

Force-based Variable Impedance Learning for Robotic Manipulation

Fares J. Abu-Dakka^{a,*}, Leonel Rozo^a, Darwin G. Caldwell^a

^a*Department of Advanced Robotics, Istituto Italiano di Tecnologia, 16163 Genoa, Italy.
The final version of this article is available at Elsevier with DOI 10.1016/j.robot.2018.07.008*

Abstract

In order for robots to successfully carry out manipulation tasks, they require to exploit contact forces and variable impedance control. The conditions of such type of robotic tasks may significantly vary in dynamic environments, which demand robots to be endowed with adaptation capabilities. This can be achieved through learning methods that allow the robot not only to model a manipulation task but also to adapt to unseen situations. In this context, this paper proposes a learning-from-demonstration framework that integrates force sensing and variable impedance control to learn force-based variable stiffness skills. The proposed approach estimates full stiffness matrices from human demonstrations, which are then used along with the sensed forces to encode a probabilistic model of the task. This model is used to retrieve a time-varying stiffness profile that allows the robot to satisfactorily react to new task conditions. The proposed framework evaluates two different stiffness representations: Cholesky decomposition and a Riemannian manifold approach. We validate the proposed framework in simulation using 2D and 7D systems and a couple of real scenarios.

Keywords: Learning from demonstration, Variable impedance, Robot learning, Robotic manipulation

1. 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 unstructured environments are dynamic, uncertain, and possibly inhabited by humans, therefore making hard-coding an unfeasible approach. Moreover, as manipulation tasks require contact with the environment (and possible humans), it is imperative to use compliant motions, which further increase the complexity of programming. In this context, human expertise can be alternatively exploited in a robot learning approach, where the robot learns, from human examples, to reproduce force-based manipulation tasks that require variable impedance.

In this vein, Learning from Demonstration (LfD) [1] is a user-friendly and intuitive methodology for non-roboticists to teach a new task to a robot. In this case, task-relevant information is extracted from several demonstrations. Standard LfD approaches have focused on trajectory-following tasks, however, recent developments have extended robot learning capabilities to the force and impedance domains [2, 3].

In this paper, we propose to exploit LfD to learn manipulation tasks that demand to use different stiffness levels according to the state of the environment and the task itself, which are substantially related to the robot force-based perceptions. 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.



Figure 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.

The demonstrations along with a virtual model of a spring-damper system are used to obtain time-varying stiffness estimates (Section 3.1). These are computed as the closest symmetric positive semi-definite stiffness matrix of a least-squares estimation (Section 3.3). Such estimates are subsequently used as the desired stiffness for corresponding force patterns observed during the demonstration phase. Both sensed forces and estimated stiffnesses are then probabilistically encoded using a Gaussian mixture model (GMM), which is exploited during reproduction phase to retrieve variable stiffness profiles by Gaussian mixture regression (GMR) (see Sections 4 and 5 for details)¹. In summary, the main contributions of our LfD framework are:

- Exploitation of force sensing and variable impedance control to learn and reproduce manipulation tasks requiring

^{*}Corresponding author

Email address: fares.abudakka@iit.it (Fares J. Abu-Dakka)

¹We 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.

different stiffness levels.

- Extensive analysis and comparison of different stiffness estimation techniques and probabilistic learning methods for encoding and retrieval of full stiffness matrices.

We validate our framework by first using the model of a 2D spring-damper system subjected to an external force, and a simulation to estimate inertia matrices for a 7 DoFs manipulator. Finally, a lifting and a valve-turning task are used as real scenarios to further evaluate the performance of the proposed approach, as shown in Section 6. Discussions of results and conclusions and future research are given in Sections 7 and 8.

2. Related Work

Traditionally, robot learning has been concerned about 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]. Impedance control can be used to achieve compliant motions, in which the controller resembles a virtual spring-damper system between the environment and robot end-effector [5]. This 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 in order 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 degrees of freedom (DoF) force feedback at the operator’s end-effector [9]. In contrast to Peternel *et al.* works, where authors used human impedance for learning, 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. One of the earliest works was introduced by Ikeura and Inooka [10] to show the advantages of variable damping control schemes for a master-slave system to perform lifting tasks, which was then extended by introducing variable stiffness [11]. Tsumugiwa *et al.* [12] introduced a variable impedance control 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 of the human arm, and differential changes in position and force. Rozo *et al.* [13] 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, so that optimal stiffness matrices are guaranteed [3]. Unlike our approach, Rozo *et al.* used 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.

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 [13, 14, 15, 16]. Such techniques were applied to tasks such as ironing [15], assembling furniture [13], and pouring [14]. Kormushev *et al.* [15] 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. Kronander *et al.* [14] used kinesthetic demonstrations to teach haptic-based stiffness variations to a robot. They estimated full stiffness matrices for given positions using GMR, which used a GMM that encoded robot Cartesian positions and the Cholesky vector of the stiffness matrix. Li *et al.* [16] omitted the damping term from the interaction model and used GMM to encode the pose of the end-effector. Then they found the impedance parameters and reference trajectory using optimization techniques. Unlike previous approaches, our approach learns full stiffness matrices that rely not only on the task dynamics but also on sensed interaction forces.

Rey *et al.* [17] used Path Integrals (PI²) policy search for time-invariant policies to learn variable stiffness and motion simultaneously. Their work was built on [18] where PI² 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 PI². 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.

3. 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. With this in mind, 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 end-effector position \mathbf{x} , velocity $\dot{\mathbf{x}}$, and acceleration $\ddot{\mathbf{x}}$, and the associated

Table 1: 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 $\tilde{\mathbf{K}}^{\mathcal{P}}$ at each datapoint using (3) and (4) (see Section 3.2),
- Compose dataset by concatenating $\{\tilde{\mathbf{f}}_{t,n}^e, \tilde{\mathbf{K}}_{t,n}^{\mathcal{P}}\}_{n=1}^N$ for learning (see Section 3.3).
2. Model fitting (see Sections 4 and 5)
- Learn model parameters Θ using (5) or (12).
4. Reproduction (see Sections 4 and 5)
- Set the input \mathcal{I} and output \mathcal{O} elements.
for $t \leftarrow 1$ to T (for each reproduction time step)
- Giving an input \mathcal{I} , estimate the output \mathcal{O} using (7) or (19).
- Compute force command \mathbf{F} at the robot end-effector using the resultant $\hat{\mathbf{K}}^{\mathcal{P}}$ under the <i>unitary-mass</i> assumption
- Compute desired joint torques using (26)
end

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 observed dynamics to a virtual spring-damper system as explained next. After, a GMM is trained to encode both force and stiffness data. The model is later used by GMR to retrieve continuous and smooth stiffness profiles given unseen force trajectories during the reproduction of the task. Table 1 summarizes our approach.

3.1. Interaction Model

In tasks that require physical interaction between a robot and its surroundings, it is highly recommended to use impedance control, which specifies a dynamic relationship between position and force. Using impedance control helps overcome position uncertainties and subsequently avoid large impact forces, since robots are controlled to modulate their motion or compliance according to force perceptions. In this work, we model the robot end-effector as an *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 [19]

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

The dynamics governing the robot behavior are modeled as a virtual spring-damper system as follows

