Geometry-aware Manipulability Learning, Tracking and Transfer

Journal Title
XX(X):1–24

DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Noémie Jaquier[1], Leonel Rozo[2,3], Darwin G. Caldwell[3] and Sylvain Calinon[1]

Abstract

Body posture influences human and robots performance in manipulation tasks, as appropriate poses facilitate motion or force exertion along different axes. In robotics, manipulability ellipsoids arise as a powerful descriptor to analyze, control and design the robot dexterity as a function of the articulatory joint configuration. This descriptor can be designed according to different task requirements, such as tracking a desired position or apply a specific force. In this context, this article presents a novel *manipulability transfer* framework, a method that allows robots to learn and reproduce manipulability ellipsoids from expert demonstrations. The proposed learning scheme is built on a tensor-based formulation of a Gaussian mixture model that takes into account that manipulability ellipsoids lie on the manifold of symmetric positive definite matrices. Learning is coupled with a geometry-aware tracking controller allowing robots to follow a desired profile of manipulability ellipsoids. Extensive evaluations in simulation with redundant manipulators, a robotic hand and humanoids agents, as well as an experiment with two real dual-arm systems validate the feasibility of the approach.

Keywords

Robot learning, programming by demonstrations, manipulability ellipsoids, manipulability optimization, Riemannian manifolds, differential kinematics.

1 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 (see Fig. 1). 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 arm kinematics plays a central role when humans plan point-to-point reaching movements, where joint trajectory patterns arise as a function of the visual target (Morasso 1981), indicating that the task requirements influence the human arm posture. This insight was also identified in more complex situations, where not only kinematic but also other biomechanic factors affect the task planning (Cos et al. 2011). For example, the human central nervous system plans arm movements considering its directional sensitivity, which is directly related to the arm posture (Sabes and Jordan 1997). This allows humans to be mechanically resistant to potential perturbations coming from obstacles occupying the workspace. Interestingly, directional preferences of human arm movements are characterized by a tendency to exploit interaction torques for movement production at the shoulder or elbow, indicating that the preferred directions are largely determined by biomechanical factors (Dounskaia et al. 2014).

The robotics community has also been aware of the impact of robot posture on reaching movements and manipulation tasks (e.g., pushing, pulling, reaching). It is well known that by varying the posture of a robot, we can change the optimal directions for generating motion or applying specific forces. This has direct implications in hybrid control, since the controller capability can be fully realized when the optimal directions for controlling velocity and force coincide with those dictated by the task (Chiu 1987). In this context, the so-called manipulability ellipsoid (Yoshikawa 1985b) serves as a geometric descriptor that indicates the ability to arbitrarily perform motion and exert a force along the different task directions in a given joint configuration.

Manipulability ellipsoids have been used to measure the compatibility of robot postures with respect to fine and coarse manipulation (Chiu 1987), and to improve minimum-time trajectory planning using a manipulability-aware inverse kinematics algorithm (Chiacchio 1990). Vahrenkamp et al. (2012) proposed a grasp selection process that favored high manipulability in the robot workspace. Other works have focused on maximizing the manipulability ellipsoid volume in trajectory generation algorithms (Guilamo et al. 2006), and task-level robot programming frameworks (Somani et al. 2016), to obtain singularity-free joint trajectories and high task-space dexterity. Nevertheless, as stated in (Lee 1989), solely maximizing the ellipsoid volume to achieve high dexterity in motion may cause a reverse effect on the flexibility in force.

[1]Idiap Research Institute, CH-1920 Martigny, Switzerland
[2]Bosch Center for Artificial Intelligence, 71272 Renningen, Germany
[3]Department of Advanced Robotics, Istituto Italiano di Tecnologia, 16163 Genova, Italy

Corresponding author:
Noémie Jaquier, Idiap Research Institute, CH-1920 Martigny, Switzerland
Email: noemie.jaquier@idiap.ch

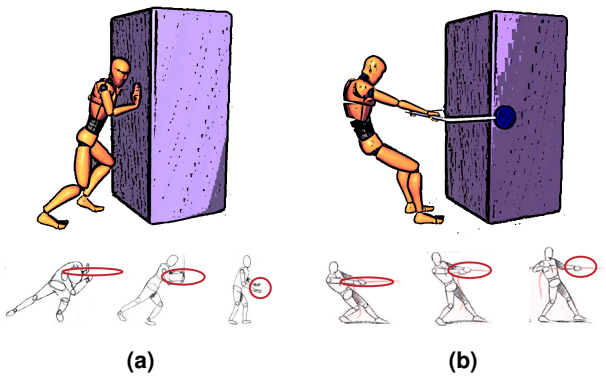


Figure 1. Illustration of pushing (*a*) and pulling (*b*) tasks for which the posture of the human significantly influences his/her ability to carry out the task.

The aforementioned approaches do not specify a desired robot manipulability for the task. In contrast, Lee and Oh (2016) proposed an optimization method to find reaching postures for a humanoid robot that achieved desired (manually-specified) manipulability volumes. Similarly, a series of desired manipulability ellipsoids was predefined according to Cartesian velocity and force requirements in dual-arm manipulation tasks (Lee 1989). Note that both (Lee 1989) and (Lee and Oh 2016) 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. Furthermore, these approaches overlooked an important characteristic of manipulability ellipsoids, namely, the fact that they lie on the manifold of symmetric positive definite (SPD) matrices. This may influence the optimal robot joint configuration for the task at hand.

Other geometric descriptors have been proposed in the literature to evaluate the velocity or force performance of robots at a given joint configuration. In contrast to manipulability ellipsoids that do not fully account for boundary limits in the space of joint velocities or torques, manipulability polytopes provide a linear estimate of the exact joint constraints in task space (Chiacchio et al. 1997; Lee 1997). Moreover, Ajoudani et al. (2015) introduced the concept of stiffness feasibility region (SFR) to represent the non-polytopic boundary where the realization of a desired Cartesian stiffness matrix is feasible. While the polytope approaches provide a more accurate estimate of the velocity or force generation capabilities of the robot compared to manipulability ellipsoids, their calculation is computationally expensive. SFR is a particular Cartesian stiffness descriptor and therefore does not generalize to other robot control settings. Manipulability ellipsoids are easy to compute, while representing an intuitive estimate of the robot ability to perform velocities, accelerations or exert forces along the different task directions.

In this article 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 a sequence of manipulability ellipsoids, and *(ii)* how to track a desired time-varying manipulability either as the main task of the robot or as a secondary objective. To address the former problem, we propose a tensor-based formulation of Gaussian mixture model (GMM) and Gaussian mixture regression (GMR) that takes into account that manipulability ellipsoids lie on the manifold of symmetric positive definite (SPD) matrices (see Section 3 for a full description of the model). The latter challenge is solved through a manipulability tracking formulation inspired by the classical inverse kinematics problem in robotics, where a first-order differential relationship between the robot manipulability ellipsoid and the robot joints is established, as explained in Section 4. This relationship also demands to consider that manipulability ellipsoids lie on the SPD manifold, which is here tackled by exploiting tensor-based representations and differential geometry (see Section 2). The geometry-awareness of our formulations is crucial for achieving successful manipulability transfer, as shown in Section 5. Note that Riemannian geometry has also been successfully exploited in robot motion optimization (Ratliff et al. 2015) and manipulability analysis of closed chains (Park and Kim 1998). For sake of clarity, different aspects of the proposed learning and tracking approaches are illustrated with simple examples using simulated planar robots throughout the article.

