# Learning Manipulability Ellipsoids for Task Compatibility in Robot Manipulation

Leonel Rozo[1], Noémie Jaquier[2], Sylvain Calinon[2,1] and Darwin G. Caldwell[1]

*Abstract*— **Posture body variation is one of the ways in which humans skillfully and naturally augment their motion and strength capabilities along specific task-space directions in order to successfully perform complex manipulation tasks. Posture variation also has a significant role in robot manipulation, where manipulability arises as a useful criterion to analyze and control the robot dexterity as a function of its joint configuration. In this context, this paper introduces the promising idea of *manipulability transfer*, a method that allows robots to learn and reproduce desired manipulability ellipsoids from expert demonstrations. The proposed framework is built on a tensor-based formulation of Gaussian mixture model that takes into account that manipulability ellipsoids lie on the manifold of symmetric positive definite matrices. This geometry-aware method is used to design a manipulability-based redundancy resolution that allows the robot to modify its posture so that its manipulability ellipsoid coincides with the desired one. Experiments in simulation validate the functionality of the proposed approach, which extends the robot learning capability beyond trajectory, force and impedance learning approaches.**

## I. INTRODUCTION

When we perform a manipulation task, we naturally place our arms (and body) in a posture that is best suited to carry out the task at hand. Such posture variation is a means through which the motion and strength characteristics of the arms are made compatible with the task requirements. For example, human biomechanics strongly influence the motion planning of reaching tasks [1]. Also, humans change their posture to be mechanically most resistant to potential perturbations coming from obstacles occupying the workspace [2]. Therefore, there exists evidence that the human body posture, governed by motor control commands, plays a relevant role on how humans perform manipulation tasks.

The robotics community has also been aware that posture variation can have a significant impact on manipulation. For instance, the robot joint configuration may greatly affect the ability to maneuver in the workspace. Hence, measures that determine the robot capabilities to perform a task are of high interest. One of the well-established tools for motion and dexterity analysis of robot manipulators is the so-called manipulability ellipsoid [3]. This geometric measure indicates the ability to arbitrarily perform motion and exert force along the different task directions in a given joint configuration.

The manipulability ellipsoid has been used to measure the compatibility of robot postures with respect to fine and coarse manipulation [4], to improve minimum-time trajectory planning tasks through a manipulability-aware inverse kinematics algorithm [5], and to support a grasp selection process that favors high manipulability in the robot workspace [6]. Other works have focused on maximizing the manipulability ellipsoid volume in trajectory generation algorithms [7], and task level robot programming frameworks [8], to obtain singularity-free joint trajectories and high task-space dexterity. However, the maximization of only the volume of an ellipsoid to achieve the maximum dexterity in motion may cause a reverse effect on the flexibility in force [9].

In contrast to the foregoing approaches that do not specify a desired robot manipulability for the task, Lee *et al.* proposed an optimization method for selecting a reaching posture for a humanoid robot that guarantees high arm manipulability [10]. The desired manipulability is manually specified from a set of ellipsoid volumes designed in accordance to the reaching task requirements. Similarly, a series of desired manipulability ellipsoids is predefined according to Cartesian velocity and force requirements in a dual-arm manipulation task [9]. Note that both [9] and [10] predetermined the task-dependent robot manipulability, which required a meticulous and demanding analysis about the motion that the robot needed to perform, which becomes impractical when the robot is required to carry out a large set of different tasks.

In this paper we introduce the novel idea that manipulability-based posture variation for task compatibility can be addressed from a robot learning from demonstration perspective. Specifically, we cast this problem as a *manipulability transfer* between a teacher and a learner. The former demonstrates how to perform a task with a desired time-varying manipulability profile, while the latter reproduces the task by exploiting its own redundant kinematic structure so that its manipulability ellipsoid matches the demonstration. Unlike classical learning frameworks that encode reference position, velocity and force trajectories, our approach offers the possibility of transferring posture-dependent task requirements such as preferred directions for motion and force exertion in operational space, which are encapsulated in the demonstrated manipulability ellipsoids.

This idea opens two main challenges, namely, *(i)* how to encode and retrieve manipulability ellipsoids, and *(ii)* how to exploit the robot redundancy to match the desired time-varying manipulability during a manipulation task. To address the former problem, we propose a tensor-based formulation of Gaussian mixture model (GMM) and Gaussian mixture regression (GMR) that take into account that

manipulability ellipsoids lie on the manifold of symmetric positive definite (SPD) matrices (see Section II for a full description of the model). The latter challenge is solved through manipulability-based redundancy resolution, where we employ a null-space posture optimization built on a cost function that measures the distance between two manipulability ellipsoids in the SPD manifold (see Section III for details). We evaluate the functionality of the proposed approach in a simulated tracking task where a robot is required to track a Cartesian position trajectory and a time-varying desired manipulability profile (see Section IV).

## II. LEARNING MANIPULABILITY ELLIPSOIDS

The first open problem in manipulability transfer is to appropriately encode and retrieve manipulability ellipsoids. In order to describe how we tackle this problem, let us first formally introduce the manipulability ellipsoids concept, followed by the mathematical formulation of a Gaussian mixture model that encodes a distribution of manipulability ellipsoids over the manifold of SPD matrices. After, we describe how desired manipulability ellipsoids can be retrieved via Gaussian mixture regression acting on the SPD manifold.

### A. Manipulability ellipsoids

Velocity and force manipulability ellipsoids introduced in [3] are kinetostatic performance measures of robotic platforms. They indicate the preferred directions in which force or velocity control commands may be performed at a given joint configuration. More specifically, the velocity manipulability ellipsoid describes the characteristics of feasible motion in Cartesian space corresponding to all the unit norm joint velocities. The velocity manipulability of an $n$-DOF robot can be found by using the kinematic relationship between task velocities $\dot{x}$ and joint velocities $\dot{q}$,

$$\dot{x} = J(q)\dot{q}, \tag{1}$$

where $q \in \mathbb{R}^n$ and $J \in \mathbb{R}^{6 \times n}$ are the joint position and Jacobian of the robot, respectively. Moreover, consider the set of joint velocities of constant (unit) norm $\|\dot{q}\|^2 = 1$ describing the points on the surface of a hypersphere in the joint velocity space, which is mapped into the Cartesian velocity space $\mathbb{R}^6$ with

$$\begin{aligned} \|\dot{q}\|^2 &= \dot{q}^{\top}\dot{q} \\ &= \dot{x}^{\top}(J^{\dagger})^{\top}J^{\dagger}\dot{x} \\ &= \dot{x}^{\top}(JJ^{\top})^{\dagger}\dot{x}, \end{aligned} \tag{2}$$

where † indicates the pseudo-inverse of a matrix.[1] Equation (2) represents the robot manipulability in terms of motion, indicating the flexibility of the manipulator in generating velocities in Cartesian space.[2] Note that the major axis of the velocity manipulability ellipsoid $\Upsilon = (JJ^{\top})^{\dagger}$ indicates the direction in which the greater velocity can be generated,

[1]Note that an additional scaling of the joint velocities may be included to consider actuator boundaries.

