the matrices are not SPD.

degrees rotation (see Fig. 4). The average computation time for the nearest-SPD approximation and convex optimization were 0.05 and 358 ms per time step, respectively.

Figure 3 shows the average distances between the ground truth stiffness and the estimations computed from the nearest-SPD approximation and convex optimization. The convex optimization method provides slightly better estimates, but significantly compromises the computational cost of the estimation process when compared to the nearest-SPD approximation. Note that in order to compute the estimation error on $\mathcal{S}^m_+$, different metrics were used, namely: (1) the affine-invariant, (2) the log-Euclidean, and (3) Jensen-Bregman log-determinant (for more details about these metrics, see [24]).

B. Force-based variable impedance for MSD system

Here, we use the MSD simulation to evaluate the approaches described in Sections IV and V for learning force-based variable impedance. The MSD started from a rest position with initial stiffness $\mathbf{K}^{\mathcal{P}}$ initialized as a horizontally-aligned ellipsoid. During the simulation, external forces $\mathbf{f}^e$ were applied to stimulate the system dynamics, while $\mathbf{K}^{\mathcal{P}}$ was continuously rotated until reaching a vertically-aligned ellipsoid (Fig. 5-*left*). These time-varying stiffness profiles along with applied forces $\mathbf{f}^e$ are then used as the dataset for training two different 4-states GMMs as described next.

GMM/GMR over Cholesky decomposition: Let $\boldsymbol{\xi}^{\mathcal{I}} = \bar{\mathbf{f}}^e$ and $\boldsymbol{\xi}^{\mathcal{O}} = \mathbf{v}^{\mathcal{P}}$, so that a training datapoint is given by $\boldsymbol{\xi} = [\bar{\mathbf{f}}^e, \mathbf{v}^{\mathcal{P}}]^{\mathsf{T}}$. During reproduction, the desired Cholesky vector is computed from the standard GMR as $\hat{\mathbf{v}}^{\mathcal{P}} \sim \mathcal{P}(\mathbf{v}^{\mathcal{P}}|\bar{\mathbf{f}}^e)$. Given $\hat{\mathbf{v}}^{\mathcal{P}}$ we can obtain $\hat{\boldsymbol{\mathcal{K}}}^{\mathcal{P}}$ to later reconstruct the desired stiffness matrices as $\hat{\tilde{\mathbf{K}}}^{\mathcal{P}} = \hat{\boldsymbol{\mathcal{K}}}^{\mathcal{P}^{\mathsf{T}}} \hat{\boldsymbol{\mathcal{K}}}^{\mathcal{P}}$

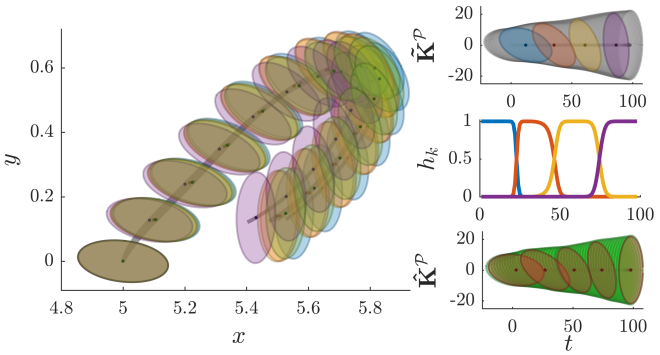

Fig. 5: Force-based stiffness learning/retrieval for MSD using GMM/GMR on $\mathcal{S}^m_+$. *Left*: The Cartesian trajectory of the MSD, starting from rest at point (5,0), and the stiffness ellipsoids at different t for all demonstrations. *Top-right*: demonstrated stiffness ellipsoids over time (in gray) and centers $\mathbf{M}_l$ of the 4-states GMM. *Middle-right*: influence of GMM components on the force-driven GMR estimates, where colors match the distributions shown in the *top* plot. *Bottom-left*: retrieved stiffness ellipsoids (in green) and five ellipsoids (in red) from the ground truth profile to show the match. Stiffness ellipsoids are centered at 0 for visualization purposes.