The proposed approach can be straightforwardly applied to different types of kinetostatic and dynamic manipulability measures. This opens the door to manipulability transfer scenarios with various types of robots where different task requirements at kinematic and dynamic levels can be learned and successfully transfered between agents of different embodiments. The functionality of the proposed approach is evaluated in different simulated manipulability tracking tasks involving a 16-DoF robotic hand and two legged robots. The full manipulability transfer is showcased in a bimanual setup where an unplugging task is kinesthetically demonstrated to a 14-DoF dual-arm robot, which then transfers the learned model to a different dual-arm system that reproduces the unplugging task successfully, as described in Section 6.

Early contributions on our learning and tracking frameworks were presented in (Rozo et al. 2017) and (Jaquier et al. 2018), respectively. In (Rozo et al. 2017), the learning approach provided a sequence of desired manipulability ellipsoids that a learner robot reproduced using gradient-based nullspace commands. Existing approaches built on the optimization of manipulability-based indices are not suitable as they do not allow the tracking of specific manipulability ellipsoids. In (Jaquier et al. 2018), the tracking framework

used manually-specified robot manipulability ellipsoids for the task. As mentioned previously, this may be tedious and cumbersome when the robot needs to carry out different and complex tasks. Therefore, the integration of the proposed learning and tracking approaches solves the aforementioned problems and offers a complete geometry-aware manipulability transfer framework where manipulability ellipsoid profiles are learned from demonstrations and reproduced accurately. This opens the possibility to transfer posture-dependent task requirements between agents of dissimilar kinematic structures. In particular, this framework also permits to transfer other velocity, force or impedance specifications with any priority order with respect to the manipulability tracking controller.

Beyond the combination of our early contributions on manipulability learning and tracking, the other contributions of this article are: *(i)* analyzing the role of the proposed differential geometry formulation of the geometry-aware tensor-based GMM/GMR adapted to manipulability ellipsoids; *(ii)* extending the geometry-aware manipulability tracking control scheme initially designed for kinetostatic manipulability measures to dynamic measures; *(iii)* demonstrating the exponential stability of the proposed manipulability tracking controllers; *(iv)* introducing various novel types of geometry-aware manipulability tracking schemes and introducing methodologies to consider the robot actuators contribution and variability-based tracking precision; *(v)* analyzing the importance of the geometry-awareness of the manipulability tracking controllers by comparison against state-of-the-art manipulability-based optimization methods.

A summary video, as well as videos of the illustrative planar examples and simulated and real experiments accompany the article and can be found at https://sites.google.com/view/manipulability. Related source codes are available at https://github.com/NoemieJaquier/Manipulability.

2 Background

2.1 Manipulability ellipsoids

Velocity and force manipulability ellipsoids introduced in (Yoshikawa 1985b) 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 $\dot{\boldsymbol{x}}$ and joint velocities $\dot{\boldsymbol{q}}$,

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

where $\boldsymbol{q} \in \mathbb{R}^n$ and $\boldsymbol{J} \in \mathbb{R}^{6\times n}$ are the joint position and Jacobian of the robot, respectively. Moreover, consider the set of joint velocities of constant (unit) norm $\|\dot{\boldsymbol{q}}\|^2 = 1$ 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*

$$\|\dot{\boldsymbol{q}}\|^2 = \dot{\boldsymbol{q}}^\mathsf{T}\dot{\boldsymbol{q}} = \dot{\boldsymbol{x}}^\mathsf{T}(\boldsymbol{J}\boldsymbol{J}^\mathsf{T})^{-1}\dot{\boldsymbol{x}}, \tag{2}$$

by using the least-squares inverse kinematics relation $\dot{\boldsymbol{q}} = \boldsymbol{J}^\dagger\dot{\boldsymbol{x}} = \boldsymbol{J}^\mathsf{T}(\boldsymbol{J}\boldsymbol{J}^\mathsf{T})^{-1}\dot{\boldsymbol{x}}$. Equation (2) represents the robot manipulability in terms of motion, indicating the flexibility of the manipulator in generating velocities in Cartesian space.†

In the literature, the velocity manipulability ellipsoid is usually defined as $(\boldsymbol{J}\boldsymbol{J}^\mathsf{T})^{-1}$, where the principal axes of the ellipsoid coincide with the eigenvectors and their length is equal to the inverse of the square root of the corresponding eigenvalues, i.e., $\frac{1}{\sqrt{\lambda_i}}$ (see e.g. (Chiu 1987)). For the sake of consistency, we here use an alternative definition of the velocity manipulability ellipsoid given by $\boldsymbol{M}^{\dot{\boldsymbol{x}}} = \boldsymbol{J}\boldsymbol{J}^\mathsf{T}$. In this case, the major axis of the manipulability ellipsoid is aligned to the eigenvector associated with the maximum eigenvalue $\lambda_{\max}$ of $\boldsymbol{M}^{\dot{\boldsymbol{x}}}$, whose length equals the square root of $\lambda_{\max}$. Thus, the interpretation and representation of the manipulability ellipsoid from the corresponding matrix are facilitated. Note that the major axis of the velocity manipulability ellipsoid $\boldsymbol{M}^{\dot{\boldsymbol{x}}} = \boldsymbol{J}\boldsymbol{J}^\mathsf{T}$ indicates the direction in which the greater velocity can be generated, which is also the direction in which the robot is more sensitive to perturbations. This occurs due to the principal axes of the force manipulability being aligned with those of the velocity manipulability, with reciprocal lengths (eigenvalues) caused by the duality of velocity and force (see (Chiu 1987) for details).

Other forms of manipulability ellipsoids exist, such as the dynamic manipulability (Yoshikawa 1985a), which gives a measure of the ability of performing end-effector accelerations along each task-space direction for a given set of joint torques. This has shown to be useful when the robot dynamics cannot be neglected in highly dynamic manipulation tasks (Chiacchio et al. 1991b). Recent works have extended this measure to analyze the robot capacity to accelerate its center of mass for locomotion stability (Azad et al. 2017; Gu et al. 2015), showing the applicability of the aforementioned tools beyond robotic manipulation.

As mentioned previously, any manipulability ellipsoid $\boldsymbol{M}$ belongs to the set of symmetric positive definite (SPD) matrices $\mathcal{S}^D_{++}$ which describe the interior of the convex cone. Consequently, our manipulability transfer formulation must consider this particular characteristic in order to properly encode, reproduce and track manipulability ellipsoids. To do so, we here propose geometry-aware formulations of both learning and tracking problems by exploiting Riemannian manifolds and tensor representations, which are introduced next.