[2]Dually, the force manipulability ellipsoid can be computed from the static relationship between joint torques and Cartesian forces [3].
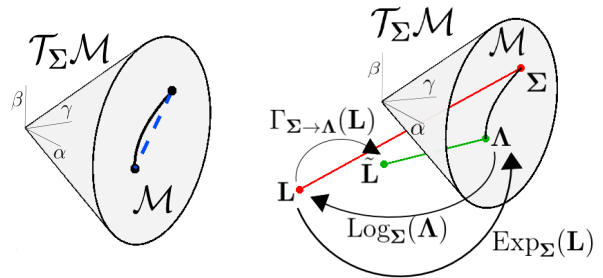


Fig. 1: Representations of the SPD manifold $\mathcal{S}^2_{++}$ embedded in its tangent space $\text{Sym}^2$. One point on the graph corresponds to a matrix $\begin{pmatrix} \alpha & \beta \\ \beta & \gamma \end{pmatrix} \in \text{Sym}^2$. Points in the manifold and tangent space are represented respectively by black and colored dots. *Left*: difference between a geodesic (black curve) and an Euclidean path (blue dash line) between two SPD matrices. *Right*: exponential, logarithm maps and parallel transportation on $\mathcal{S}^2_{++}$. Points on the tangent space of $\Sigma$ and $\Lambda$ are represented respectively by red and green dots.

which is also the direction in which the robot is more sensitive to perturbations. To better understand this, one needs to consider that the force manipulability has the same principal axes of the velocity manipulability, but with lengths (i.e., eigenvalues) that are inversely proportional because of the duality of velocity and force (see [4] for details).

Note that $\Upsilon$ belongs to the set of symmetric positive definite matrices $\mathcal{S}^D_{++}$ which describe the interior of the convex cone, and consequently the manipulability transfer framework must consider these particular characteristics in order to encode and reproduce proper manipulability ellipsoids. To successfully do so, we propose a geometry-aware formulation of both GMM and GMR as described below.

### B. Manifolds of symmetric positive definite (SPD) matrices

The set of $D \times D$ SPD matrices $\mathcal{S}^D_{++}$ is not a vector space since it is not closed under addition and scalar product [11], and thus the use of classical Euclidean space methods for treating and analyzing these matrices is inadequate. A compelling solution is to endow these matrices with a Riemannian metric so that these form a Riemannian manifold.[3] This metric permits to define lengths of curves in the manifold. These curves, called geodesics, are the generalization of straight lines to Riemannian manifolds. Similarly to straight lines in Euclidean space, geodesics are the minimum length curves between two points on the manifold (see Fig. 1).

A Riemannian manifold $\mathcal{M}$ is a mathematical space for which each point locally resembles a Euclidean space. For each point $p \in \mathcal{M}$, there exists a tangent space $\mathcal{T}_p\mathcal{M}$ equipped with a positive definite inner product. In the case of the SPD manifold, the tangent space at any point $\Sigma \in \mathcal{S}^D_{++}$ is identified by the space of symmetric matrices $\text{Sym}^D$. The space of SPD matrices can be represented as the interior of a convex cone embedded in its tangent space $\text{Sym}^D$ (see Fig. 1). Note that the existence of tangent spaces is what

[3]The original cone of SPD matrices has been changed into a regular and complete (but curved) manifold with an infinite development in each of its $D(D + 1)/2$ directions [11].

allows us to carry out tractable statistics on $\mathcal{M}$. To utilize these tangent spaces, we need mappings back and forth between $\mathcal{T}_p\mathcal{M}$ and $\mathcal{M}$, which are known as exponential and logarithm maps.

The exponential map $\text{Exp}_{\boldsymbol{\Sigma}} : \mathcal{T}_{\boldsymbol{\Sigma}}\mathcal{M} \to \mathcal{M}$ maps a point $\boldsymbol{L}$ in the tangent space to a point $\boldsymbol{\Lambda}$ on the manifold, so that it lies on the geodesic starting at $\boldsymbol{\Sigma}$ in the direction $\boldsymbol{L}$ and such that the distance between $\boldsymbol{\Sigma}$ and $\boldsymbol{\Lambda}$ is equal to the distance between $\boldsymbol{\Sigma}$ and $\boldsymbol{L}$. The inverse map is called the logarithm map $\text{Log}_{\boldsymbol{\Sigma}} : \mathcal{M} \to \mathcal{T}_{\boldsymbol{\Sigma}}\mathcal{M}$ (see Fig. 1). Specifically, the exponential and logarithm maps on the SPD manifold are computed as (see [11] for details)

$$\boldsymbol{\Lambda} = \text{Exp}_{\boldsymbol{\Sigma}}(\boldsymbol{L}) = \boldsymbol{\Sigma}^{\frac{1}{2}} \exp(\boldsymbol{\Sigma}^{-\frac{1}{2}} \boldsymbol{L} \boldsymbol{\Sigma}^{-\frac{1}{2}}) \boldsymbol{\Sigma}^{\frac{1}{2}}, \qquad (3)$$

$$\boldsymbol{L} = \text{Log}_{\boldsymbol{\Sigma}}(\boldsymbol{\Lambda}) = \boldsymbol{\Sigma}^{\frac{1}{2}} \log(\boldsymbol{\Sigma}^{-\frac{1}{2}} \boldsymbol{\Lambda} \boldsymbol{\Sigma}^{-\frac{1}{2}}) \boldsymbol{\Sigma}^{\frac{1}{2}}. \qquad (4)$$