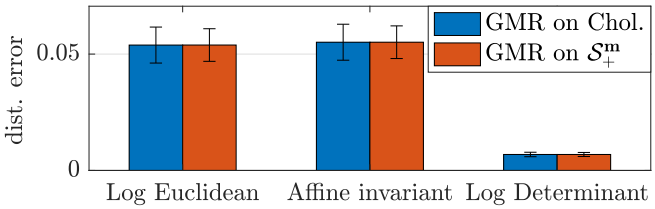

Fig. 6: Different distance-based errors for the stiffness profiles obtained by GMR using the Cholesky vector and $\mathcal{S}^m_+$.

TABLE II: Computation time in seconds for both approaches.

	Cholesky factor	SPD manifold
GMM	0.0629	54.0591
GMR	3.2E−4	6.9E−2

GMM/GMR over SPD manifold: Let $\boldsymbol{\Xi}_{\mathcal{I}\mathcal{I}} = \mathrm{diag}(\bar{\mathbf{f}}^e)$ and $\boldsymbol{\Xi}_{\mathcal{O}\mathcal{O}} = \tilde{\mathbf{K}}^{\mathcal{P}}$, so that a training datapoint is defined by

$$\boldsymbol{\Xi} = \begin{pmatrix} \mathrm{diag}(\bar{\mathbf{f}}^e) & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{K}}^{\mathcal{P}} \end{pmatrix}.$$

Here, the desired stiffness is computed directly from GMR as $\hat{\mathbf{K}}^{\mathcal{P}} \sim \mathcal{P}(\tilde{\mathbf{K}}^{\mathcal{P}}|\bar{\mathbf{f}}^e)$.

Figure 5-*top-right* displays the demonstrated stiffness ellipsoids (in gray) along with the centers $\mathbf{M}_l$ of the four components of the GMM in the SPD manifold. Note that the first and fourth GMM components (in blue and violet, respectively) satisfactorily encapsulate the pattern observed in the MSD stiffness at the beginning and end of the demonstrations, in both shape and orientation. The remaining components encode the temporal evolution of the stiffness. The *middle-right* graph displays the influence of the GMM components over time. The *bottom-right* plot of Fig. 5 shows the desired stiffness profile (in green) retrieved by GMR, and five red ellipsoids corresponding to the ground truth profile. Notice the accurate match between these two profiles. Both estimated stiffness matrices used in GMM and the reproduced ones by GMR lie within the SPD manifold $\mathcal{S}^m_+$ as shown in Fig.7-*left*.

Figure 6 shows a comparison of the average distances

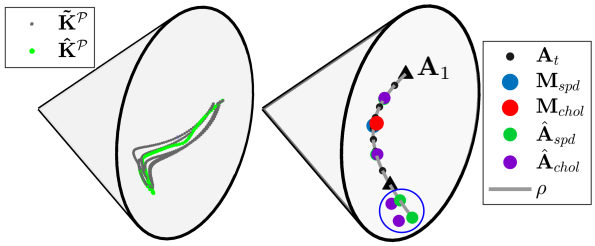

Fig. 7: Representation of SPD data in the SPD manifold $\mathcal{S}_+^m$. *Left*: representation of demonstrated stiffness matrices (gray) used in GMM and stiffness profile retrieved (green) from GMR for the experiment reported in Section VI-B. *Right*: set of covariance data $\{\mathbf{A}_t\}_{t=1}^{20}$, means of single-state GMM in Cholesky and $\mathcal{S}_+^m$ representations, and resulting GMR for both models.

between the ground truth stiffness and the GMR output when using the Cholesky decomposition or Riemannian manifolds. The distances are almost identical in both cases. However, note that the computational time of GMM/GMR over the SPD manifold is significantly higher than that of the model encoding the Cholesky decomposition (see Table II).