2.2 Riemannian manifold of 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 (Pennec et al. 2006), 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

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

†Dually, the force manipulability ellipsoid can be computed from the static relationship between joint torques and Cartesian forces (Yoshikawa 1985b).

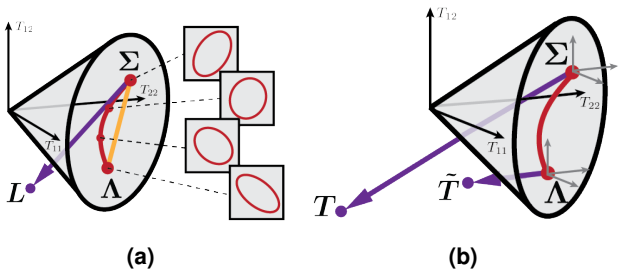


Figure 2. SPD manifold $\mathcal{S}^2_{++}$ embedded in its tangent space Sym^2. One point corresponds to a matrix $\left(\begin{smallmatrix} T_{11} & T_{12} \\ T_{12} & T_{22} \end{smallmatrix}\right) \in \text{Sym}^2$. Points inside the cone, such as $\boldsymbol{\Sigma}$ and $\boldsymbol{\Lambda}$, belong to $\mathcal{S}^2_{++}$. (*a*) $\boldsymbol{L}$ lies on the tangent space of $\boldsymbol{\Sigma}$ such that $\boldsymbol{L} = \text{Log}_{\boldsymbol{\Sigma}}(\boldsymbol{\Lambda})$. The shortest path between $\boldsymbol{\Sigma}$ and $\boldsymbol{\Lambda}$ is the geodesic represented as a red curve, which differs from the Euclidean path depicted in yellow. (*b*) $\tilde{\boldsymbol{T}} \in \mathcal{T}_{\boldsymbol{\Lambda}}\mathcal{M}$ is the result of the parallel transport of $\boldsymbol{T} \in \mathcal{T}_{\boldsymbol{\Sigma}}\mathcal{M}$ from $\mathcal{T}_{\boldsymbol{\Sigma}}\mathcal{M}$ to $\mathcal{T}_{\boldsymbol{\Lambda}}\mathcal{M}$. The coordinate axes of the tangent spaces $\mathcal{T}_{\boldsymbol{\Sigma}}\mathcal{M}$ and $\mathcal{T}_{\boldsymbol{\Lambda}}\mathcal{M}$ are represented in dark gray.

a Riemannian manifold.[‡] This metric permits to define geodesics, which 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.

Intuitively, a Riemannian manifold $\mathcal{M}$ is a mathematical space for which each point locally resembles a Euclidean space. For each point $\boldsymbol{\Sigma} \in \mathcal{M}$, there exists a tangent space $\mathcal{T}_{\boldsymbol{\Sigma}}\mathcal{M}$ equipped with a positive definite inner product. In the case of the SPD manifold, the tangent space at any point $\boldsymbol{\Sigma} \in \mathcal{S}^D_{++}$ is identified by the space of symmetric matrices Sym^D and the inner product between two matrices $\boldsymbol{T}_1, \boldsymbol{T}_2 \in \mathcal{T}_{\boldsymbol{\Sigma}}\mathcal{M}$ is

$$\langle \boldsymbol{T}_1, \boldsymbol{T}_2 \rangle_{\boldsymbol{\Sigma}} = \text{tr}(\boldsymbol{\Sigma}^{-\frac{1}{2}} \boldsymbol{T}_1 \boldsymbol{\Sigma}^{-1} \boldsymbol{T}_2 \boldsymbol{\Sigma}^{-\frac{1}{2}}). \quad (3)$$

The space of SPD matrices can be represented as the interior of a convex cone embedded in its tangent space Sym^D. To utilize these tangent spaces, we need mappings back and forth between $\mathcal{T}_{\boldsymbol{\Sigma}}\mathcal{M}$ and $\mathcal{M}$, which are known as exponential and logarithmic 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 norm of $\boldsymbol{L}$ in the tangent space. The inverse operation is called the logarithmic map $\text{Log}_{\boldsymbol{\Sigma}} : \mathcal{M} \to \mathcal{T}_{\boldsymbol{\Sigma}}\mathcal{M}$. Both operations are illustrated in Fig. 2a.

Specifically, the exponential and logarithmic maps on the SPD manifold corresponding to the affine-invariant distance

$$d(\boldsymbol{\Lambda}, \boldsymbol{\Sigma}) = \| \log(\boldsymbol{\Sigma}^{-\frac{1}{2}} \boldsymbol{\Lambda} \boldsymbol{\Sigma}^{-\frac{1}{2}}) \|_{\text{F}}, \quad (4)$$

are computed as (see (Pennec et al. 2006) for details)

$$\boldsymbol{\Lambda} = \text{Exp}_{\boldsymbol{\Sigma}}(\boldsymbol{L}) = \boldsymbol{\Sigma}^{\frac{1}{2}} \exp(\boldsymbol{\Sigma}^{-\frac{1}{2}} \boldsymbol{L} \boldsymbol{\Sigma}^{-\frac{1}{2}}) \boldsymbol{\Sigma}^{\frac{1}{2}}, \quad (5)$$

$$\boldsymbol{L} = \text{Log}_{\boldsymbol{\Sigma}}(\boldsymbol{\Lambda}) = \boldsymbol{\Sigma}^{\frac{1}{2}} \log(\boldsymbol{\Sigma}^{-\frac{1}{2}} \boldsymbol{\Lambda} \boldsymbol{\Sigma}^{-\frac{1}{2}}) \boldsymbol{\Sigma}^{\frac{1}{2}}, \quad (6)$$

where $\exp(\cdot)$ and $\log(\cdot)$ are the matrix exponential and logarithm functions.

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. 2b). The parallel transport of $\boldsymbol{T} \in \mathcal{T}_{\boldsymbol{\Sigma}}\mathcal{S}^D_{++}$ to $\mathcal{T}_{\boldsymbol{\Lambda}}\mathcal{S}^D_{++}$ is given by

$$\tilde{\boldsymbol{T}} = \Gamma_{\boldsymbol{\Sigma}\to\boldsymbol{\Lambda}}(\boldsymbol{T}) = \boldsymbol{A}_{\boldsymbol{\Sigma}\to\boldsymbol{\Lambda}}\, \boldsymbol{T}\, \boldsymbol{A}^{\text{T}}_{\boldsymbol{\Sigma}\to\boldsymbol{\Lambda}}, \quad (7)$$

with $\boldsymbol{A}_{\boldsymbol{\Sigma}\to\boldsymbol{\Lambda}} = \boldsymbol{\Lambda}^{\frac{1}{2}} \boldsymbol{\Sigma}^{-\frac{1}{2}}$ (see (Sra and Hosseini 2015) for details). This operation is exploited when it is necessary to move SPD matrices along a curve on the nonlinear manifold. Finally, for a complete introduction to differential geometry and Riemannian manifolds, we refer the interested reader to e.g., (do Carmo 1992; Lee 2012).