Another useful operation over manifolds is the parallel transport $\Gamma_{\boldsymbol{\Sigma} \to \boldsymbol{\Lambda}} : \mathcal{T}_{\boldsymbol{\Sigma}}\mathcal{M} \to \mathcal{T}_{\boldsymbol{\Lambda}}\mathcal{M}$, which moves elements between tangent spaces such that the angle between two elements in the tangent space remains constant (see Fig. 1). The parallel transport of $\boldsymbol{V} \in \mathcal{T}_{\boldsymbol{\Sigma}}\mathcal{S}_{++}^D$ to $\mathcal{T}_{\boldsymbol{\Lambda}}\mathcal{S}_{++}^D$ is given by

$$\Gamma_{\boldsymbol{\Sigma} \to \boldsymbol{\Lambda}}(\boldsymbol{V}) = \boldsymbol{A}_{\boldsymbol{\Sigma} \to \boldsymbol{\Lambda}} \, \boldsymbol{\Sigma} \, \boldsymbol{A}_{\boldsymbol{\Sigma} \to \boldsymbol{\Lambda}}^\top, \qquad (5)$$

with $\boldsymbol{A}_{\boldsymbol{\Sigma} \to \boldsymbol{\Lambda}} = \boldsymbol{\Sigma}^{\frac{1}{2}} \exp(\frac{1}{2} \boldsymbol{\Sigma}^{-\frac{1}{2}} \boldsymbol{V} \boldsymbol{\Sigma}^{-\frac{1}{2}}) \boldsymbol{\Sigma}^{-\frac{1}{2}}$ (see [12] for details). This operation is exploited when it is necessary to move SPD matrices along a curve on the nonlinear manifold.

Notice that SPD matrices, or more broadly, any kind of matrix can be seen as 2nd-order tensor. Tensors are a generalization of matrices to higher order, which allows us to represent the computation of covariance of SPD matrices as a 4th-order covariance tensor $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{D \times D \times D \times D}$ of $N$ matrices $\boldsymbol{X}_n$ that is defined as

$$\boldsymbol{\mathcal{S}} = \frac{1}{N-1} \sum_{n=1}^N \boldsymbol{X}_n \otimes \boldsymbol{X}_n, \qquad (6)$$

where $\otimes$ denotes the tensor product between two tensors, which is a generalization of the outer product to tensors.

As described in [13], we denote element $(p, q, i, j)$ of a 4th-order tensor $\boldsymbol{\mathcal{S}}$ by $\boldsymbol{\mathcal{S}}_{pq}^{ij}$ with two covariant indices $p$, $q$ and two contravariant indices $i$, $j$. An element $(i, j)$ of a matrix $\boldsymbol{X}$ is denoted by $\boldsymbol{X}_{ij}$ with two covariant indices $i$, $j$. A tensor contraction between two tensors is performed when one or more contravariant and covariant indices are identical. For example, the tensor contraction of $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{D \times D \times D \times D}$ and $\boldsymbol{X} \in \mathbb{R}^{D \times D}$ is written as

$$\boldsymbol{\mathcal{S}} \boldsymbol{X} = \sum_{i=1}^D \sum_{j=1}^D \boldsymbol{\mathcal{S}}_{pq}^{ij} \boldsymbol{X}_{ij}. \qquad (7)$$

Using 4th-order covariance tensors, the tensor-variate normal distribution of a random 2nd-order tensor $\boldsymbol{X}$ with mean $\boldsymbol{M}$ and covariance $\boldsymbol{\mathcal{S}}$ is expressed by [14]

$$\mathcal{N}(\boldsymbol{X}|\boldsymbol{M}, \boldsymbol{\mathcal{S}}) = \frac{1}{\sqrt{(2\pi)^{\tilde{D}}|\boldsymbol{\mathcal{S}}|}} \, e^{-\frac{1}{2}(\boldsymbol{X}-\boldsymbol{M})\boldsymbol{\mathcal{S}}^{-1}(\boldsymbol{X}-\boldsymbol{M})}, \quad (8)$$

where the scalar value $(\boldsymbol{X} - \boldsymbol{M})\boldsymbol{\mathcal{S}}^{-1}(\boldsymbol{X} - \boldsymbol{M})$ is computed using the tensor contraction defined previously, and $\tilde{D} = D + D(D-1)/2$.

## C. Gaussian Mixture Model on SPD manifolds

Similarly to multivariate distribution (see [15], [16], [17]), a tensor-variate distribution maximizing the entropy in the tangent space is approximated by

$$\mathcal{N}_{\mathcal{M}}(\boldsymbol{X}|\boldsymbol{M}, \boldsymbol{\mathcal{S}}) = \frac{1}{\sqrt{(2\pi)^{\tilde{D}}|\boldsymbol{\mathcal{S}}|}} \, e^{-\frac{1}{2}\text{Log}_{\boldsymbol{X}}(\boldsymbol{M}) \, \boldsymbol{\mathcal{S}}^{-1} \, \text{Log}_{\boldsymbol{X}}(\boldsymbol{M})}, \tag{9}$$

where $\boldsymbol{X} \in \mathcal{T}_{\boldsymbol{M}}\mathcal{M}$, $\boldsymbol{M} \in \mathcal{M}$ is the origin in the tangent space and $\boldsymbol{\mathcal{S}} \in \mathcal{T}_{\boldsymbol{M}}\mathcal{M}$ is the covariance tensor.

Similarly to the Euclidean case, a GMM on the SPD manifold is defined by

$$p(\boldsymbol{X}) = \sum_{k=1}^K \pi_k \mathcal{N}_{\mathcal{M}}(\boldsymbol{X}|\boldsymbol{M}_k, \boldsymbol{\mathcal{S}}_k), \qquad (10)$$

with $K$ being the number of components of the model, and $\pi_k$ representing the priors such that $\sum_k \pi_k = 1$. The parameters of the GMM on the manifold are estimated by Expectation-Maximization (EM) algorithm.