C. Geometry-awareness vs. Cholesky factor

As stated in [23], the set of $\mathcal{S}_+^m$ matrices is not a vector space since it is not closed under addition and scalar product, and thus the use of classical Euclidean space methods for treating and analyzing these matrices is inadequate. The use of Euclidean operations to treat $\mathcal{S}_+^m$ matrices is still possible if accuracy is not imperative. In this sense, Cholesky decomposition is a suitable representation to work with SPD matrices in Euclidean space, while significantly reducing the computational burden as shown in [12], [25], and in Section VI-B in this paper. Probabilistic models, which typically assume uncertainties in the data, can properly handle the inaccuracies introduced by the Cholesky decomposition as previously shown in the MSD example.

The inaccuracy introduced by the Cholesky representation arises from the fact that this decomposition does not permit to compute actual distances (geodesic) between two SPD matrices on $\mathcal{S}_+^m$. This can be observed in Fig. 7-*right*, which displays two single-state GMM trained using Cholesky and SPD manifold over a set of covariance data $\{\mathbf{A}_t\}_{t=1}^{20}$. An initial arbitrary $\mathbf{A}_1 \in \mathcal{S}_+^2$ was set and then interpolated through a geodesic $\boldsymbol{\rho}$ that connects it with $\mathbf{A}_{20} = \mathbf{R}^\mathsf{T}\mathbf{A}_1\mathbf{R}$. Note that the mean $\mathbf{M}_{spd}$ lies on the geodesics while $\mathbf{M}_{chol}$ does not. Regressed covariances are computed directly from a time driven GMR using $t = [5.5, 15.5, 22, 24]$ as inputs. The first two values are used to test interpolation cases while the last two evaluated extrapolation capabilities. Both approaches successfully regressed the first two outputs, however, GMR on Cholesky failed to accurately extrapolate the new covariances (as indicated inside the blue circle). GMR on SPD successfully extrapolated new covariances that follow the geodesic path in the manifold.

D. Real Experiment

Robots in industrial floors or unstructured environments may need to open valves, doors, etc., which demands to apply different control forces according to the task conditions. We hypothesize that sensed forces convey relevant information regarding the control commands needed for successful robot performance. These can be governed through stiffness variations, which is suitable for task involving physical interaction with the environment. In this context, the proposed approach was experimentally evaluated using a 7-DoF KUKA robot to perform a valve-turning task (see Fig. 1). The robot was equipped with a gripper and a six-axis force-torque sensor (ATI Mini45) attached between its wrist and the gripper. Note that the presented approach is not task oriented, but a general approach to learn variable impedance skills based on interaction forces. The valve-turning task introduced in this section is used as an example of the practical usability of the proposed approach.

A human teacher provided several demonstrations of the task by kinesthetic teaching. The user held the robot end-effector and physically guided it along the desired trajectory to turn the valve. The kinesthetic demonstrations were given so that the teacher guidance did not interfere with the force sensor readings. We recorded the robot Cartesian position $\mathbf{x}$, and corresponding time derivatives $\dot{\mathbf{x}}$ and $\ddot{\mathbf{x}}$, as well as the sensed forces at the end-effector $\mathbf{f}^e$. In order to vary the valve friction across demonstrations, a screw controlling its friction was tightened up, and subsequently four demonstrations were recorded for each different friction. From these data we estimated the stiffness profile using (3) and (4) with window size $L = 3$ for each case (i.e. valve friction value). In total we modified the valve friction five times (four of which were used as training dataset, while the remaining corresponded to a test case). The demonstration dataset was composed of the estimated stiffness profiles and the corresponding sensed forces, which were used to train a 3-states GMM ($G = 3$).

This learning model was later used by GMR (as described in Section IV) to compute the desired time-varying stiffness profile for unseen sensed forces during the reproduction of the valve-turning. Given each $\hat{\mathbf{K}}^{\mathcal{P}}$ at every t, a force command $\mathbf{F}$ at the robot end-effector was computed by using an impedance controller, which is then transformed to desired joint torques $\boldsymbol{\tau}$ using the Jacobian transpose $\mathbf{J}^\mathsf{T}$ as follows

$$\boldsymbol{\tau} = \mathbf{J}^\mathsf{T}\mathbf{F}, \tag{24}$$

Figure 8 shows the estimated stiffness obtained by GMR, where the five red ellipsoids correspond to one of the demonstrations used to train the GMM. Note how the model retrieves a time-varying stiffness profile similar to those observed during demonstration. The difference lies on the fact that the sensed forces during reproduction corresponded to a valve whose friction coefficient differ from those used during the demonstration phase. Notice also that the robot starts and finishes the task with low stiffness values as the task dynamics required, while displaying higher stiffness while turning the valve.