In this article, we first exploit the Riemannian manifold framework to propose a probabilistic learning model that encodes and retrieves manipulability ellipsoids considering that these belong to $\mathcal{S}^D_{++}$. Secondly, we take advantage of the Riemannian geometry to compute the difference between manipulability ellipsoids in the tracking problem, and consequently propose novel velocity- and acceleration-based controllers. This geometry-aware approach proves to be crucial for learning and tracking manipulability ellipsoids in terms of accuracy, stability and convergence, beyond providing an appropriate mathematical treatment of both problems.

2.3 Tensor representation

Tensors are generalization of matrices to arrays of higher dimensions (Kolda and Bader 2009), where vectors and matrices may respectively be seen as 1st and 2nd-order tensors. Tensor representation permits to represent and exploit data structure of multidimensional arrays. In this article, such representation is first used in the learning process to encode a distribution of manipulability ellipsoids (as explained in Section 3). Then, tensor representation is also exploited in the proposed manipulability tracking formulation to find the first-order differential relationship between the robot joints and the robot manipulability ellipsoid (1st- and 2nd-order tensors, respectively), which results in a 3rd-order tensor (see Section 4). To do so, we first introduce the tensor operations needed for our mathematical treatment.

2.3.1 Tensor product The tensor product is a multilinear generalization of the outer product of two vectors $\boldsymbol{x} \otimes \boldsymbol{y} = \boldsymbol{x}\boldsymbol{y}^{\text{T}}$. The tensor product of two tensors $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times \ldots \times I_M}$, $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{J_1 \times \ldots \times J_N}$ is $\boldsymbol{\mathcal{X}} \otimes \boldsymbol{\mathcal{Y}} \in \mathbb{R}^{I_1 \times \ldots \times I_M \times J_1 \times \ldots \times J_N}$ with elements

$$(\boldsymbol{\mathcal{X}} \otimes \boldsymbol{\mathcal{Y}})_{i_1,\ldots,i_M,j_1,\ldots,j_N} = x_{i_1,\ldots,i_M}\, y_{j_1,\ldots,j_N}. \quad (8)$$

2.3.2 n-mode product The multiplication of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times \ldots \times I_n \times \ldots \times I_N}$ by a matrix $\boldsymbol{A} \in \mathbb{R}^{J \times I_n}$, known as the n-mode product is defined as

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}} \times_n \boldsymbol{A} \iff \boldsymbol{Y}_{(n)} = \boldsymbol{A}\boldsymbol{X}_{(n)}, \quad (9)$$

where $\boldsymbol{X}_{(n)} \in \mathbb{R}^{I_n \times I_1 I_2 \ldots I_N}$ is the n-mode matricization or unfolding of tensor $\boldsymbol{\mathcal{X}}$. Element-wise, this n-mode product can be written as $(\boldsymbol{\mathcal{X}} \times_n \boldsymbol{A})_{i_1 \ldots i_{n-1} j_n i_{n+1} \ldots i_N} = \sum_{i_n} a_{j_n i_n} x_{i_1 \ldots i_{n-1} i_n i_{n+1} \ldots i_N}$.

[‡]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 (Pennec et al. 2006).

2.3.3 Tensor contraction As described in (Tyagi and Davis 2008), we denote the element (i, j, k, l) of a 4th-order tensor $\boldsymbol{\mathcal{S}}$ by $\boldsymbol{\mathcal{S}}_{ij}^{kl}$ with two covariant indices i, j and two contravariant indices k, l. The element (k,l) of a matrix $\boldsymbol{X}$ is denoted by $\boldsymbol{X}_{kl}$ with two covariant indices k, l. 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

$$\boldsymbol{\mathcal{S}}\boldsymbol{X} = \sum_{k=1}^{D}\sum_{l=1}^{D} \boldsymbol{\mathcal{S}}_{ij}^{kl}\boldsymbol{X}_{kl}. \tag{10}$$

2.3.4 Tensor covariance Similarly to the covariance of vectors, the $2M$th-order covariance tensor $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{I_1\times\ldots\times I_M\times I_1\times\ldots\times I_M}$ of centered tensors $\boldsymbol{\mathcal{X}}_n \in \mathbb{R}^{I_1\times\ldots\times I_M}$ is given by

$$\boldsymbol{\mathcal{S}} = \frac{1}{N-1}\sum_{n=1}^{N} \boldsymbol{\mathcal{X}}_n \otimes \boldsymbol{\mathcal{X}}_n, \tag{11}$$

where N is the total number of datapoints. This definition is used in the formulation of tensor-variate normal distributions.

2.3.5 Normal distribution of symmetric matrices The tensor-variate normal distribution of a random 2nd-order symmetric matrix $\boldsymbol{X} \in \mathrm{Sym}^D$ with mean $\boldsymbol{\Xi} \in \mathrm{Sym}^D$ and covariance $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{D\times D\times D\times D}$ is defined as (Basser and Pajevic 2007)

$$\mathcal{N}(\boldsymbol{X}|\boldsymbol{\Xi}, \boldsymbol{\mathcal{S}}) = \frac{1}{\sqrt{(2\pi)^{\tilde{D}}|\boldsymbol{\mathcal{S}}|}}\, e^{-\frac{1}{2}(\boldsymbol{X}-\boldsymbol{\Xi})\boldsymbol{\mathcal{S}}^{-1}(\boldsymbol{X}-\boldsymbol{\Xi})}, \tag{12}$$

with $\tilde{D} = D + D(D-1)/2$. This formulation is used in Section 3 to formulate a normal distribution of SPD matrices necessary to adapt the formulations of GMM and GMR to encode and retrieve manipulability ellipsoids.

2.3.6 Derivative of a matrix w.r.t a vector In the following identities, the matrix $\boldsymbol{Y}\in\mathbb{R}^{I\times J}$ is a function of $\boldsymbol{x}\in\mathbb{R}^{K}$, while $\boldsymbol{A}\in\mathbb{R}^{L\times I}$ and $\boldsymbol{B}\in\mathbb{R}^{J\times L}$ are constant matrices. The derivative of a matrix function $\boldsymbol{Y}$ with respect to a vector $\boldsymbol{x}$ is a 3rd-order tensor $\frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{x}} \in \mathbb{R}^{I\times J\times K}$ such that

$$\left(\frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{x}}\right)_{ijk} = \frac{\partial y_{ij}}{\partial x_k}. \tag{13}$$

Note that when the matrix function $\boldsymbol{Y}$ is multiplied by a constant matrix, the partial derivatives of $\boldsymbol{Y}$ are given by:

Left multiplication by a constant matrix

$$\frac{\partial \boldsymbol{A}\boldsymbol{Y}}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{x}} \times_1 \boldsymbol{A} \tag{14}$$

Right multiplication by a constant matrix

$$\frac{\partial \boldsymbol{Y}\boldsymbol{B}}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{x}} \times_2 \boldsymbol{B}^{\mathsf{T}} \tag{15}$$

Finally, another useful operation for our manipulability tracking formulation is the derivative of the inverse of the matrix $\boldsymbol{Y}$ with respect to the vector $\boldsymbol{x}$, which results in a 3rd-order tensor, namely

$$\frac{\partial \boldsymbol{Y}^{-1}}{\partial \boldsymbol{x}} = -\frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{x}}^{\mathsf{T}} \times_1 \boldsymbol{Y}^{-1} \times_2 \boldsymbol{Y}^{-\mathsf{T}} \tag{16}$$

Note that the proposed geometry-aware manipulability tracking, introduced in the section 4, takes inspiration from the computation of the robot Jacobian, which is computed from the 1st-order time derivative of the robot forward kinematics. We use the tensor representation to similarly compute the 1st-order derivative of the function that describes the relationship between a manipulability ellipsoid $\boldsymbol{M}$ and the robot joint configuration $\boldsymbol{q}$. Mathematical proofs for (14), (15) and (16) are given in Appendix A.