The responsibility of each component $k$ is computed in the E-step as:

$$p(k|\boldsymbol{X}_i) = \frac{\pi_k \, \mathcal{N}_{\mathcal{M}}(\boldsymbol{X}_i|\boldsymbol{M}_k, \boldsymbol{\mathcal{S}}_k)}{\sum_{j=1}^K \pi_j \, \mathcal{N}_{\mathcal{M}}(\boldsymbol{X}_i|\boldsymbol{M}_j, \boldsymbol{\mathcal{S}}_j)}, \qquad (11)$$

$$N_k = \sum_{i=1}^N p(k|\boldsymbol{X}_i). \qquad (12)$$

During the M-step, the mean $\boldsymbol{M}_k$ is first updated iteratively until convergence for each component. The covariance tensor $\boldsymbol{\mathcal{S}}_k$ and prior $\pi_k$ are then updated using the new mean:

$$\boldsymbol{M}_k \leftarrow \frac{1}{N_k} \text{Exp}_{\boldsymbol{M}_k} \left( \sum_{i=1}^N p(k|\boldsymbol{X}_i) \, \text{Log}_{\boldsymbol{M}_k}(\boldsymbol{X}_i) \right), \qquad (13)$$

$$\boldsymbol{\mathcal{S}}_k \leftarrow \frac{1}{N_k} \sum_{i=1}^N p(k|\boldsymbol{X}_i) \, \text{Log}_{\boldsymbol{M}_k}(\boldsymbol{X}_i) \otimes \text{Log}_{\boldsymbol{M}_k}(\boldsymbol{X}_i), \quad (14)$$

$$\pi_k \leftarrow \frac{N_k}{N}. \qquad (15)$$

## D. Gaussian Mixture Regression on SPD manifolds

GMR computes the conditional distribution $p(\boldsymbol{X}_{\mathcal{O}\mathcal{O}}|\boldsymbol{X}_{\mathcal{I}\mathcal{I}})$ of the joint distribution $p(\boldsymbol{X})$, where the sub-indices $\mathcal{I}$ and $\mathcal{O}$ denote the sets of dimensions that span the input and output variables. We use the following block decomposition of the datapoints, means and covariances:

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{X}_{\mathcal{I}\mathcal{I}} & 0 \\ 0 & \boldsymbol{X}_{\mathcal{O}\mathcal{O}} \end{pmatrix}, \boldsymbol{M} = \begin{pmatrix} \boldsymbol{M}_{\mathcal{I}\mathcal{I}} & 0 \\ 0 & \boldsymbol{M}_{\mathcal{O}\mathcal{O}} \end{pmatrix},$$

$$\boldsymbol{\mathcal{S}} = \left( \begin{array}{cc|cc} \boldsymbol{\mathcal{S}}_{\mathcal{I}\mathcal{I}}^{\mathcal{I}\mathcal{I}} & 0 & 0 & 0 \\ 0 & \boldsymbol{\mathcal{S}}_{\mathcal{I}\mathcal{I}}^{\mathcal{O}\mathcal{O}} & 0 & 0 \\ \hline 0 & 0 & \boldsymbol{\mathcal{S}}_{\mathcal{O}\mathcal{O}}^{\mathcal{I}\mathcal{I}} & 0 \\ 0 & 0 & 0 & \boldsymbol{\mathcal{S}}_{\mathcal{O}\mathcal{O}}^{\mathcal{O}\mathcal{O}} \end{array} \right), \qquad (16)$$

where we represent the 4th-order tensor by separating the different fibers with bars. With this decomposition, manifold

functions can be applied individually on input and output parts, for example

$$\text{Exp}_{\boldsymbol{M}_k}(\boldsymbol{X}) = \begin{pmatrix} \text{Exp}_{\boldsymbol{M}_{\mathcal{II}}}(\boldsymbol{X}_{\mathcal{II}}) & \boldsymbol{0} \\ \boldsymbol{0} & \text{Exp}_{\boldsymbol{M}_{\mathcal{OO}}}(\boldsymbol{X}_{\mathcal{OO}}) \end{pmatrix}.$$

Similarly to GMR in Euclidean space [18] and in manifolds where data are represented by vectors [15], GMR on SPD manifold approximates the conditional distribution by a single Gaussian

$$p(\boldsymbol{X}_{\mathcal{OO}}|\boldsymbol{X}_{\mathcal{II}}) \sim \mathcal{N}(\hat{\boldsymbol{M}}_{\mathcal{OO}}, \hat{\boldsymbol{\mathcal{S}}}_{\mathcal{OO}}^{\mathcal{OO}}), \tag{17}$$

where the mean $\hat{\boldsymbol{M}}_{\mathcal{OO}}$ is computed iteratively until convergence in its tangent space using

$$\boldsymbol{\Delta}_k = \text{Log}_{\hat{\boldsymbol{M}}_{\mathcal{OO}}}(\boldsymbol{M}_{\mathcal{OO},k}) - \tilde{\boldsymbol{\mathcal{S}}}_{\mathcal{OO},k}^{\mathcal{II}} \tilde{\boldsymbol{\mathcal{S}}}_{\mathcal{II},k}^{\mathcal{II}-1} \text{Log}_{\boldsymbol{X}_{\mathcal{II}}}(\boldsymbol{M}_{\mathcal{II},k}), \tag{18}$$

$$\hat{\boldsymbol{M}}_{\mathcal{OO}} \leftarrow \text{Exp}_{\hat{\boldsymbol{M}}_{\mathcal{OO}}}\Big(\sum_k h_k \boldsymbol{\Delta}_k\Big), \tag{19}$$

with $h_k$ describing the responsibilities of the GMM components in the regression, namely

$$h_k = \frac{\pi_k \mathcal{N}(\boldsymbol{X}_{\mathcal{II}}|\boldsymbol{M}_{\mathcal{II},k}, \boldsymbol{\mathcal{S}}_{\mathcal{II},k}^{\mathcal{II}})}{\sum_{j=1}^K \pi_j \mathcal{N}(\boldsymbol{X}_{\mathcal{II}}|\boldsymbol{M}_{\mathcal{II},j}, \boldsymbol{\mathcal{S}}_{\mathcal{II},j}^{\mathcal{II}})}. \tag{20}$$