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

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

3.2. Stiffness Estimation from Demonstrations

The full stiffness matrix $\mathbf{K}^{\mathcal{P}}$ can be estimated from demonstrations using different approaches. For example, GMM is

used in [3] to cluster the demonstrated data, and then WLS was applied to estimate a $\mathbf{K}_i^{\mathcal{P}}$ associated with each Gaussian component i using convex optimization. Each $\mathbf{K}_i^{\mathcal{P}}$ was constrained to fulfill a local dynamics extracted from human demonstrations. This approach required several iterations for convergence, which increases the computational cost of the estimation process. Calinon *et al.* [20] estimated $\mathbf{K}^{\mathcal{P}}$ 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, 20], 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 , see Fig. 3. Note that the use of clustering techniques such as GMM may demand to use a high number of Gaussian components 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 along the demonstration data $\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}$, and \mathbf{f}^e , and the full stiffness matrix estimate is computed over the data covered by the window for each time step t . Formally, from (2), let us define $\{\tilde{\mathbf{x}} = \hat{\mathbf{x}} - \mathbf{x}_t\}_t$ 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. Afterwards, 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. 3). The stiffness matrix estimation is carried out at each t by solving the following linear system using least-squares,

$$\mathbf{X}\mathbf{K}^{\mathcal{P}} = \mathbf{Y}. \quad (3)$$

However, a stiffness matrix $\mathbf{K}^{\mathcal{P}} \in \mathcal{S}_+^m$ is symmetric and positive definite (SPD), which means that the estimation computed by (3) is just a rough approximation that does not fulfill SPD constraints. Inspired by [13], we resort to the approximation formulated in [21] to compute the nearest-SPD matrix from the approximation computed in (3), as follows,

$$\tilde{\mathbf{K}}^{\mathcal{P}} = \frac{\mathbf{B} + \mathbf{H}}{2}, \quad (4)$$

$$\mathbf{B} = \frac{\mathbf{K}^{\mathcal{P}} + (\mathbf{K}^{\mathcal{P}})^{\top}}{2}, \quad \mathbf{H} = \mathbf{V}\mathbf{S}\mathbf{V}^{\top},$$

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

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 $\tilde{\mathbf{K}}^{\mathcal{P}} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{\top}$. 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. $\mathbf{K}_t^{\mathcal{V}} = \mathbf{Q}(\zeta\mathbf{\Lambda}^{\frac{1}{2}})\mathbf{Q}^{\top}$, where \mathbf{Q} and $\mathbf{\Lambda}$ are respectively the eigenvectors and eigenvalues of $\tilde{\mathbf{K}}^{\mathcal{P}}$, and $\zeta \in \mathbb{R}^+$ is a tuning parameter.

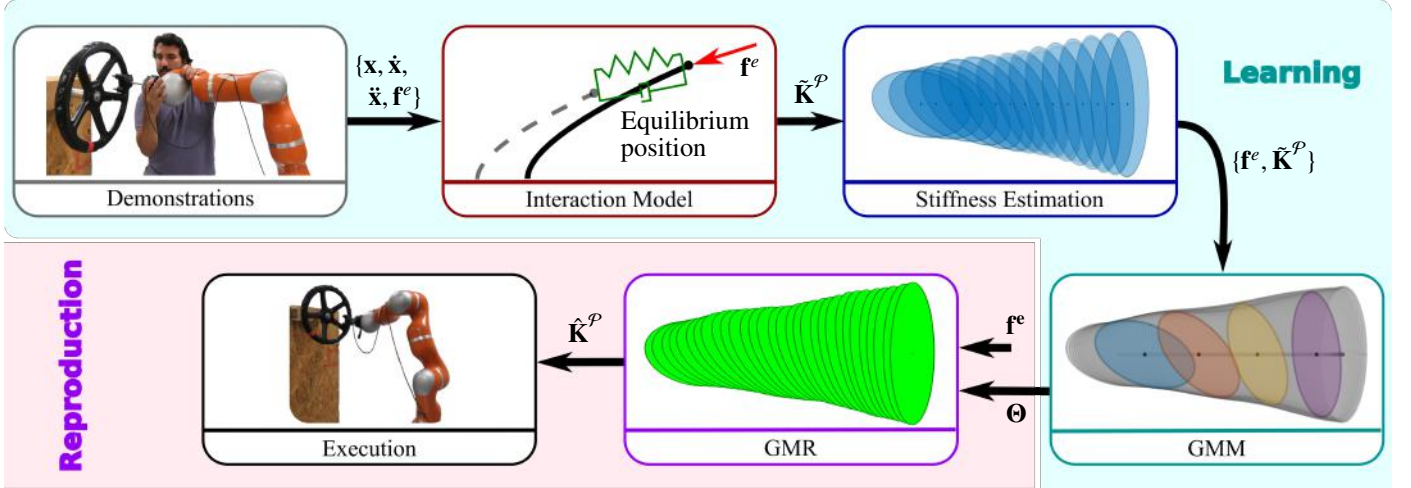


Figure 2: Diagram of the proposed framework for force-based variable impedance learning. *Learning*: The interaction model is used to estimate a stiffness profile from the collected demonstrations. The sensed forces along with the estimated stiffness matrices are used to train a GMM. *Reproduction*: A GMM is generated and later used for retrieving a new stiffness profile by applying GMR, which then will be used through the interaction model to execute the task in new situations.

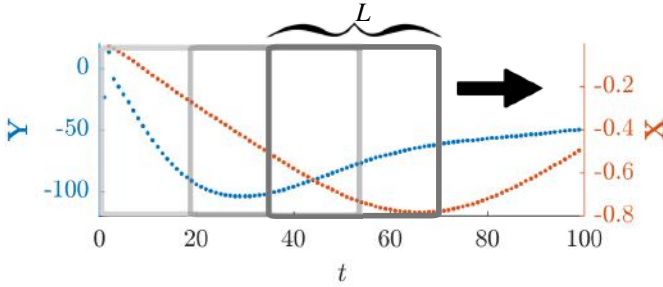


Figure 3: Sliding window approach. A rectangular window with a fixed width L slides across the data over time. For each of these windows, \mathbf{K}^P is estimated using (3). The arrow indicates the sliding direction.

3.3. Stiffness Learning

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

- 1– Learn $\tilde{\mathbf{K}}^P$ using GMM/GMR for Euclidean data. Here, we represent $\tilde{\mathbf{K}}^P$ by its Cholesky vector \mathbf{v}^P obtained from the vectorization of the upper triangle factor \mathbf{K}^P in $\tilde{\mathbf{K}}^P = \mathbf{K}^{P\top} \mathbf{K}^P$. Using this representation we reduce the data space dimensionality to $m + m(m-1)/2$ instead of vectorizing the whole matrix $\tilde{\mathbf{K}}^P$, which replicates stiffness information due to its symmetry.
- 2– Use a tensor-based formulation of GMM/GMR on the Riemannian manifold \mathcal{S}_+^m to directly learn and reproduce $\tilde{\mathbf{K}}^P$ [22], 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 $\tilde{\mathbf{K}}^P$ 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 $n = 1, \dots, N$, 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 $\{\tilde{\mathbf{f}}_t^e, \tilde{\mathbf{K}}_{t,n}^P\}_{n=1}^N$, where $\tilde{\mathbf{f}}_t^e = 1/k (\sum_{s=1}^k \mathbf{f}_{t,s}^e)$ is the mean sensed force across the subset of k demonstrations at each t . Figure. 2 shows an illustrative diagram of the proposed framework.

4. GMM/GMR on Cholesky Factor

Recall from Section 3.2, the stiffness matrix $\mathbf{K}^P \in \mathcal{S}_+^m$, and it can therefore be decomposed using Cholesky decomposition. For generic representation, let us denote \mathbf{A} as any arbitrary SPD matrix and \mathbf{L} its Cholesky factor, so that $\mathbf{A} = \mathbf{L}^\top \mathbf{L}$. 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 ξ_n^I and output vector ξ_n^O , which form a demonstration datapoint $\xi_n = \{\xi_n^I, \xi_n^O\}$.

The training dataset $\{\{\xi_n\} = \{\xi_n^I, \xi_n^O\}\}_{n=1}^N$ is assumed to be normally distributed and modeled by a mixture of G Gaussians

$$\mathcal{P}(\xi_n^I, \xi_n^O) = \sum_{l=1}^G \pi_l \mathcal{N}(\xi_n; \mu_l, \Sigma_l), \quad (5)$$

where $\Theta = \{\pi_l, \mu_l, \Sigma_l\}_{l=1}^G$ are the prior, mean, and covariance parameters. The superindexes I and O respectively indicate input and output dimensions. Maximum-likelihood estimation of the mixture parameters is iteratively performed using the standard Expectation-Maximization (EM) algorithm [23].