3 Learning Manipulability Ellipsoids

The first open problem in manipulability transfer is to appropriately encode sequences of demonstrated manipulability ellipsoids and subsequently retrieve a desired manipulability profile that encapsulates the patterns observed during the demonstrations. In order to describe how we tackle this problem, we first introduce the mathematical formulation of a Gaussian mixture model that encodes a set of demonstrated manipulability ellipsoids over the manifold of SPD matrices. This probabilistic formulation models the trend of the demonstrated manipulability sequences along with their variability, reflecting their dispersion through the different demonstrations. After, we describe how a distribution of the desired manipulability ellipsoids can be retrieved via Gaussian mixture regression on the SPD manifold.

3.1 Gaussian Mixture Model on SPD manifolds

Similarly to multivariate distribution (see (Zeestraten et al. 2017; Simo-Serra et al. 2017; Dubbelman 2011)), we can extend the normal distribution (12) to the SPD manifold. Thus, a tensor-variate distribution maximizing the entropy in the tangent space is approximated by

$$\mathcal{N}_{\mathcal{M}}(\boldsymbol{X}|\boldsymbol{\Xi}, \boldsymbol{\mathcal{S}}) = \frac{1}{\sqrt{(2\pi)^{\tilde{D}}|\boldsymbol{\mathcal{S}}|}}\, e^{-\frac{1}{2}\mathrm{Log}_{\boldsymbol{\Xi}}(\boldsymbol{X})\,\boldsymbol{\mathcal{S}}^{-1}\,\mathrm{Log}_{\boldsymbol{\Xi}}(\boldsymbol{X})}, \tag{17}$$

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

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

$$p(\boldsymbol{X}) = \sum_{k=1}^{K} \pi_k \mathcal{N}_{\mathcal{M}}(\boldsymbol{X}|\boldsymbol{\Xi}_k, \boldsymbol{\mathcal{S}}_k), \tag{18}$$

with K being the number of components of the model, and π_k representing the priors such that $\sum_k \pi_k = 1$.

The parameters of a GMM on the manifold of SPD matrices are estimated by Expectation-Maximization (EM) algorithm. Specifically, the responsibility of each component k is computed in the E-step as:

$$p(k|\boldsymbol{X}_i) = \frac{\pi_k\, \mathcal{N}_{\mathcal{M}}(\boldsymbol{X}_i|\boldsymbol{\Xi}_k, \boldsymbol{\mathcal{S}}_k)}{\sum_{j=1}^{K} \pi_j\, \mathcal{N}_{\mathcal{M}}(\boldsymbol{X}_i|\boldsymbol{\Xi}_j, \boldsymbol{\mathcal{S}}_j)}, \tag{19}$$

$$N_k = \sum_{i=1}^{N} p(k|\boldsymbol{X}_i). \tag{20}$$

During the M-step, the mean $\boldsymbol{\Xi}_k$ is first updated iteratively until convergence for each component. The covariance tensor

$\boldsymbol{\mathcal{S}}_k$ and prior π_k are then updated using the new mean:

$$\boldsymbol{\Xi}_k \leftarrow \frac{1}{N_k}\text{Exp}_{\boldsymbol{\Xi}_k}\left(\sum_{i=1}^{N} p(k|\boldsymbol{X}_i)\ \text{Log}_{\boldsymbol{\Xi}_k}(\boldsymbol{X}_i)\right), \tag{21}$$

$$\boldsymbol{\mathcal{S}}_k \leftarrow \frac{1}{N_k}\sum_{i=1}^{N} p(k|\boldsymbol{X}_i)\ \text{Log}_{\boldsymbol{\Xi}_k}(\boldsymbol{X}_i) \otimes \text{Log}_{\boldsymbol{\Xi}_k}(\boldsymbol{X}_i), \tag{22}$$

$$\pi_k \leftarrow \frac{N_k}{N}. \tag{23}$$

3.2 Gaussian Mixture Regression on SPD manifolds

GMR computes the conditional distribution $p(\boldsymbol{X}_{\mathcal{OO}}|\boldsymbol{X}_{\mathcal{II}})$ 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:

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{X}_{\mathcal{II}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{X}_{\mathcal{OO}} \end{pmatrix}, \boldsymbol{\Xi} = \begin{pmatrix} \boldsymbol{\Xi}_{\mathcal{II}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Xi}_{\mathcal{OO}} \end{pmatrix},$$
$$\boldsymbol{\mathcal{S}} = \left(\begin{array}{cc|cc} \boldsymbol{\mathcal{S}}^{\mathcal{II}}_{\mathcal{II}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\mathcal{S}}^{\mathcal{OO}}_{\mathcal{II}} & \boldsymbol{0} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\mathcal{S}}^{\mathcal{II}}_{\mathcal{OO}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\mathcal{S}}^{\mathcal{OO}}_{\mathcal{OO}} \end{array}\right), \tag{24}$$

where we represent the 4th-order tensor by separating the components of the 3rd- and 4th-modes with horizontal and vertical bars, respectively. With this decomposition, manifold functions can be applied individually on input and output parts, for example the exponential map would be

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

Similarly to GMR in Euclidean space (Rozo et al. 2016) and in manifolds where data are represented by vectors (Zeestraten et al. 2017), GMR on SPD manifold approximates the conditional distribution by a single Gaussian

$$p(\boldsymbol{X}_{\mathcal{OO}}|\boldsymbol{X}_{\mathcal{II}}) \sim \mathcal{N}(\hat{\boldsymbol{\Xi}}_{\mathcal{OO}}, \hat{\boldsymbol{\mathcal{S}}}^{\mathcal{OO}}_{\mathcal{OO}}), \tag{25}$$

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

$$\boldsymbol{\Delta}_k = \text{Log}_{\hat{\boldsymbol{\Xi}}_{\mathcal{OO}}}(\boldsymbol{\Xi}_{\mathcal{OO},k}) - \tilde{\boldsymbol{\mathcal{S}}}^{\mathcal{II}}_{\mathcal{OO},k}\ \tilde{\boldsymbol{\mathcal{S}}}^{\mathcal{II}-1}_{\mathcal{II},k}\ \text{Log}_{\boldsymbol{X}_{\mathcal{II}}}(\boldsymbol{\Xi}_{\mathcal{II},k}), \tag{26}$$

$$\hat{\boldsymbol{\Xi}}_{\mathcal{OO}} \leftarrow \text{Exp}_{\hat{\boldsymbol{\Xi}}_{\mathcal{OO}}}\Big(\sum_k h_k \boldsymbol{\Delta}_k\Big), \tag{27}$$