The covariance $\hat{\boldsymbol{\mathcal{S}}}_{\mathcal{OO}}^{\mathcal{OO}}$ is then computed in the tangent space of the mean

$$\hat{\boldsymbol{\mathcal{S}}}_{\mathcal{OO}}^{\mathcal{OO}} = \sum_k h_k \Big( \tilde{\boldsymbol{\mathcal{S}}}_{\mathcal{OO},k}^{\mathcal{OO}} - \tilde{\boldsymbol{\mathcal{S}}}_{\mathcal{OO},k}^{\mathcal{II}} \tilde{\boldsymbol{\mathcal{S}}}_{\mathcal{II},k}^{\mathcal{II}-1} \tilde{\boldsymbol{\mathcal{S}}}_{\mathcal{II},k}^{\mathcal{OO}} + \boldsymbol{\Delta}_k \otimes \boldsymbol{\Delta}_k \Big), \tag{21}$$

where $\tilde{\boldsymbol{\mathcal{S}}}$ is the parallel transported covariance tensor

$$\tilde{\boldsymbol{\mathcal{S}}} = \Gamma_{\boldsymbol{M} \to \hat{\boldsymbol{X}}}(\boldsymbol{\mathcal{S}}) \quad \text{with} \quad \hat{\boldsymbol{X}} = \begin{pmatrix} \boldsymbol{X}_{\mathcal{II}} & \boldsymbol{0} \\ \boldsymbol{0} & \hat{\boldsymbol{M}}_{\mathcal{OO}} \end{pmatrix}. \tag{22}$$

Note that the definition of the tangent space $\mathcal{T}_{\boldsymbol{p}}\mathcal{M}$ (which has the structure of a Euclidean vector space) is what allow us to compute the conditional distribution above. Also notice that to parallel transport a 4th-order covariance tensor $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{D \times D \times D \times D}$, the covariance is first converted to a 2nd-order tensor $\boldsymbol{\Sigma} \in \mathbb{R}^{\tilde{D} \times \tilde{D}}$ with $\tilde{D} = D + D(D-1)/2$, as proposed in [14]. We can then compute its eigentensors $\boldsymbol{V}_k$, which are used to parallel transport the covariance matrix between tangent spaces [19]. Let $\tilde{\boldsymbol{V}}_k = \Gamma_{\boldsymbol{M} \to \hat{\boldsymbol{X}}}(\boldsymbol{V}_k)$ be the $k$-th parallel transported eigentensor with (5) and $\lambda_k$ the $k$-th eigenvalue. The parallel transported 4th-order covariance tensor is then obtained with (see [20] for more details)

$$\Gamma_{\boldsymbol{M} \to \hat{\boldsymbol{X}}}(\boldsymbol{\mathcal{S}}) = \sum_k \lambda_k \tilde{\boldsymbol{V}}_k \otimes \tilde{\boldsymbol{V}}_k. \tag{23}$$

## III. MANIPULABILITY-BASED OPTIMAL REDUNDANCY RESOLUTION

Given a desired manipulability ellipsoid profile and desired reference trajectories in the form of Cartesian position or force, the goal of the robot is to reproduce the task by tracking these reference trajectories while exploiting its redundancy to maximize the match between the current manipulability ellipsoid and the desired one. In this situation,
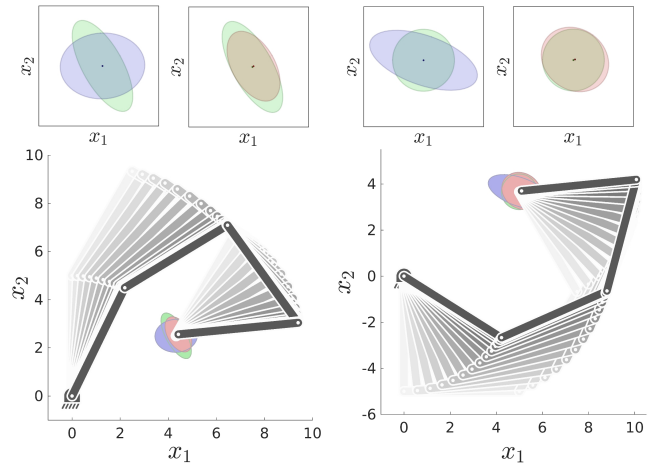


Fig. 2: Illustration of the manipulability-based redundancy resolution. Two different cases are shown where both initial robot configuration and desired manipulability change. The robot color goes from light gray to black to show the evolution of the posture. Initial, final, and desired manipulability ellipsoids are respectively depicted in blue, red, and green. The *top* row shows close-up plots corresponding to the initial and final manipulability ellipsoids (left and right graphs, respectively).

we assume that the tracking task is a high-priority objective while the manipulability ellipsoid transfer is assigned a secondary role. The major difficulty in this problem is the selection of the appropriate criterion for the null-space posture optimization process.

Formally, consider a redundant robot with rigid body dynamics equations given in the form

$$\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{b}(\boldsymbol{q}) = \boldsymbol{\tau}, \tag{24}$$

where $\boldsymbol{H}(\boldsymbol{q})$, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and $\boldsymbol{b}(\boldsymbol{q})$ are the inertia matrix, the vector of centrifugal and Coriolis forces, and the gravity components, respectively. Additionally, the pose of the robot in joint space is denoted by $\boldsymbol{q}$, while $\boldsymbol{\tau}$ represents the actuation torques vector. Let us also define the desired joint torque vector $\boldsymbol{\tau}_d$ as

$$\boldsymbol{\tau}_d = \boldsymbol{J}^\top \boldsymbol{F} + (\boldsymbol{I} - \boldsymbol{J}^\top \bar{\boldsymbol{J}}^\top)\boldsymbol{\tau}_N, \tag{25}$$

where $\boldsymbol{J}$ is the Jacobian of the robot, $\boldsymbol{F}$ is the end-effector force designed to track the reference Cartesian trajectories, $\bar{\boldsymbol{J}}$ is the inertia-weighted pseudoinverse of $\boldsymbol{J}$, and $\boldsymbol{\tau}_N$ a desired null-space torque vector [21]. The right-hand side of (25) does not produce any force, and thus motion, in the task space, and allows us to find a $\boldsymbol{\tau}_N$ such that the internal motion will move the robot towards a posture where the match between its manipulability ellipsoid and the desired one is maximum. In order to do so, $\boldsymbol{\tau}_N$ is designed to minimize a cost function that measures the similarity between manipulability ellipsoids.