In literature, GMM is usually combined with GMR, providing an elegant solution for encoding and synthesizing motor skills in robotics [24]. GMR is used here to retrieve the vectorized versions of $\tilde{\mathbf{L}}$, which are subsequently used to reconstruct

a desired SPD matrix $\hat{\mathbf{A}}$. Formally, let us decompose the GMM parameters μ_l and Σ_l as

$$\mu_l = \begin{bmatrix} \mu_l^I \\ \mu_l^O \end{bmatrix}, \quad \Sigma_l = \begin{bmatrix} \Sigma_l^I & \Sigma_l^{IO} \\ \Sigma_l^{OI} & \Sigma_l^O \end{bmatrix}. \quad (6)$$

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

$$\mathcal{P}(\xi_n^O | \xi_n^I) = \mathcal{N}(\xi_n^O; \hat{\mu}_n^O, \hat{\Sigma}_n^O), \quad (7)$$

where

$$\begin{aligned} \hat{\mu}_n^O &= \sum_{l=1}^G h_l(\xi_n^I) \hat{\mu}_l^O(\xi_n^I), \\ \hat{\mu}_l^O(\xi_n^I) &= \mu_{l,l}^O + \Sigma_{l,l}^{OI} \Sigma_{l,l}^{I-1} (\xi_n^I - \mu_{l,l}^I), \\ \hat{\Sigma}_n^O &= \sum_{l=1}^G h_l(\xi_n^I) (\hat{\Sigma}_l^O + \hat{\mu}_l^O(\xi_n^I) \hat{\mu}_l^O(\xi_n^I)^\top) - \hat{\mu}_n^O \hat{\mu}_n^{O\top}, \\ \hat{\Sigma}_l^O &= \Sigma_{l,l}^O - \Sigma_{l,l}^{OI} \Sigma_{l,l}^{I-1} \Sigma_{l,l}^{IO}. \end{aligned} \quad (8)$$

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

5. GMM/GMR on SPD Manifolds

Recall that $\mathbf{K}^P \in \mathcal{S}_+^m$, therefore, 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 [22, 25] 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 [26]. 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.

5.1. 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. Note that the space of $d \times d$ SPD matrices can be represented as the interior of a convex cone embedded in its tangent space \mathcal{S}_+^d . 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 $\text{Exp}_{\Gamma}(\Delta): \mathcal{T}_{\Gamma}\mathcal{M} \mapsto \mathcal{M}$ is a function that maps a point Δ in the tangent space to a point $\mathbf{Q} \in \mathcal{M}$, so that it lies on the geodesic starting from $\Gamma \in \mathcal{S}_+^m$ in the direction of Δ . On the other hand, the logarithm map $\text{Log}_{\Gamma}(\mathbf{Q}): \mathcal{M} \mapsto \mathcal{T}_{\Gamma}\mathcal{M}$ is

the inverse of the exponential map. The aforementioned mappings are computed as [22]

$$\text{Exp}_{\Gamma}(\Delta) = \Gamma^{\frac{1}{2}} \exp(\Gamma^{-\frac{1}{2}} \Delta \Gamma^{-\frac{1}{2}}) \Gamma^{\frac{1}{2}}, \quad (10)$$

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

Moreover, in order to move elements between tangent spaces while maintaining the angle between them constant, we need to use parallel transport $\mathbb{B}_{\Gamma \mapsto \mathbf{Q}}: \mathcal{T}_{\Gamma}\mathcal{M} \mapsto \mathcal{T}_{\mathbf{Q}}\mathcal{M}$. This is computed for a matrix \mathbf{V} as $\mathbb{B}_{\Gamma \mapsto \mathbf{Q}}(\mathbf{V}) = \mathbf{C}_{\Gamma \mapsto \mathbf{Q}} \Gamma \mathbf{C}_{\Gamma \mapsto \mathbf{Q}}^\top$, where $\mathbf{C}_{\Gamma \mapsto \mathbf{Q}} = \Gamma^{\frac{1}{2}} \exp(\frac{1}{2} \Gamma^{-\frac{1}{2}} \mathbf{V} \Gamma^{-\frac{1}{2}}) \Gamma^{-\frac{1}{2}}$ (see [22] for details).

5.2. GMM on SPD

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

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

where G is the number of Gaussian components, $\mathbf{M} \in \mathcal{M}$ is the mean (and in turn, the origin in the tangent space $\mathcal{T}_{\mathbf{M}}\mathcal{M}$), and $\mathbf{Y} \in \mathcal{T}_{\mathbf{M}}\mathcal{M}$ is the 4th-order covariance tensor. The GMM parameters $\Theta = \{\pi_l, \mathbf{M}_l, \Xi_l\}_{l=1}^G$ on the manifold are estimated using EM algorithm as follows,

E-step:

$$\mathcal{P}(l|\Xi_i) = \frac{\pi_l \mathcal{N}_{\mathcal{M}}(\Xi_i; \mathbf{M}_l, \mathbf{Y}_l)}{\sum_{j=1}^G \pi_j \mathcal{N}_{\mathcal{M}}(\Xi_i; \mathbf{M}_j, \mathbf{Y}_j)}, \quad (13)$$

$$N_l = \sum_{i=1}^N \mathcal{P}(l|\Xi_i). \quad (14)$$

M-step:

$$\mathbf{M}_l \leftarrow \frac{1}{N_l} \text{Exp}_{\mathbf{M}_l} \left(\sum_{i=1}^N \mathcal{P}(l|\Xi_i) \text{Log}_{\mathbf{M}_l}(\Xi_i) \right), \quad (15)$$

$$\mathbf{Y}_l \leftarrow \frac{1}{N_l} \sum_{i=1}^N \mathcal{P}(l|\Xi_i) \text{Log}_{\mathbf{M}_l}(\Xi_i) \otimes \text{Log}_{\mathbf{M}_l}(\Xi_i), \quad (16)$$

$$\pi_l \leftarrow \frac{N_l}{N}. \quad (17)$$

5.3. GMR on SPD

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

$$\begin{aligned} \Xi &= \begin{bmatrix} \Xi_{II} & 0 \\ 0 & \Xi_{OO} \end{bmatrix}, \quad \mathbf{M} = \begin{bmatrix} \mathbf{M}_{II} & 0 \\ 0 & \mathbf{M}_{OO} \end{bmatrix}, \\ \mathbf{Y} &= \left[\begin{array}{cc|cc} \mathbf{Y}_{II}^{II} & 0 & 0 & 0 \\ 0 & \mathbf{Y}_{II}^{OO} & 0 & 0 \\ \hline 0 & 0 & \mathbf{Y}_{OO}^{II} & 0 \\ 0 & 0 & 0 & \mathbf{Y}_{OO}^{OO} \end{array} \right], \end{aligned} \quad (18)$$

where the 4th-order tensor \mathbf{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.

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

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

$$\mathcal{P}(\Xi_{OO}|\Xi_{II}) \sim \mathcal{N}(\hat{\mathbf{M}}_{OO}, \hat{\mathbf{Y}}_{OO}^{OO}), \quad (19)$$

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

$$\Upsilon_l = \text{Log}_{\hat{\mathbf{M}}_{OO}}(\mathbf{M}_{OO,l}) - \tilde{\mathbf{Y}}_{OO,l}^{II} \tilde{\mathbf{Y}}_{II,l}^{II-1} \text{Log}_{\Xi_{II}}(\mathbf{M}_{II,l}), \quad (20)$$

$$\hat{\mathbf{M}}_{OO} \leftarrow \text{Exp}_{\hat{\mathbf{M}}_{OO}}\left(\sum_l h_l \Upsilon_l\right), \quad (21)$$

where

$$h_l = \frac{\pi_l \mathcal{N}(\Xi_{II}; \mathbf{M}_{II,l}, \mathbf{Y}_{II,l}^{II})}{\sum_{j=1}^G \pi_j \mathcal{N}(\Xi_{II}; \mathbf{M}_{II,j}, \mathbf{Y}_{II,j}^{II})} \quad (22)$$

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

$$\hat{\mathbf{Y}}_{OO}^{OO} = \sum_l h_l (\tilde{\mathbf{Y}}_{OO,l}^{OO} - \tilde{\mathbf{Y}}_{OO,l}^{II} \tilde{\mathbf{Y}}_{II,l}^{II-1} \tilde{\mathbf{Y}}_{II,l}^{OO} + \Upsilon_l \otimes \Upsilon_l), \quad (23)$$

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

6. Experiments

Our framework was evaluated in both simulations and a couple of real scenarios. The approaches have been 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.