VII. CONCLUSIONS

We proposed a framework for learning force-based variable stiffness, which is exploited in robotic manipulation tasks. Local stiffness estimates were computed directly from

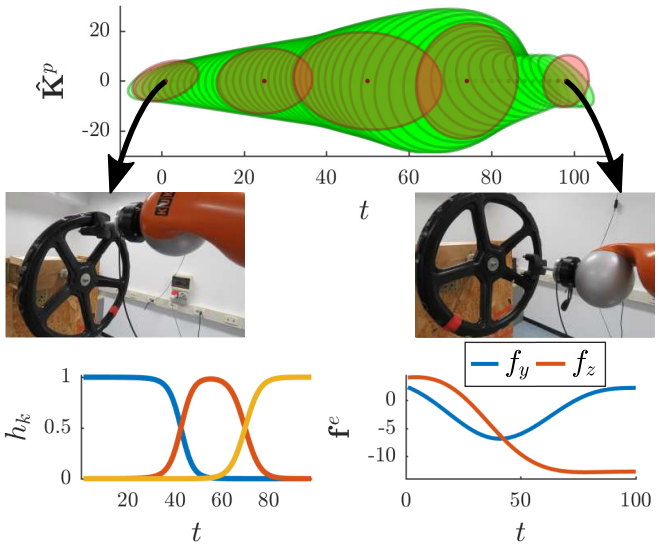

Fig. 8: Force-based variable stiffness retrieval for the valve-turning experiment using GMM/GMR for Cholesky decomposition. *Top*: desired stiffness profiles retrieved by GMR one of the demonstrations, shown as green and red ellipsoids, respectively. *Bottom-left*: influence of the three GMM components on the force-driven GMR estimates. *Bottom-right*: sensed forces profile during the execution of the task for a new valve, which was used as input for GMR. Stiffness ellipsoids are centered at 0 for visualization purposes.

the task dynamics using a computationally efficient sliding-window approach. We evaluated two different approaches to encode stiffness matrices. The first one used a GMM/GMR over Euclidean space, where the Cholesky decomposition of the stiffness matrix was employed as in [12]. The second alternative exploited, for the first time, the SPD manifold-based approach [20] to encode stiffness matrices in its original space. Both approaches were evaluated using a simulation of a 2D MSD system and a real experiment where a robotic arm performed a valve-turning task that was learned from human demonstrations. The reported results showed no significant differences regarding the stiffness profile retrieved by both approaches, which means that both representations (and corresponding learning algorithms) are suitable when learning and retrieving stiffness matrices. Nevertheless, the geometry-aware approach shows better extrapolation capabilities when compared to the Cholesky-based approach.

In future work, we plan to investigate the impact of both Cholesky decomposition and Riemannian manifolds on exploration-based learning methods. In this context, Riemannian-based models may exploit the geometry of the SPD space to reduce the number of explorations and to find better local optima. This may prove to be crucial when a robot needs to rapidly adapt its stiffness in order to perform successfully in a large diversity of task situations. We will also extend the proposed learning framework to handle orientational stiffness learning.

REFERENCES

[1] A. Billard, S. Calinon, and R. Dillmann, *Learning from Humans*. Springer Berlin Heidelberg, 2016, pp. 1995–2014.

[2] 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," *AUTON ROBOT*, vol. 39, no. 2, pp. 199–217, 2015.

[3] L. Rozo, S. Calinon, D. G. Caldwell, P. Jimenez, and C. Torras, "Learning physical collaborative robot behaviors from human demonstrations," *IEEE T ROBOT*, vol. 32, no. 3, pp. 513–527, 2016.

[4] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *IEEE T SYST MAN CY B*, vol. 37, no. 2, pp. 286–298, 2007.

[5] N. Hogan, "Impedance control-an approach to manipulation. i-theory. ii-implementation. iii-applications," *J DYN SYS, MEAS, CONTROL B*, vol. 107, pp. 1–24, 1985.