In this paper we propose to use a geometry-aware criterion for computing the desired null-space vector. Specifically, we define a cost function

$$g_t(\boldsymbol{q}) = \log \det \left( \frac{\hat{\boldsymbol{\Upsilon}}_t + \boldsymbol{\Upsilon}_t(\boldsymbol{q})}{2} \right) - \frac{1}{2} \log \det \left( \hat{\boldsymbol{\Upsilon}}_t \boldsymbol{\Upsilon}_t(\boldsymbol{q}) \right), \tag{26}$$
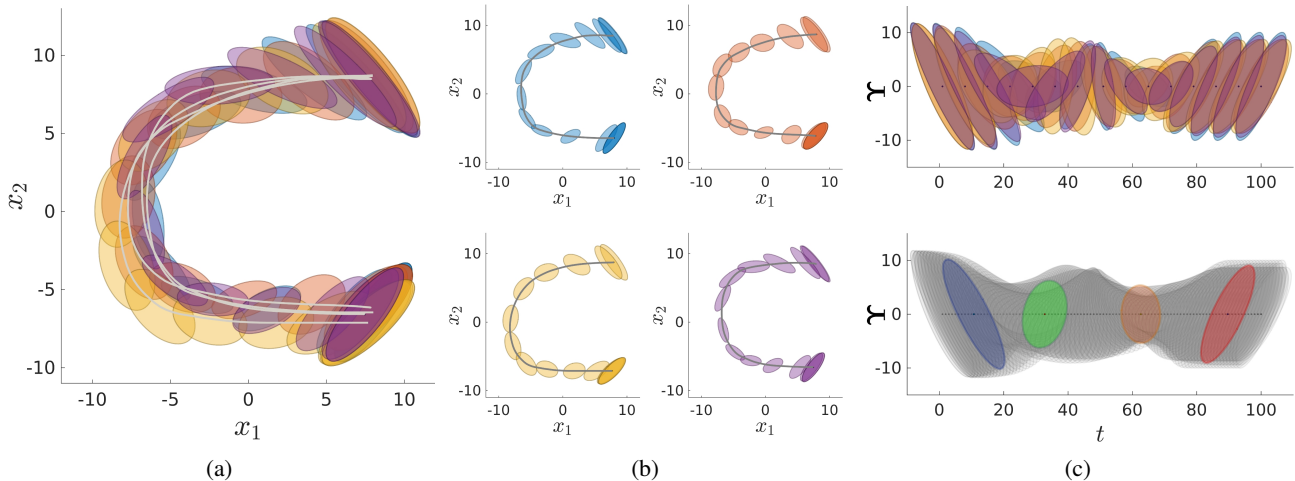
Fig. 3: Four demonstrations of a 3-DOF planar robot tracking a C-shape trajectory. (*a*) shows the path followed by the robot end-effector (light gray solid lines) and the manipulability ellipsoids at different time steps for all the demonstrations. (*b*) displays the demonstrations independently to exhibit more clearly the differences produced in the manipulability ellipsoids due to the robot posture variation across demonstrations. (*c*) shows the demonstrated manipulability ellipsoids over time (*top* in color, *bottom* in gray), and the centers $M_k$ of the 4-states GMM in the SPD manifold (*bottom*). Position $x$ and time $t$ are given in centimeters and seconds, respectively.

where $\hat{\Upsilon}_t$ is the desired manipulability ellipsoid retrieved by GMR at each time step $t$ (using (18) and (19)), and $\Upsilon_t(q)$ denotes the current robot manipulability ellipsoid. Note that (26) describes a distance-like function on the open cone of SPD matrices, which was introduced in [22] as the Stein divergence. The squared root of this function works as a metric on SPD matrices and is preferred over the classical Riemannian distance because it is cheaper to compute, and so is its derivative. Moreover, (26) still considers the non-Euclidean geometry of $\mathcal{S}_{++}^D$, therefore being consistent with the learning framework introduced in Section II.

Then, the manipulability-based redundancy resolution is carried out by computing a null-space torque vector $\tau_N$ that is proportional to the negative gradient of (26), that is $\tau_N = -\alpha\nabla g_t(q)$, which leads to the desired joint torques

$$\tau_d = J^\top F - (I - J^\top\bar{J}^\top)\alpha\nabla g_t(q). \quad (27)$$

Note that other redundancy resolution schemes at velocity and acceleration levels can alternatively be used for controlling the robot [21].

To show the effectiveness of the proposed redundancy resolution based on manipulability ellipsoids, we carried out experiments where a simulated 4-DOF planar robot is required to keep its end-effector at a fixed Cartesian position while moving its joints to make its manipulability ellipsoid coincide with that of the demonstration. Figure 2 shows how the manipulator configuration is successfully adjusted so that $g_t(q)$ is minimum, that is $\Upsilon_t(q) \approx \hat{\Upsilon}_t$. These results show that Stein divergence is a suitable criterion for driving the manipulability-based redundancy resolution. This method, combined with the learning framework introduced in Section II, makes manipulability transfer possible as shown next.

## IV. EXPERIMENTS

In this section we illustrate and evaluate the proposed approach using a couple of simulated planar robots with dissimilar embodiments and a different number of articulations. The central idea is to teach a redundant robot to track a reference trajectory in Cartesian space with a desired time-varying manipulability ellipsoid. The description of the experiment, the obtained results and the corresponding discussion are given in the following paragraphs. Moreover, source codes of the proposed framework are available at https://gitlab.idiap.ch/rli/pbdlib-matlab/.

### A. Description

For the demonstration, a 3-DOF *teacher* robot executes the tracking task four times, from which we extract both the end-effector position $x_t$ and robot manipulability ellipsoid $\Upsilon_t(q)$, at each time step $t$. The collected time-aligned data is split in two datasets, namely, $X^x \in \mathbb{R}^{D+1\times T}$ and $X^\Upsilon \in \mathbb{R}^{D+1\times D+1\times T}$, where $X^x$ and $X^\Upsilon$ denote the position and manipulability training datasets, with $D=2$ as the dimensionality of the task space, and $T$ the total number of observations. A datapoint of each dataset is described as

$$X_t^x = \begin{pmatrix} t \\ x_t \end{pmatrix} \quad \text{and} \quad X_t^\Upsilon = \begin{pmatrix} t & \mathbf{0} \\ \mathbf{0} & \Upsilon_t(q) \end{pmatrix}.$$

We trained a classical GMM over the time-driven Cartesian trajectories $X^x$ and a GMM over the manifold of time-driven manipulability ellipsoids $X^\Upsilon$, using models with four components, i.e. $K = 4$ (the number was selected by the experimenter).