6.1. Simulation

Three different experiments were carried out to evaluate the proposed framework and to clearly explain its different phases. The simulated tests consisted of:

- 1– Comparison between the SPD approximation (4) and convex optimization proposed in [3] to estimate $\tilde{\mathbf{K}}^P$.
- 2– Stiffness estimation, learning and reproduction with a 2-DoF spring-damper system (MSD).
- 3– Comparison of the probabilistic encoding and retrieval using Cholesky decomposition and SPD manifold approaches in high dimensional spaces.

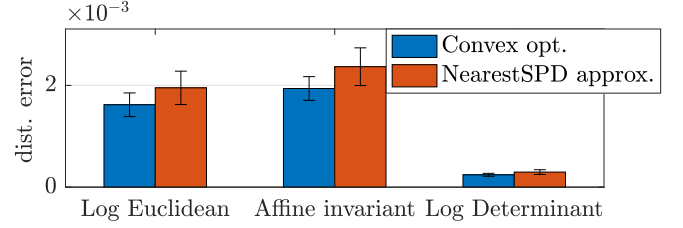


Figure 4: Comparison between the stiffness profile estimated by nearestSPD approximation (4) and the ones estimated by convex optimization (24) using different SPD matrix distances.

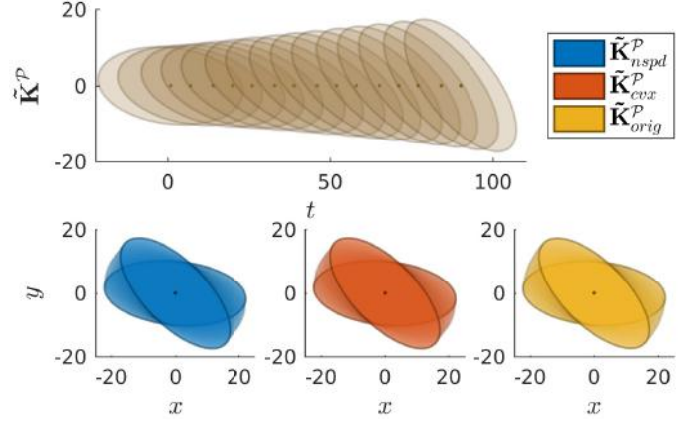


Figure 5: Compare the ground truth stiffness ellipsoids (yellow) of the MSD with the estimated ones using nearestSPD approximation (blue) and convex optimization (orange). Top: stiffness ellipsoids of the ground truth at different time steps matching the nearestSPD and convex optimization estimations at the same time steps. Bottom: 2D plot showing the hand-coded rotation of the stiffness ellipsoids.

For the first two simulated experiments, manually-designed external force profiles \mathbf{f}^e were used to compute the MSD dynamics $[\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}]$ at each time step 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 $\tilde{\mathbf{K}}^P$ was obtained from five demonstrations (differing in the applied external forces) using a window length $L = 3$ and damping coefficient $\mathbf{K}_t^V = \text{diag}(50) [N.s/m]$.²

6.1.1. Stiffness estimation

In this section, a comparison between two methods for estimating SPD matrices from a least-squares method is presented and analyzed. As described in Section 3.2, $\tilde{\mathbf{K}}^P$ can be estimated 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}^P :

$$\text{minimize} \|\mathbf{X}\mathbf{K}^P - \mathbf{Y}\|_2, \quad \text{subject to } \mathbf{K}^P \succeq \mathbf{0}. \quad (24)$$

Figures 4 and 5 show a comparison between the ground truth stiffness matrices and the estimations obtained from the

²Parameters such as window length L , number of Gaussian components G , small eigenvalue ϵ , and number of demonstrations were experimentally chosen.

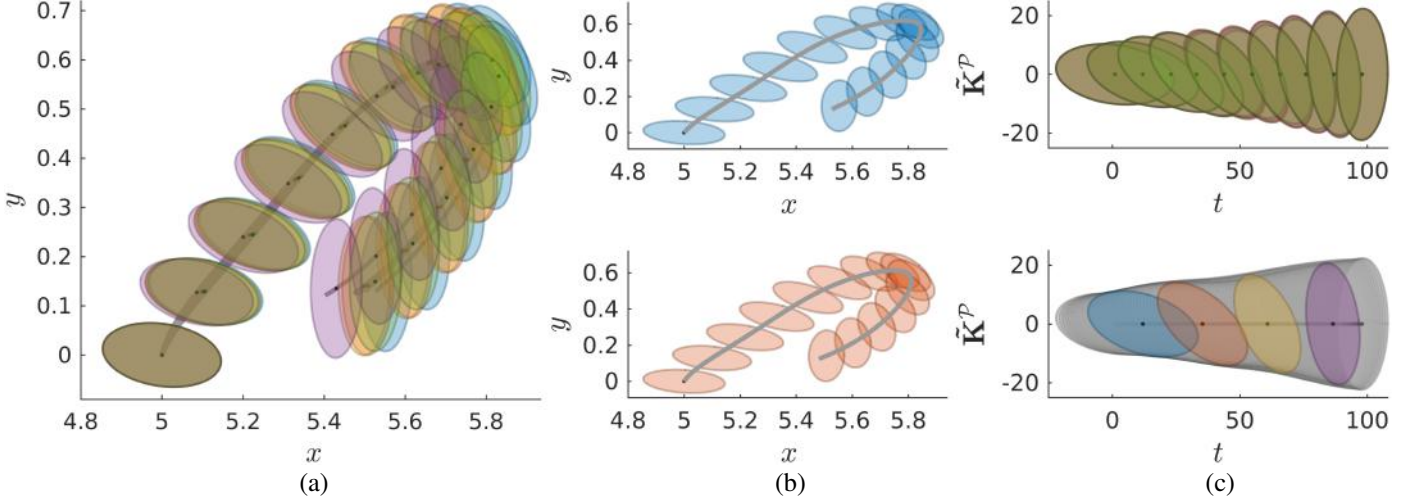


Figure 6: Learning the demonstrated stiffness profile for MSD using GMM on \mathcal{S}_+^m . (a): shows the Cartesian trajectory of the MSD, starting from rest at point (5,0), and the stiffness ellipsoids at different time steps for all demonstrations. (b): shows two demonstrations separately out of five for better illustration. (c): shows the demonstrated stiffness ellipsoids over time (*top* in color, *bottom* in gray) and the centers \mathbf{M}_i of the 4-state GMM (*bottom*).

nearest-SPD approximation (4) and convex optimization (24). In this comparison we used ten MSD systems of different time-varying stiffness. During the evolution of the MSD dynamics in the demonstration phase, the stiffness ellipsoids were initialized from an horizontally-aligned ellipsoid, which was then continuously rotated clockwise through $\mathbf{R}^T \mathbf{K}^P \mathbf{R}$ until reaching a 45 degrees rotation (as shown in Fig. 5). The average computation time for the nearest-SPD approximation and convex optimization were 0.05 and 358 milliseconds per time step, respectively.

Figure 4 shows the average distances between the ground truth matrices of $\tilde{\mathbf{K}}^P$ and the estimations computed from the nearest-SPD approximation and convex optimization methods. It is observed that convex optimization 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, readers are advised to refer to [27]).