[6] E. Burdet, R. Osu, D. W. Franklin, T. E. Milner, and M. Kawato, "The central nervous system stabilizes unstable dynamics by learning optimal impedance," *Nature*, vol. 414, no. 6862, p. 446, 2001.

[7] A. Ajoudani, *Transferring Human Impedance Regulation Skills to Robots*. Springer, 2016.

[8] L. Peternel, T. Petrič, E. Oztop, and J. Babič, "Teaching robots to cooperate with humans in dynamic manipulation tasks based on multi-modal human-in-the-loop approach," *AUTON ROBOT*, vol. 36, no. 1-2, pp. 123–136, 2014.

[9] L. Peternel, T. Petrič, and J. Babič, "Robotic assembly solution by human-in-the-loop teaching method based on real-time stiffness modulation," *AUTON ROBOT*, vol. 42, no. 1, pp. 1–17, 2018.

[10] 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 ICRA*, vol. 1, Washington, DC, USA, 2002, pp. 644–650.

[11] L. Rozo, S. Calinon, D. Caldwell, P. Jiménez, and C. Torras, "Learning collaborative impedance-based robot behaviors," in *AAAI*, Washington, USA, 2013, pp. 1422–1428.

[12] K. Kronander and A. Billard, "Learning compliant manipulation through kinesthetic and tactile human-robot interaction," *IEEE T HAPTICS*, vol. 7, no. 3, pp. 367–380, 2014.

[13] P. Kormushev, S. Calinon, and D. G. Caldwell, "Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input," *ADV ROBOTICS*, vol. 25, no. 5, pp. 581–603, 2011.

[14] M. Li, H. Yin, K. Tahara, and A. Billard, "Learning object-level impedance control for robust grasping and dexterous manipulation," in *IEEE ICRA*, Hong Kong, China, 2014, pp. 6784–6791.

[15] J. Rey, K. Kronander, F. Farshidian, J. Buchli, and A. Billard, "Learning motions from demonstrations and rewards with time-invariant dynamical systems based policies," *AUTON ROBOT*, vol. 42, no. 1, pp. 45–64, 2018.

[16] J. Buchli, F. Stulp, E. Theodorou, and S. Schaal, "Learning variable impedance control," *INT J ROBOT RES*, vol. 30, no. 7, pp. 820–833, 2011.

[17] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE J ROBOT AUTOM*, vol. 3, no. 1, pp. 43–53, 1987.

[18] S. Calinon, I. Sardellitti, and D. G. Caldwell, "Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies," in *IEEE/RSJ IROS*, Taipei, Taiwan, 2010, pp. 249–254.

[19] N. J. Higham, "Computing a nearest symmetric positive semidefinite matrix," *LINEAR ALGEBRA APPL*, vol. 103, pp. 103–118, 1988.

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

[21] L. Rozo, N. Jaquier, S. Calinon, and D. G. Caldwell, "Learning manipulability ellipsoids for task compatibility in robot manipulation," in *IEEE/RSJ IROS*, Vancouver, Canada, 2017, pp. 3183–3189.

[22] S. Calinon, F. D'halluin, E. L. Sauser, D. G. Caldwell, and A. G. Billard, "Learning and reproduction of gestures by imitation," *IEEE ROBOT AUTOM MAG*, vol. 17, no. 2, pp. 44–54, 2010.

[23] X. Pennec, P. Fillard, and N. Ayache, "A riemannian framework for tensor computing," *INT J COMPUT VISION*, vol. 66, no. 1, pp. 41–66, 2006.

[24] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi, "Kernel methods on riemannian manifolds with gaussian rbf kernels," *IEEE T PATTERN ANAL*, vol. 37, no. 12, pp. 2464–2477, 2015.

[25] H. Zhu, Y. Chen, J. G. Ibrahim, Y. Li, C. Hall, and W. Lin, "Intrinsic regression models for positive-definite matrices with applications to diffusion tensor imaging," *J AM STAT ASSOC*, vol. 104, no. 487, pp. 1203–1212, 2009.