During the reproduction phase, a 5-DOF *student* robot executes the time-driven task by following a desired Cartesian trajectory $\hat{x}_t$ computed from a classical GMR as $\hat{x}_t \sim \mathcal{P}(x|t)$. The robot implements a proportional-derivative controller in task space that defines the desired end-effector force $F$ in (27). The robot also varies its joint configuration according to (27) for matching desired manipulability ellipsoids $\hat{\Upsilon}_t \sim \mathcal{P}(\Upsilon|t)$, estimated by GMR over the SPD manifold (as explained in Section II-D).
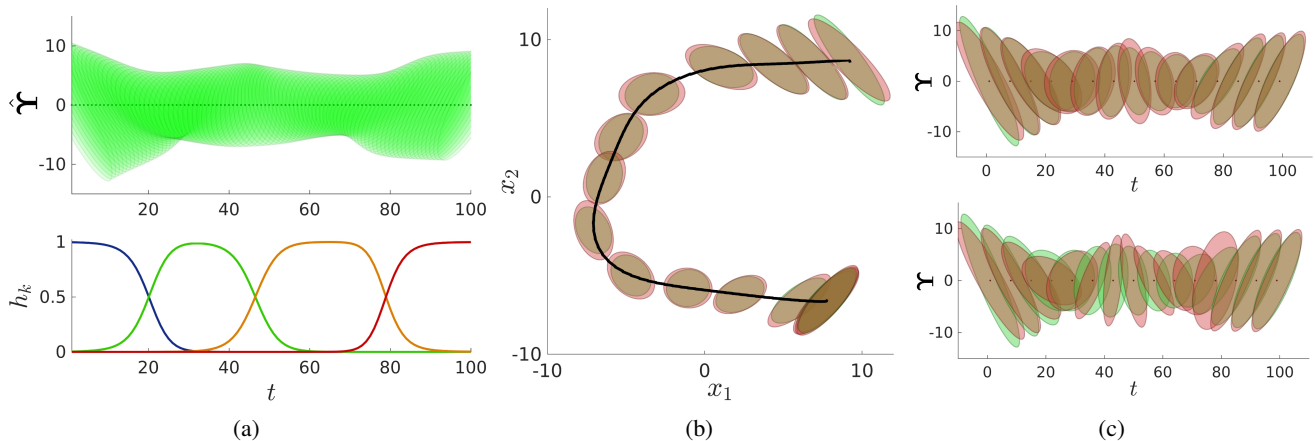
Fig. 4: Reproduction of a C-shape tracking task with desired robot manipulability. (*a*) Desired manipulability ellipsoids profile estimated by GMR (*top*), and influence of GMM components on the time-driven GMR estimates (*bottom*), where colors match the distributions shown in Fig. 3c. (*b*) Executed Cartesian trajectory (black solid line) given in centimeters, desired and reproduced manipulability ellipsoids (in green and red, respectively), at different time steps of the task (reference trajectory $\hat{x}$ coincides with the executed trajectory). (*c*) Desired and reproduced manipulability ellipsoids over time (given in seconds). The *top* plot shows the manipulability ellipsoids match resulting from the proposed approach. The *bottom* graph displays the match using the major-axis alignment method in [23].

## B. Results

Figure 3 shows the four demonstrations carried out by the 3-DOF robot, where both the Cartesian trajectory and manipulability ellipsoids are displayed. Note that the recorded manipulability ellipsoids slightly change across demonstrations as a side effect of the variation observed in both the initial end-effector position and the generated trajectory. Figure 3c-*bottom* displays, over time, the demonstrated ellipsoids (in gray) along with the center $M_k$ of the four components of the GMM encoding $X^{\Upsilon}$. Notice how the first and fourth GMM components (in blue and red, respectively) satisfactorily encapsulate the pattern observed in the robot manipulability at the beginning and end of the demonstrations, in both shape and orientation.

A successful reproduction of the tracking task with manipulability-based redundancy resolution is shown in Fig. 4. Note that the robot is able not only to track the desired C-shape trajectory, but also to change its posture so that its manipulability ellipsoid matches that of the demonstration, in both shape and orientation (see Figs 4b and 4c-*top*). These results validate that the proposed approach allows the robot to learn and reproduce reference trajectories while fulfilling additional task requirements encapsulated in a profile of desired manipulability ellipsoids.

In order to show the importance of the proposed geometry-aware cost function for the redundancy resolution, we compared our approach against a method that relies on the alignment of the ellipsoids major axis for reproducing desired Cartesian stiffness ellipsoids [23]. Figure 4c shows the desired and reproduced manipulability ellipsoids using our approach (*top* plot) and the major-axis alignment method (*bottom* plot). Note that despite the major-axis alignment method retrieves ellipsoids that are approximately aligned with the desired manipulability, their shapes significantly differ for some periods of time (e.g. between time steps 40 and 80). This discrepancy may considerably affect the

robot performance due to the task requirements encapsulated in the desired manipulability not being fulfilled precisely. In contrast, our approach offers a more accurate matching in both shape and orientation because the cost function (26) takes into account the geometry of manipulability by measuring the divergence in the SPD manifold. Moreover, the manipulability-based approach shows a faster convergence than the major-axis alignment method, as shown in Fig. 5, where the $\alpha$ value was set to provide the fastest convergence rate for each algorithm, while ensuring stable reproductions starting from the same joint configuration. This faster convergence can be attributed to the fact that the gradient of the manipulability-based criterion (26) is more informative about the joint motions, favoring an accurate matching between manipulability ellipsoids.

## C. Discussion

The reported results show the effectiveness of the proposed approach for transferring manipulability ellipsoids between robots that differ in their kinematic structure. Our framework allows a robot to learn posture-dependent task requirements without explicitly encoding a model in the joint space of the demonstrator, which would require complex kinematic mapping algorithms and would make task analysis less interpretable at first sight. More importantly, the proposed framework extends the robot learning capability beyond the transfer of trajectory, force and impedance.

Manipulability transfer, as proposed in this paper, can make use of the different manipulability criteria proposed in the literature, such as velocity/force/dynamic manipulability ellipsoids [24], in order to learn and reproduce task requirements at different kinematics and dynamics levels. For example, for a specific task, the shape and orientation of velocity and force ellipsoids may vary over time, and so may the optimal directions for controlling velocity and force. Such variations can be compactly encoded in our approach, which may be later exploited to define optimal directions for
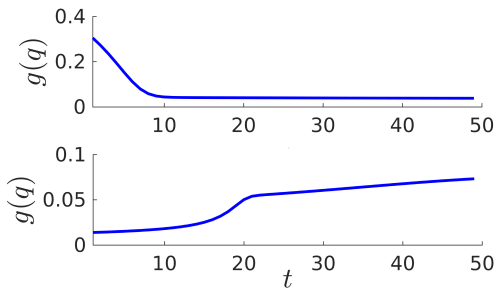
Fig. 5: Convergence rate for redundancy resolution. The *top* graph shows the convergence for the proposed manipulability-based cost function, while the *bottom* graph displays the results using the maximization of the major-axis alignment criterion.

controlling velocity and force in a hybrid-controlled robot to successfully fulfill the velocity/force task requirements. In other words, our method may be used to assist in the design of task control strategies for robots [25].