6.1.2. Force-based variable impedance for MSD system

In this section, we introduce a toy example using an MSD simulation to evaluate the two approaches described in Sections 4 and 5 for learning force-based variable impedance. The MSD system started from a rest position with an initial spring stiffness \mathbf{K}^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}^P was continuously rotated through $\mathbf{R}^T \mathbf{K}^P \mathbf{R}$ until reaching a vertically-aligned ellipsoid (as shown in Fig. 6a and 6c-*top*). These time-varying stiffness profiles, along with the 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 us denote \mathbf{v}^P as the vectorized version of the Cholesky factor \mathbf{K}^P . Then, we define $\xi^I = \tilde{\mathbf{f}}^e$ and $\xi^O = \mathbf{v}^P$, so that a training datapoint is given by $\xi = [\tilde{\mathbf{f}}^e, \mathbf{v}^P]^T$. During reproduction, the desired Cholesky

vector is computed from the standard GMR as $\hat{\mathbf{v}}^P \sim \mathcal{P}(\mathbf{v}^P | \tilde{\mathbf{f}}^e)$. Given $\hat{\mathbf{v}}^P$ we can obtain $\hat{\mathbf{K}}^P$ to later reconstruct the desired stiffness matrices as $\tilde{\mathbf{K}}^P = \hat{\mathbf{K}}^P \hat{\mathbf{K}}^P$.

GMM/GMR over SPD manifold: We define $\Xi_{II} = \text{diag}(\tilde{\mathbf{f}}^e)$ and $\Xi_{OO} = \tilde{\mathbf{K}}^P$, so that, a training datapoint is defined by

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

Here, the desired stiffness is computed directly from the GMR over an SPD manifold as $\hat{\mathbf{K}}^P \sim \mathcal{P}(\mathbf{K}^P | \tilde{\mathbf{f}}^e)$ (see Fig. 7-*middle*).

Figure 6c-*bottom* displays the demonstrated stiffness ellipsoids (in gray) along with the centers \mathbf{M}_i of the four components of the GMM encoding Ξ_{OO} 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 desired stiffness profiles reproduced by GMR are displayed in Fig. 7-*top* plot as ellipsoids, whose colors match the distribution of the GMM components in Fig. 6c-*bottom*. The middle plot of Fig. 7 shows the match between the ground truth ellipsoids profile in red and the retrieved one from GMR at different time steps. The *bottom* plot of the same figure shows the reproduced stiffness ellipsoids profile in different time steps coinciding with the ground truth, both displayed over the Cartesian trajectory of the MSD. Note that the results of GMM/GMR for both stiffness representations are almost identical, so we here report only the results corresponding to GMM/GMR over the SPD manifold.

Figure 8 illustrates a comparison of the average distances 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, it is worth highlighting that the computational time of GMM/GMR

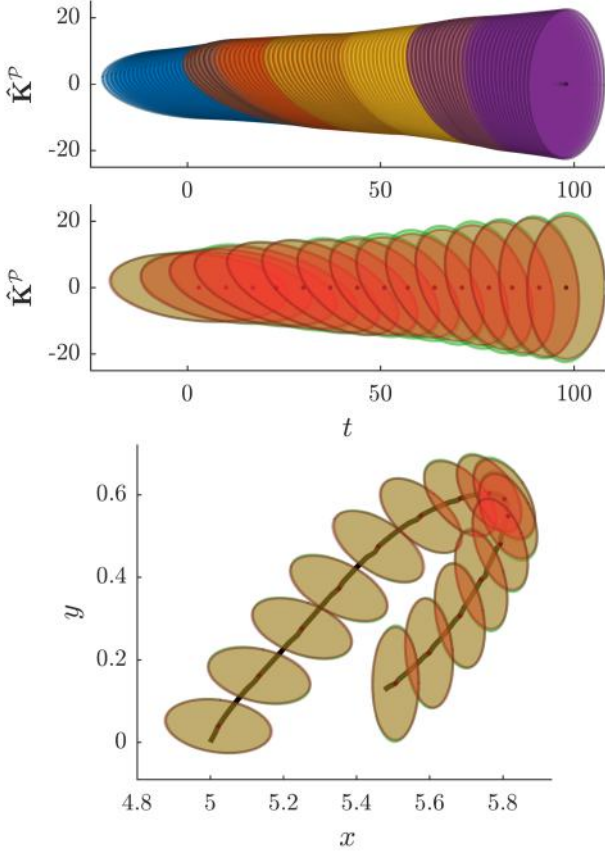


Figure 7: Reproduction of the stiffness profile for MSD using GMR on \mathcal{S}_+^m . *Top*: Desired stiffness ellipsoids estimated by GMR, where colors match the GMM components distribution in Fig. 6c-bottom. Color overlapping corresponds to more than one GMM components influencing the regression. *Middle*: stiffness ellipsoids of the ground truth (in red) at different time steps matching the GMR estimates. *Bottom*: The retrieved stiffness ellipsoids in different time steps coinciding the ground truth over the Cartesian trajectory of the MSD.

over the SPD manifold is significantly higher than that of the model encoding the Cholesky decomposition (see Table 2).

Table 2: Computation time in seconds for GMM/GMR in both Cholesky and SPD in the case of 2D.

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

6.1.3. Learning SPD matrices in high dimensional spaces

Recall from section 6.1.2, the efficiency of GMM/GMR on Cholesky and SPD manifold is almost identical for a 2D system. Here, we are interested in testing if their performance deteriorate in higher dimensional spaces, for example when the model is required to encode the inertia matrix of a redundant manipulator.

In this simulation, we used the Robotics Toolbox for MATLAB [28] to simulate a Barrett WAM 7-DoF manipulator, whose dynamics is described by

$$\tau(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = \mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}), \quad (25)$$

where $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \tau \in \mathbb{R}^7$ are joint positions, velocities, accelerations, and torques of the manipulator. $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ denote the

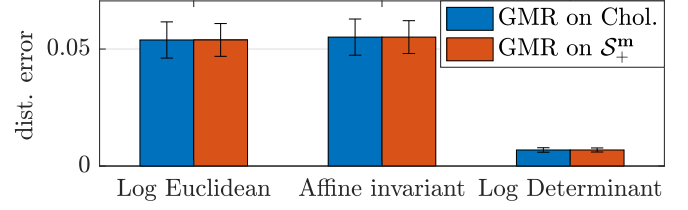


Figure 8: The distance error between the stiffness profile obtained by GMR over Cholesky decomposition and those obtained by GMR on SPD manifold using different SPD matrix distances.

Coriolis and centripetal forces while $\mathbf{G}(\mathbf{q})$ is the gravity term. $\mathbf{B}(\mathbf{q}) \in \mathbb{R}^{7 \times 7}$ is SPD and represents the inertia matrix of the manipulator.

The simulated experiment is designed by generating different trajectories (which act as different demonstrations), using a minimum jerk algorithm, between an initial and final configuration, which belong to different areas in the robot workspace. Each trajectory is composed of 100 datapoints where $\mathbf{B}(\mathbf{q})$ is calculated at each datapoint. The joint trajectories and corresponding inertia matrices are used later as training data for both GMM/GMR methods. In this simulation we used five demonstrations to train two, three, and four states GMM ($G = 2, 3$, and 4).

Regarding the model trained with Cholesky decomposition, the inputs and outputs are defined as $\xi^I = \mathbf{q}$ and $\xi^O = \mathbf{b}$ (where \mathbf{b} is the vectorization of the Cholesky factor of \mathbf{B}). We learn a GMM using datapoints given by $\xi = [\mathbf{q}, \mathbf{b}]^T$ and subsequently obtain a desired vectorized version of the Cholesky factor of the inertia matrix computed from the standard GMR as $\hat{\mathbf{b}} \sim \mathcal{P}(\mathbf{b}|\mathbf{q})$. From $\hat{\mathbf{b}}$ we can now reconstruct the desired inertia matrices $\hat{\mathbf{B}}$ using Cholesky decomposition.