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

$$h_k = \frac{\pi_k\, \mathcal{N}(\boldsymbol{X}_{\mathcal{II}}|\boldsymbol{\Xi}_{\mathcal{II},k}, \boldsymbol{\mathcal{S}}^{\mathcal{II}}_{\mathcal{II},k})}{\sum_{j=1}^{K} \pi_j\, \mathcal{N}(\boldsymbol{X}_{\mathcal{II}}|\boldsymbol{\Xi}_{\mathcal{II},j}, \boldsymbol{\mathcal{S}}^{\mathcal{II}}_{\mathcal{II},j})}. \tag{28}$$

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

$$\hat{\boldsymbol{\mathcal{S}}}^{\mathcal{OO}}_{\mathcal{OO}} = \sum_k h_k \left(\tilde{\boldsymbol{\mathcal{S}}}^{\mathcal{OO}}_{\mathcal{OO},k} - \tilde{\boldsymbol{\mathcal{S}}}^{\mathcal{II}}_{\mathcal{OO},k}\tilde{\boldsymbol{\mathcal{S}}}^{\mathcal{II}-1}_{\mathcal{II},k}\tilde{\boldsymbol{\mathcal{S}}}^{\mathcal{OO}}_{\mathcal{II},k} + \boldsymbol{\Delta}_k \otimes \boldsymbol{\Delta}_k\right) - \hat{\boldsymbol{\Xi}}_{\mathcal{OO}} \otimes \hat{\boldsymbol{\Xi}}_{\mathcal{OO}}, \tag{29}$$

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

$$\tilde{\boldsymbol{\mathcal{S}}} = \Gamma_{\boldsymbol{\Xi}\to\hat{\boldsymbol{X}}}(\boldsymbol{\mathcal{S}}) \quad \text{with} \quad \hat{\boldsymbol{X}} = \begin{pmatrix} \boldsymbol{X}_{\mathcal{II}} & \boldsymbol{0} \\ \boldsymbol{0} & \hat{\boldsymbol{\Xi}}_{\mathcal{OO}} \end{pmatrix}. \tag{30}$$

This covariance has been typically used to define the controller gains of robotic systems for trajectory tracking problems (see also Section 4.4). Note that the definition of the tangent space $\mathcal{T}_{\boldsymbol{\Xi}}\mathcal{M}$ (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 (Basser and Pajevic 2007). We can then compute its eigentensors $\boldsymbol{V}_k$, which are used to parallel transport the covariance matrix between tangent spaces (Freifeld et al. 2014). Let $\tilde{\boldsymbol{V}}_k = \Gamma_{\boldsymbol{\Xi}\to\hat{\boldsymbol{X}}}(\boldsymbol{V}_k)$ be the k-th parallel transported eigentensor with (7) and λ_k the k-th eigenvalue. The parallel transported 4th-order covariance tensor is then obtained with (see (Jaquier and Calinon 2017) for more details)

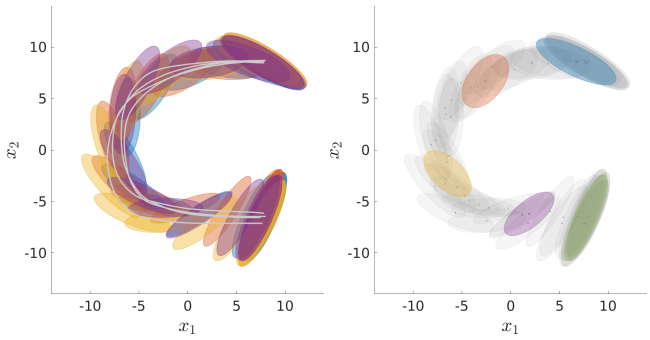
$$\Gamma_{\boldsymbol{\Xi}\to\hat{\boldsymbol{X}}}(\boldsymbol{\mathcal{S}}) = \sum_k \lambda_k \tilde{\boldsymbol{V}}_k \otimes \tilde{\boldsymbol{V}}_k. \tag{31}$$

3.3 Manipulability Learning Example with 2 Planar Robots

In order to illustrate the functionality of the proposed learning approach, we carried out an experiment using a couple of simulated planar robots with dissimilar embodiments and a different number of joints. The central idea is to teach a redundant robot to track a reference trajectory in Cartesian space with a desired time-varying manipulability ellipsoid. For the demonstration phase, a 3-DoF *teacher* robot follows a C-shape trajectory four times, from which we extracted both the end-effector position $\boldsymbol{x}_t$ and robot manipulability ellipsoid $\boldsymbol{M}_t(\boldsymbol{q})$, at each time step t. The collected time-aligned data were split into two training datasets of time-driven trajectories, namely Cartesian position and manipulability. We trained a classical GMM over the time-driven Cartesian trajectories and a geometry-aware GMM over the time-driven manipulability ellipsoids, using models with five components, i.e. $K=5$ (the number was selected by the experimenter).

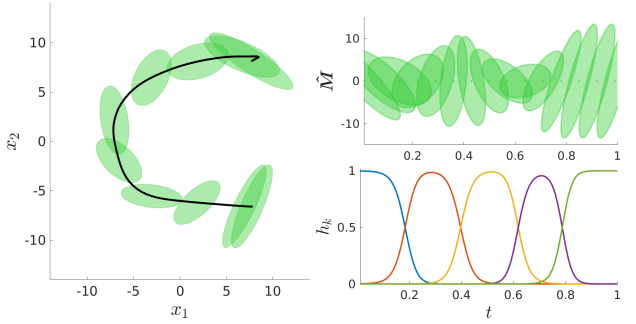
During the reproduction phase, a 5-DoF *student* robot executed the time-driven task by following a desired Cartesian trajectory $\hat{\boldsymbol{x}}_t$ computed from a classical GMR as $\hat{\boldsymbol{x}}_t \sim \mathcal{P}(\boldsymbol{x}\,|\,t)$. As secondary task, the robot was also required to vary its joint configuration for matching desired manipulability ellipsoids $\hat{\boldsymbol{M}}_t \sim \mathcal{P}(\boldsymbol{M}|t)$, estimated by GMR over the SPD manifold.

Figure 3a 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 3b displays the demonstrated ellipsoids (in gray) along with the center $\boldsymbol{\Xi}_k$ of the five components of the GMM encoding $\boldsymbol{X}^{\boldsymbol{M}}$. These are centered at the end-effector position recovered by the classical GMR for



(a) Demonstrations **(b)** Learned model

Figure 3. (*a*) Four demonstrations of a 3-DoF planar robot tracking a C-shape trajectory. The end-effector path (light gray solid lines) and the manipulability ellipsoids at different time steps are shown for all the demonstrations. (*b*) Demonstrated manipulability ellipsoids (in gray) and centers Ξ_k of the 5-states GMM in the SPD manifold. Position x and time t are given in centimeters and seconds, respectively.