Note that during reproduction, we did not take advantage of the variability observed in the demonstrated manipulability ellipsoids associated with $\hat{\mathcal{S}}_{\circ\circ}^{\circ\circ}$ (see (17) and (21)). Such information, in the form of full covariance (4th-order tensor), could potentially be exploited in future work to characterize the precision of the manipulability-based redundancy optimization, where low variability would demand high precision in ellipsoids matching, and vice-versa.

## V. CONCLUSIONS AND FUTURE WORK

This paper presented a novel framework for transferring manipulability ellipsoids to robots. The proposed approach is first built on a probabilistic learning model that allows for the geometry of the SPD manifold to encode and retrieve appropriate manipulability ellipsoids. This geometry-aware approach is later exploited for redundancy resolution, allowing the robot to modify its posture so that its manipulability ellipsoid coincides with that of a demonstration. This approach enables the learning of posture-dependent task requirements. It provides a skill transfer strategy going beyond the imitation of trajectory, force or impedance behaviors. To the best of our knowledge, this is the first work that proposes manipulability transfer between agents of different embodiments from a robot learning from demonstration perspective.

Future work will explore manipulability transfer between humans and robots. The tasks of interest will be those in which velocity and force control requirements vary over the course of the task execution, which will be directly related to changes in the velocity/force manipulability ellipsoids. We also envisage to exploit the covariance information associated with the desired manipulability ellipsoids retrieved by GMR to regulate how precisely the manipulability should be matched. This may avoid unnecessary posture changes in parts of the task in which variable manipulability ellipsoids were observed in the demonstrations.

## REFERENCES

[1] I. Cos, N. Bélanger, and P. Cisek, "The influence of predicted arm biomechanics on decision making," *Journal of Neurophysiology*, vol. 105, no. March, pp. 3022–3033, 2011.

[2] P. Sabes and M. Jordan, "Obstacle avoidance and a perturbation sensitivity model for motor planning," *Journal of Neuroscience*, vol. 17, pp. 7119–7128, 1997.

[3] T. Yoshikawa, "Manipulability of robotic mechanisms," *Intl. Journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.

[4] S. Chiu, "Control of redundant manipulators for task compatibility," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 1987, pp. 1718–1724.

[5] P. Chiacchio, "Exploiting redundancy in minimum-time path following robot control," in *American Control Conference*, 1990, pp. 2313–2318.

[6] N. Vahrenkamp, T. Asfour, G. Metta, G. Sandini, and R. Dillmann, "Manipulability analysis," in *IEEE/RAS Intl. Conf. on Humanoid Robots (Humanoids)*, 2012, pp. 568–573.

[7] L. Guilamo, J. Kuffner, K. Nishiwaki, and S. Kagami, "Manipulability optimization for trajectory generation," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2006, pp. 2017–2022.

[8] N. Somani, M. Rickert, A. Gaschler, C. Cai, A. Perzylo, and A. Knoll, "Task level robot programming using prioritized non-linear inequality constraints," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2016, pp. 430–437.

[9] S. Lee, "Dual redundant arm configuration optimization with task-oriented dual arm manipulability," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 1, pp. 78–97, 1989.

[10] I. Lee and J. Oh, "Humanoid posture selection for reaching motion and a cooperative balancing controller," *Journal of Intelligent and Robotics Systems*, vol. 8, no. 3-4, pp. 301–316, 2016.

[11] X. Pennec, P. Fillard, and N. Ayache, "A Riemannian framework for tensor computing," *Intl. Journal on Computer Vision*, vol. 66, no. 1, pp. 41–66, 2006.

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

[13] A. Tyagi and J. W. Davis, "A recursive filter for linear systems on Riemannian manifolds," in *Conf. on Computer Vision and Pattern Recognition (CVPR)*, Anchorage, Alaska, June 2008, pp. 1–8.

[14] P. Basser and S. Pajevic, "Spectral decomposition of a 4th-order covariance tensor: Applications to diffusion tensor MRI," *Signal Processing*, vol. 87, no. 2, pp. 220–236, 2007.

[15] M. J. A. Zeestraten, I. Havoutis, J. Silvério, S. Calinon, and D. G. Caldwell, "An approach for imitation learning on Riemannian manifolds," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1240–1247, June 2017.

[16] E. Simo-Serra, C. Torras, and F. Moreno-Noguer, "3D human pose tracking priors using geodesic mixture models," *Intl. Journal on Computer Vision*, vol. 122, no. 2, pp. 388–408, 2017.

[17] G. Dubbelman, "Intrinsic statistical techniques for robust pose estimation," PhD thesis, University of Amsterdam, Netherlands, 2011.

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

[19] O. Freifeld, S. Hauberg, and M. J. Black, "Model transport: Towards scalable transfer learning on manifolds," in *Conf. on Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, USA, June 2014, pp. 1378–1385.

[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 Intl. Conf. on Intelligent Robots and Systems (IROS)*, Vancouver, Canada, September 2017.

[21] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal, "Comparative experiments on task space control with redundancy resolution," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2005, pp. 3901–3908.

[22] S. Sra, "A new metric on the manifold of kernel matrices with application to matrix geometric means," in *Neural Information Processing Systems (NIPS)*, 2012, pp. 144–152.

[23] A. Ajoudani, N. Tsagarakis, and A. Bicchi, "On the role of robot configuration in cartesian stiffness control," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2015, pp. 1010–1016.

[24] K. Doty, E. Schwartz, C. Melchiorri, and C. Bonivento, "Robot manipulability," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 3, pp. 462–468, 1995.

[25] L. Peternel, L. Rozo, D. G. Caldwell, and A. Ajoudani, "A method for derivation of robot task-frame authority from repeated sensory observations," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 719–726, 2017.