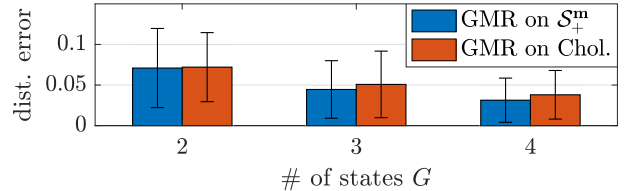


Figure 9: Comparison between the inertia matrix computed using GMM/GMR on Cholesky and on SPD. Log-Euclidean is used to compare the GMR results by training 2, 3, and 4 states GMM using 5 demonstrations.

Table 3: Computation time in seconds for both GMM/GMR approaches

	2-states	3-states	4-states
GMM on Chol.	0.0778	0.0504	0.0377
GMR on Chol.	3.0E−4	3.3E−4	3.4E−4
GMM on SPD	64.5438	88.7110	127.9287
GMR on SPD	0.5702	0.8553	1.1917

On the other hand, the input and output variables for the model trained over the SPD manifold are defined as $\Xi_{II} = \text{diag}(\mathbf{q})$ and $\Xi_{OO} = \mathbf{B}$, and therefore a training datapoint of each dataset for GMM is given by

$$\Xi = \begin{pmatrix} \text{diag}(\mathbf{q}) & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{pmatrix}.$$

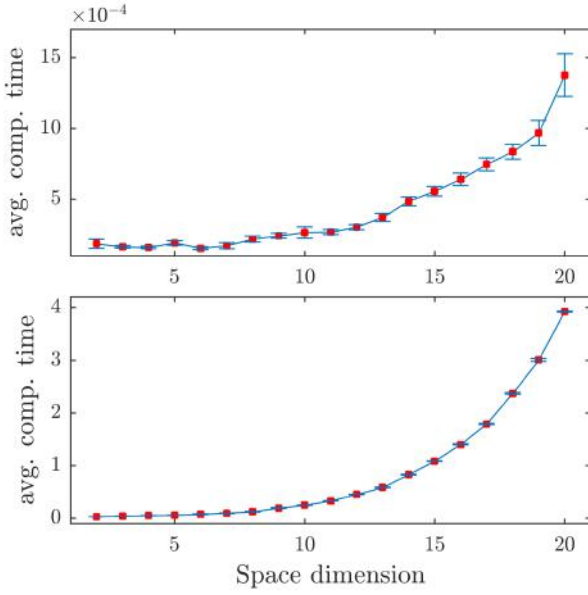


Figure 10: Mean and standard deviation of the computational time [sec] for 50 runs as a function of data dimensionality. *Top*: GMR over Cholesky vector. *Bottom*: GMR over Riemannian manifold.

Here, the desired inertia is computed directly from the GMR over an SPD manifold as $\hat{\mathbf{B}} \sim \mathcal{P}(\mathbf{B}|\text{diag}(\mathbf{q}))$. As observed in Fig. 9, it is clear that GMM/GMR on an SPD manifold provides a slightly better estimation of the inertia matrix than the model trained with the Cholesky decomposition, but with significantly higher computational cost, (see Table 3). In this experiment we used log-Euclidean metric for comparison. The other metrics have also been tested, however, they are omitted as they do not add new information.

Finally, we evaluated the computational time required to implement GMR at each time step for different dataspace dimensionalities (see Fig. 10). The statistical evaluation was computed over 50 runs for each dimensionality value using GMR for both Cholesky decomposition (Fig. 10-*top*) and Riemannian manifold approaches (Fig. 10-*bottom*). Here we used arbitrary SPD matrices of dimensionality \mathcal{S}_+^2 to \mathcal{S}_+^{20} . Notice that, for both approaches, the computational time increases exponentially as a function of the space dimensionality. Moreover, GMR with Cholesky decomposition is significantly less expensive in computational terms when compared to the Riemannian manifold-based approach (e.g. ≈ 0.0014 and ≈ 3.9240 seconds at \mathcal{S}_+^{20} , respectively).

6.2. Real Experiments

The proposed approach was also evaluated experimentally using a KUKA LWR 7-DoF robot, with the Fast-Research Interface, to perform two different tasks: (1) valve-turning task (see Fig. 1), and (2) lifting task (see Fig. 13). In both setups, the robot was equipped with a gripper and a six-axis force-torque sensor (ATI Mini45) attached between its wrist and the gripper. In both experiments, we programmed the KUKA robot in joint impedance control mode. In both experiments we used the Cholesky representation to encode the stiffness matrices. The accompanying videos show both the demonstration and reproduction phases of the aforementioned experiments.

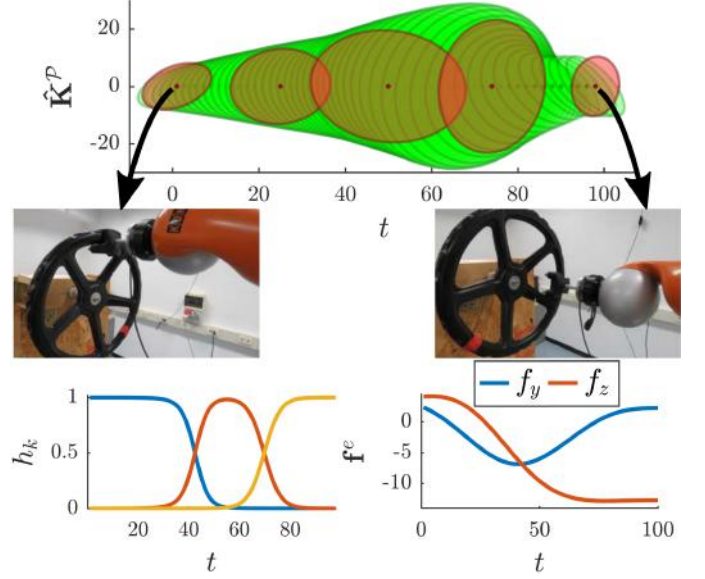


Figure 11: Force-based variable stiffness retrieval of the valve-turning experiment using GMM/GMR for Cholesky decomposition. *Top*: desired stiffness profiles retrieved by GMR and one of the demonstrations, shown as green and red ellipsoids, respectively. *Bottom-left*: influence of the three GMM components on the force-driven GMR estimates. *Bottom-right*: the sensed force profile during the execution of the task for a new valve, which was used as an input for GMR.

6.2.1. Valve-turning experiment

Robots in industrial floors, unstructured hostile environments or other similar scenarios (where it is dangerous for human to access) may require to open valves, doors, etc., that demand to apply different control forces according to different friction forces, mass, etc. In that sense, we hypothesize that sensed forces convey relevant information regarding the control forces needed to opening valves, which can be governed through stiffness variations.

In order to perform our experiment, the human teacher provides several demonstrations by kinesthetic teaching. The user holds the robot end-effector and guides it along the desired trajectory (in this case, to turn the valve) in such a way that the desired task is successfully executed. Note that the kinesthetic demonstrations are given so that the teacher guidance does not interfere with the force sensor readings. We recorded robot position \mathbf{x} , and its 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 friction value. From these demonstrations we estimated the stiffness profile using (3) and (4) and window size $L = 3$ for each case (i.e. valve stiffness 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 sensed forces, which were used to train the learning model.

A 3-states GMM ($G = 3$) was trained using four demonstrations. This model was later used by GMR (as described in Section 4) to compute the desired time-varying stiffness profile for the unseen sensed forces during the reproduction of the valve



Figure 12: Objects used in lifting task experiment. The object emulates unexpected mass variations by interconnecting three blocks of wood using strings. Five different block sets were used for learning and reproduction of the lifting task.

turning. Given each stiffness matrix $\hat{\mathbf{K}}^p$ at every time step, 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 using the Jacobian transpose \mathbf{J}^T as follows

$$\boldsymbol{\tau} = \mathbf{J}^T \mathbf{F}. \quad (26)$$

The estimated stiffness profile obtained by GMR is shown at the top plot in Fig. 11, where the five red ellipsoids represent one of the demonstrations used to train the GMM. The figure shows how the model retrieves a time-varying stiffness profile similar to the stiffness estimated from demonstrations. The difference lies on the fact that the sensed forces during reproduction corresponded to a valve with a different friction force (changed by the screw) that was not used during the demonstration phase. Note 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.