(a) Desired execution **(b)** Time profile

Figure 4. (*a*) Desired execution of a C-shape tracking task. The desired Cartesian trajectory and manipulability profile are depicted as a black curve and green ellipsoids. (*b*)-*top* Desired manipulability ellipsoids estimated by GMR. (*b*)-*bottom* Influence of GMM components on the time-driven GMR. The colors match the distributions shown in Fig. 3b.

the corresponding time steps represented in the geometry-aware GMM. Figure 4 shows the desired Cartesian trajectory and manipulability ellipsoid profile respectively estimated by classical GMR and GMR in the SPD manifold. Both manipulability and Cartesian path are references to be tracked by the student robot.

These results validate that the proposed learning framework permits to learn and plan the reproduction of reference trajectories, while fulfilling additional task requirements encapsulated in a profile of desired manipulability ellipsoids. In Section 4, we develop a manipulability tracking formulation that will then be used by the 5-DoF student robot to track the desired manipulability profile obtained in the learning phase.

4 Tracking Manipulability Ellipsoids

Several robotic manipulation tasks may demand the robot to track a desired trajectory with certain velocity specifications, or apply forces along different task-related axes. These requirements are more easily achieved if the robot adopts a posture that suits velocity or force control commands. In other tasks, the robot may be required to adopt a posture that comply several aligned velocity or force requirements. These problems can be viewed as matching a set of desired manipulability ellipsoids that are compatible with the task requirements. In this section, we introduce an approach that addresses this problem by exploiting the mathematical concepts presented in Section 2.

4.1 Manipulability Jacobian

Given a desired profile of manipulability ellipsoids, the goal of the robot is to adapt its posture to match the desired manipulability, either as its main task or as a secondary objective. We here propose a formulation inspired by the classical inverse kinematics problem in robotics, which permits to compute the joint angle commands to track a desired manipulability ellipsoid.

First, the manipulability ellipsoid is expressed as a function of time

$$\boldsymbol{M}(t) = f\Big(\boldsymbol{J}\big(\boldsymbol{q}(t)\big)\Big), \tag{32}$$

for which we can compute the first-order time derivative by applying the chain rule as

$$\frac{\partial \boldsymbol{M}(t)}{\partial t} = \frac{\partial f(\boldsymbol{J}(\boldsymbol{q}))}{\partial \boldsymbol{q}} \times_3 \frac{\partial \boldsymbol{q}(t)}{\partial t}^{\mathsf{T}} = \boldsymbol{\mathcal{J}}(\boldsymbol{q}) \times_3 \dot{\boldsymbol{q}}^{\mathsf{T}}, \tag{33}$$

where $\boldsymbol{\mathcal{J}} \in \mathbb{R}^{6\times 6\times n}$ is the *manipulability Jacobian* of an n-DoF robot, representing the linear sensitivity of the changes in the robot manipulability ellipsoid $\dot{\boldsymbol{M}} = \frac{\partial \boldsymbol{M}(t)}{\partial t}$ to the joint velocity $\dot{\boldsymbol{q}} = \frac{\partial \boldsymbol{q}(t)}{\partial t}$. Note that the computation of the manipulability Jacobian depends on the type of manipulability ellipsoid that is used. We develop here the expressions for the force, velocity and dynamic manipulability ellipsoids.

The derivation of the manipulability Jacobian $\boldsymbol{\mathcal{J}}^{\dot{\boldsymbol{x}}}$ corresponding to the velocity manipulability ellipsoid $\boldsymbol{M}^{\dot{\boldsymbol{x}}} = \boldsymbol{J}\boldsymbol{J}^{\mathsf{T}}$ is straightforward by using (14) and (15) [§]

$$\boldsymbol{\mathcal{J}}^{\dot{\boldsymbol{x}}} = \frac{\partial \boldsymbol{J}}{\partial \boldsymbol{q}} \times_2 \boldsymbol{J} + \frac{\partial \boldsymbol{J}^{\mathsf{T}}}{\partial \boldsymbol{q}} \times_1 \boldsymbol{J}. \tag{34}$$

Similarly, the manipulability Jacobian $\boldsymbol{\mathcal{J}}^{\boldsymbol{F}}$ corresponding to the force manipulability ellipsoid $\boldsymbol{M}^{\boldsymbol{F}} = (\boldsymbol{J}\boldsymbol{J}^{\mathsf{T}})^{-1}$ is obtained using (14), (15) and (16),

$$\boldsymbol{\mathcal{J}}^{\boldsymbol{F}} = -\left(\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{q}} \times_2 \boldsymbol{J} + \frac{\partial \boldsymbol{J}^{\mathsf{T}}}{\partial \boldsymbol{q}} \times_1 \boldsymbol{J}\right) \times_1 \boldsymbol{M}^{\dot{\boldsymbol{x}}} \times_2 \boldsymbol{M}^{\dot{\boldsymbol{x}}}. \tag{35}$$

In a similar fashion, the manipulability Jacobian $\boldsymbol{\mathcal{J}}^{\ddot{\boldsymbol{x}}}$ corresponding to the dynamic manipulability ellipsoid $\boldsymbol{M}^{\ddot{\boldsymbol{x}}} = \boldsymbol{\Upsilon}\boldsymbol{\Upsilon}^{\mathsf{T}}$ with $\boldsymbol{\Upsilon} = \boldsymbol{J}\boldsymbol{\Lambda}(\boldsymbol{q})^{-1}$ (as defined in (Yoshikawa 1985a), where $\boldsymbol{\Lambda}(\boldsymbol{q})$ is the robot inertia matrix), is computed as follows

$$\boldsymbol{\mathcal{J}}^{\ddot{\boldsymbol{x}}} = \frac{\partial \boldsymbol{\Upsilon}}{\partial \boldsymbol{q}} \times_2 \boldsymbol{\Upsilon} + \frac{\partial \boldsymbol{\Upsilon}^{\mathsf{T}}}{\partial \boldsymbol{q}} \times_1 \boldsymbol{\Upsilon}, \tag{36}$$

[§] In the remainder of the article we drop dependencies on $\boldsymbol{q}$ to simplify the notation.

where

$$\frac{\partial \boldsymbol{\Upsilon}}{\partial \boldsymbol{q}} = \frac{\partial \boldsymbol{J}}{\partial \boldsymbol{q}} \times_2 \boldsymbol{\Lambda}^{-\mathsf{T}} + \frac{\partial \boldsymbol{\Lambda}^{-1}}{\partial \boldsymbol{q}} \times_1 \boldsymbol{J} = \frac{\partial \boldsymbol{J}}{\partial \boldsymbol{q}} \times_2 \boldsymbol{\Lambda}^{-\mathsf{T}} - \frac{\partial \boldsymbol{\Lambda}}{\partial \boldsymbol{q}} \times_1 \boldsymbol{\Upsilon} \times_2 \boldsymbol{\Lambda}^{-\mathsf{T}}.$$

Details on the computation of the derivative of the Jacobian and inertia matrix w.r.t the joint angles are given in Appendices B and C.