6.2.2. Lifting task

Manipulating objects whose mass varies over time has many applications. For example, lifting deformable objects (e.g.: iron chains in industry or mattress at home), or in the case of waiters loading and unloading their trays with a large diversity of objects. These scenarios share the same idea, which is manipulating dynamic mass objects, which requires to adapt the arm stiffness according to the sensed mass variation.

The aim of this experiment is to illustrate the ability of the proposed approach to adapt the robot stiffness to sudden force variations. For example, sudden force variations induced by the gravitational loading during a lifting task. To emulate that, we designed an object composed of three wooden blocks interconnected by strings, as shown in Fig. 12.

In order to perform the experiment, the human operator demonstrates the task several times using kinesthetic guiding, (see Fig. 13). In this case, we provided five demonstrations for four different sets of objects (their weights are in Table 4) to estimate the stiffness profile, using (3) and (4) and window size $L = 3$, resulting in four demonstrations of different stiffness profiles with corresponding sensed forces, which were used to

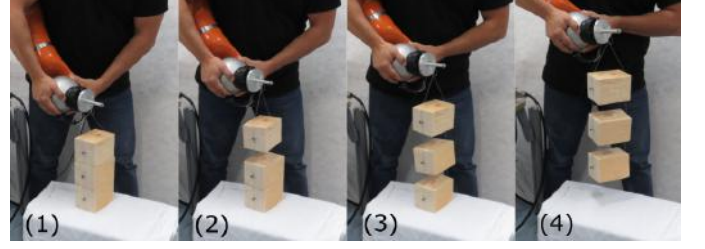


Figure 13: A human operator teaches the robot how to lift a chain of blocks.

Table 4: Weights in [Kg] of the blocks used in the lifting task experiment.

Demonstrations	1 st set	0.561	0.553	0.552
	2 nd set	0.416	0.434	0.428
	3 rd set	0.346	0.288	0.319
	4 th set	0.527	0.593	0.563
Reproduction	unseen set	0.386	0.353	0.350

train a 3-states GMM. This model was used later by GMR (Section 4) to estimate the desired time-varying stiffness profile for unseen sensed forces (produced by a set of objects of different mass), during the reproduction of the lifting task. The desired time-varying stiffness profile is used by an impedance controller to calculate the Cartesian control forces at the robot end-effector, which are then transformed to desired joint torques using the Jacobian transpose.

Figure 14 shows the reproduced Cartesian trajectory of the manipulator end-effector in the lifting task experiment. Note that the resulting Cartesian trajectory (blue circles) followed by the robot end-effector is similar to those observed during the demonstration phase. Figure 15 shows the desired stiffness ellipsoids (in green-top), for the lifting task, reproduced by GMR over time. In the same figure, the five red ellipsoids are taken from one of the demonstrations in order to show a reference stiffness evolution for comparison purpose. Note that the stiffness ellipsoids profile retrieved by our model follows a similar temporal evolution as the stiffness estimated from demonstrations. The difference lies on the fact that the sensed forces during reproduction corresponded to a new set of blocks that was not used during the demonstration phase. Noticeably, the robot lifted the new set of objects following the desired vertical trajectory as its stiffness adapted over time.

7. Discussion

In this paper, we exploited a simple interaction model to obtain local stiffness estimates that heavily rely on force perceptions, which entails a relevant research work aimed at learning complex manipulation skills. The stiffness estimation is approximated to the nearest SPD matrix using (4). However, we also demonstrated the possibility to achieve slightly better stiffness estimates using convex optimization, which significantly compromises the computational cost of the estimation process.

Moreover, we provided an extensive analysis to compare Cholesky-based and geometry-aware models to learn and retrieve full stiffness matrices. In this vein, as stated in [26], the set of \mathcal{S}_+^m matrices is not a vector space since it is not closed

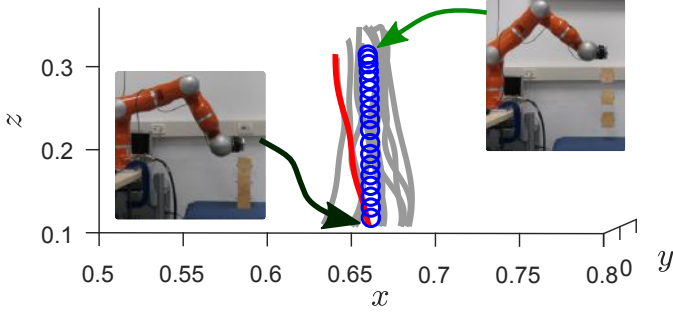


Figure 14: Lifting task: gray curves represent the Cartesian trajectories of the robot end-effector during the demonstrations, the red one depicts the least-square estimation, while the blue circles represent the Cartesian trajectory reproduced by GMR on Cholesky decomposition.

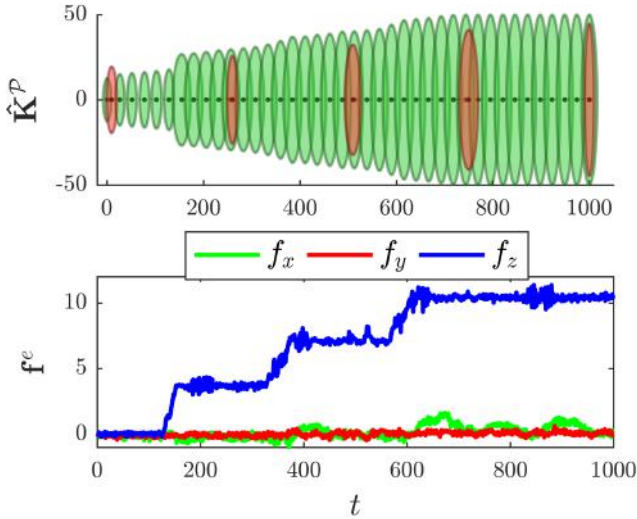


Figure 15: Force-based variable stiffness retrieval of the lifting task using GMM/GMR for Cholesky decomposition. *Top*: desired stiffness profiles retrieved by GMR and one demonstration, shown as green and red ellipsoids, respectively. *Bottom*: the sensed force profile during the execution of the task for a new set of blocks, which was used as an input for GMR.

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 [14, 29], and in Section 6.1.2 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. 16, which displays two single-state GMM trained using Cholesky and Riemannian 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 ρ that connects it with $\mathbf{A}_{20} = \mathbf{R}^T \mathbf{A}_1 \mathbf{R}$. Note that the mean \mathbf{M}_{spd} lies on the geodesics while \mathbf{M}_{chol} does not. Regressed covari-

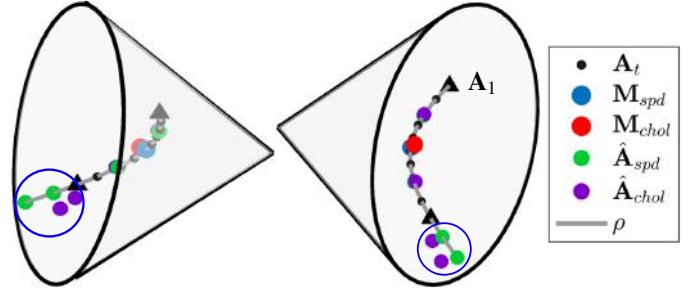


Figure 16: Set of covariance data $\{\mathbf{A}_t\}_{t=1}^{20}$, means of single-state GMM in Cholesky and \mathcal{S}_+^m representations, and resulting GMR for both cases.

ances 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.

Kronander *et al.* [14] also used GMM in Euclidean space to directly encode full stiffness matrices using Cholesky decomposition, however, their learning approach is exclusively governed by the robot position. Furthermore, during the execution of the task, the teacher wiggles the robot to make it more compliant, or increases the pressure with which he holds the robot to make it stiffer. However, this way of varying robot stiffness during the execution is not suitable in tasks that demand physical contact. In contrast, our approach directly uses the sensed forces to automatically estimate suitable full stiffness matrices at each time step during the execution of the task (e.g. MSD simulation and valve-turning experiment). Unlike [14] and [17], 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).

On the other hand, while it is clear that there exist significant differences in terms of computational cost between the learning models presented in Sections 4 and 5, we will explore how robust these approaches may be when used with iterative methods such as those in reinforcement learning. We expect that geometry-aware methods provide more accurate information about the search space, which may therefore help the algorithm convergence with a fewer number of iterations when compared with Euclidean-based techniques.

Finally, our approach successfully adapts robot stiffness not only to continuous variation of forces over time, but also to sudden force changes (e.g. lifting task). However, our learning framework requires the robot to dynamically interact with the environment, which might present some limitations for quasi-stationary scenarios where different stiffness levels may also be required (e.g., a waiter loading or unloading the tray). This may involve the use of more sophisticated interaction models which would, in turn, make the stiffness estimation process more complex. Another alternative is to use simple interaction models to obtain rough stiffness estimates which can be improved by refinement techniques based on RL approaches [18].

8. Conclusions

In this paper, we introduced a new framework for learning force-based variable impedance, which is exploited in robotic manipulation tasks that required adaptive stiffness levels. The skills learned in this paper relied on stiffness estimations computed from human demonstrations, which were then used along with the sensed forces to encode a probabilistic model of the task. The resulting model was then used to retrieve a time-varying stiffness profile by GMR. This framework endowed the robot with the ability to adapt its stiffness autonomously during the execution in function of the perceived forces.

We evaluated two different approaches to encode stiffness matrices, namely, GMM/GMR over Cholesky decomposition of the stiffness matrix, and GMM/GMR over the SPD manifold. Both approaches were extensively evaluated using two simulated models and two real experiments. The reported results showed no significant differences regarding the stiffness profile retrieved by both approaches in 2D, but the performance of the Cholesky-based model slightly deteriorated when learning high dimensional SPD matrices. The geometry-aware approach displayed a higher accuracy with a significant increased computational cost. The analysis pointed out that either approach may be used for force-based variable impedance learning.

In future work, we plan to find connections between the Cholesky decomposition and the operations that take place in the SPD manifold. Also, we will work on exploration-based learning methods in order to test which representation may be more suitable, which will prove to be crucial when a robot needs to significantly adapt its stiffness in order to perform successfully in a large diversity of task situations.

References

- [1] A. Billard, S. Calinon, R. Dillmann, *Learning from Humans*, Springer Berlin Heidelberg, 2016, pp. 1995–2014.
- [2] F. J. Abu-Dakka, B. Nemeš, J. A. Jørgensen, T. R. Savarimuthu, N. Krüger, A. Ude, Adaptation of manipulation skills in physical contact with the environment to reference force profiles, *Autonomous Robots* 39 (2015) 199–217.
- [3] L. Rozo, S. Calinon, D. G. Caldwell, P. Jimenez, C. Torras, Learning physical collaborative robot behaviors from human demonstrations, *IEEE Transactions on Robotics* 32 (2016) 513–527.
- [4] P. Ouyang, W. Zhang, M. M. Gupta, An adaptive switching learning control method for trajectory tracking of robot manipulators, *Mechatronics* 16 (2006) 51–61.
- [5] N. Hogan, Impedance control—an approach to manipulation. i-theory. ii-implementation. iii-applications, *ASME Transactions Journal of Dynamic Systems and Measurement Control* B 107 (1985) 1–24.
- [6] E. Burdet, R. Osu, D. W. Franklin, T. E. Milner, M. Kawato, The central nervous system stabilizes unstable dynamics by learning optimal impedance, *Nature* 414 (2001) 446–449.
- [7] A. Ajoudani, *Transferring Human Impedance Regulation Skills to Robots*, Springer, 2016.
- [8] L. Peternel, T. Petrič, E. Oztop, J. Babič, Teaching robots to cooperate with humans in dynamic manipulation tasks based on multi-modal human-in-the-loop approach, *Autonomous Robots* 36 (2014) 123–136.
- [9] L. Peternel, T. Petrič, J. Babič, Robotic assembly solution by human-in-the-loop teaching method based on real-time stiffness modulation, *Autonomous Robots* 42 (2018) 1–17.
- [10] R. Ikeura, H. Inooka, Variable impedance control of a robot for cooperation with a human, in: *IEEE International Conference on Robotics and Automation*, volume 3, Nagoya, Japan, 1995, pp. 3097–3102.
- [11] M. Rahman, R. Ikeura, K. Mizutani, Investigating the impedance characteristic of human arm for development of robots to co-operate with human operators, in: *IEEE International Conference on Systems, Man, and Cybernetics*, volume 2, Tokyo, Japan, 1999, pp. 676–681.
- [12] T. Tsumugiwa, R. Yokogawa, 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*, volume 1, Washington, DC, USA, USA, 2002, pp. 644–650.
- [13] L. Rozo, S. Calinon, D. Caldwell, P. Jiménez, C. Torras, Learning collaborative impedance-based robot behaviors, in: *27th AAAI Conference on Artificial Intelligence*, 2013, pp. 1422–1428.
- [14] K. Kronander, A. Billard, Learning compliant manipulation through kinesthetic and tactile human-robot interaction, *IEEE Transactions on Haptics* 7 (2014) 367–380.
- [15] P. Kormushev, S. Calinon, D. G. Caldwell, Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input, *Advanced Robotics* 25 (2011) 581–603.
- [16] M. Li, H. Yin, K. Tahara, A. Billard, Learning object-level impedance control for robust grasping and dexterous manipulation, in: *IEEE International Conference on Robotics and Automation*, Hong Kong, China, 2014, pp. 6784–6791.
- [17] J. Rey, K. Kronander, F. Farshidian, J. Buchli, A. Billard, Learning motions from demonstrations and rewards with time-invariant dynamical systems based policies, *Autonomous Robots* 42 (2018) 45–64.
- [18] J. Buchli, F. Stulp, E. Theodorou, S. Schaal, Learning variable impedance control, *The International Journal of Robotics Research* 30 (2011) 820–833.
- [19] O. Khatib, A unified approach for motion and force control of robot manipulators: The operational space formulation, *IEEE Journal on Robotics and Automation* 3 (1987) 43–53.
- [20] S. Calinon, I. Sardellitti, D. G. Caldwell, Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies, in: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, 2010, pp. 249–254.
- [21] N. J. Higham, Computing a nearest symmetric positive semidefinite matrix, *Linear Algebra and Its Applications* 103 (1988) 103–118.
- [22] L. Rozo, N. Jaquier, S. Calinon, 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.
- [23] S. Calinon, F. Guenter, A. Billard, On learning, representing, and generalizing a task in a humanoid robot, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 37 (2007) 286–298.
- [24] S. Calinon, F. D’halluin, E. L. Sauser, D. G. Caldwell, A. G. Billard, Learning and reproduction of gestures by imitation, *IEEE Robotics & Automation Magazine* 17 (2010) 44–54.
- [25] N. Jaquier, 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.
- [26] X. Pennec, P. Fillard, N. Ayache, A riemannian framework for tensor computing, *International Journal of Computer Vision* 66 (2006) 41–66.
- [27] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, M. Harandi, Kernel methods on riemannian manifolds with gaussian rbf kernels, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37 (2015) 2464–2477.
- [28] P. Corke, *Robotics, vision and control fundamental algorithms in matlab® second, completely revised, extended and updated edition*, 2017.
- [29] H. Zhu, Y. Chen, J. G. Ibrahim, Y. Li, C. Hall, W. Lin, Intrinsic regression models for positive-definite matrices with applications to diffusion tensor imaging, *Journal of the American Statistical Association* 104 (2009) 1203–1212.