4.2 Geometry-aware manipulability tracking formulation

4.2.1 Velocity-based controller A solution to control a robot so that it tracks a desired end-effector trajectory is to compute the desired joint velocities using the inverse kinematics formulation derived from (1). We use here a similar approach to compute the joint velocities $\dot{\boldsymbol{q}}$ to track a desired manipulability profile. More specifically, by minimizing the ℓ^2 norm of the residuals

$$\min_{\dot{\boldsymbol{q}}} \|\dot{\boldsymbol{M}} - \boldsymbol{\mathcal{J}} \times_3 \dot{\boldsymbol{q}}^{\mathsf{T}}\| = \min_{\dot{\boldsymbol{q}}} \|\mathrm{vec}(\dot{\boldsymbol{M}}) - \boldsymbol{\mathcal{J}}_{(3)}^{\mathsf{T}} \dot{\boldsymbol{q}}\|,$$

we can compute the required joint velocities of the robot to track a profile of desired manipulability ellipsoids as its main task with

$$\dot{\boldsymbol{q}} = (\boldsymbol{\mathcal{J}}_{(3)}^{\dagger})^{\mathsf{T}} \mathrm{vec}(\dot{\boldsymbol{M}}), \tag{37}$$

where $\mathrm{vec}(\dot{\boldsymbol{M}})$ is the vectorization of the matrix $\dot{\boldsymbol{M}}$.

Note that (37) allows us to define a controller to track a reference manipulability ellipsoid as main task, similarly as the classical velocity-based control that tracks a desired task-space velocity. To do so, we propose to use a geometry-aware similarity measure to compute the joint velocities necessary to move the robot towards a posture where the match between the current manipulability ellipsoid $\boldsymbol{M}_t$ and the desired one $\hat{\boldsymbol{M}}_t$ is maximum. Specifically, the difference between manipulability ellipsoids is computed using the logarithmic map (6) on the SPD manifold. Therefore, the corresponding controller is given by

$$\dot{\boldsymbol{q}}_t = (\boldsymbol{\mathcal{J}}_{(3)}^{\dagger})^{\mathsf{T}} \boldsymbol{K}_{\boldsymbol{M}} \mathrm{vec}\Big(\mathrm{Log}_{\boldsymbol{M}_t}(\hat{\boldsymbol{M}}_t)\Big), \tag{38}$$

where $\boldsymbol{K}_{\boldsymbol{M}}$ is a gain matrix.

Alternatively, for the case in which the main task of the robot is to track reference trajectories in the form of Cartesian positions or force profiles, the tracking of a profile of manipulability ellipsoids is assigned a secondary role. Thus, the robot task objectives are to track the reference trajectories while exploiting the kinematic redundancy to minimize the difference between current and desired manipulability ellipsoids. In this situation, a manipulability-based redundancy resolution is carried out by computing a nullspace velocity that similarly exploits the geometry of the SPD manifold. Thus, the corresponding controller is given by

$$\dot{\boldsymbol{q}}_t = \boldsymbol{J}^{\dagger} \boldsymbol{K}_{\boldsymbol{x}} (\hat{\boldsymbol{x}}_t - \boldsymbol{x}_t) + (\boldsymbol{I} - \boldsymbol{J}^{\dagger}\boldsymbol{J}) (\boldsymbol{\mathcal{J}}_{(3)}^{\dagger})^{\mathsf{T}} \boldsymbol{K}_{\boldsymbol{M}} \mathrm{vec}\Big(\mathrm{Log}_{\boldsymbol{M}_t}(\hat{\boldsymbol{M}}_t)\Big). \tag{39}$$

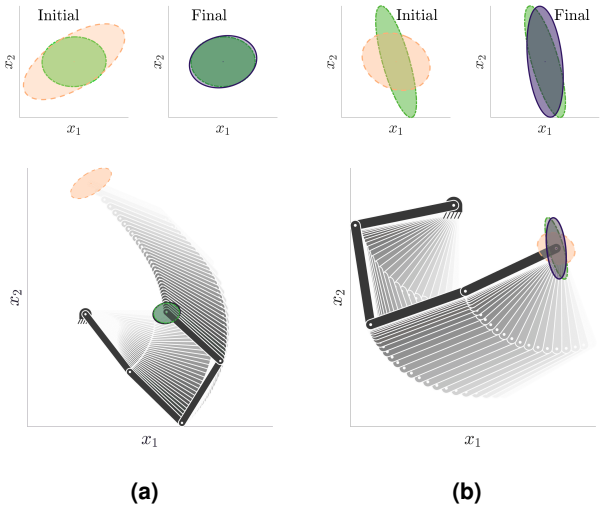


Figure 5. (*a*) Manipulability tracking as main task. (*b*) Manipulability-based redundancy resolution with Cartesian position control. The robot color goes from light gray to black to show the evolution of the posture. Initial, final, and desired manipulability ellipsoids are respectively depicted in yellow, dark purple, and green. The *top* rows show close-up plots corresponding to the initial and final manipulability.

Table 1. Initial and final distances $d(\hat{\boldsymbol{M}}, \boldsymbol{M}_t)$ between the current and desired manipulability for the experiments illustrated in Fig. 5.

	Initial	Final
Main task	1.342	0.199
Redundancy resolution	2.194	0.955

Note that matricization and vectorization operations can be defined using Mandel notation to alleviate the computational cost of the controllers using tensor representations, such that

$$\mathcal{X}_{(3)} = \begin{pmatrix} \mathrm{vec}\,(\boldsymbol{\mathcal{X}}_{:,:,1})^{\mathsf{T}} \\ \vdots \\ \mathrm{vec}\,(\boldsymbol{\mathcal{X}}_{:,:,K})^{\mathsf{T}} \end{pmatrix} \text{ and } \mathrm{vec}\left(\begin{pmatrix} \alpha & \beta \\ \beta & \gamma \end{pmatrix}\right) = \begin{pmatrix} \alpha \\ \gamma \\ \sqrt{2}\beta \end{pmatrix}, \tag{40}$$

for $2\times2\times K$ third-order tensors and 2×2 matrices.

In order to show the functionality of the proposed approach where the goal of the robot is to reproduce a given manipulability ellipsoid either as its main task or as a secondary objective, we carried out experiments with a simulated 4-DoF planar robot. In the first case, the robot is required to vary its joint configuration to make its manipulability ellipsoid $\boldsymbol{M}_t$ coincide with the desired one $\hat{\boldsymbol{M}}$, without any task requirement at the level of its end-effector. In the second case, the robot needs to keep its end-effector at a fixed Cartesian position while moving its joints to match the desired manipulability ellipsoid. Fig. 5 shows how the manipulator configuration is successfully adjusted so that $\boldsymbol{M}_t \simeq \hat{\boldsymbol{M}}$ when the manipulability ellipsoid tracking is considered as the main task or as a secondary objective (see Table 1). These results show that our geometry-aware controllers inspired by the inverse kinematics formulation

are suitable to solve the manipulability ellipsoid tracking problem.

Stability analysis We here analyze the stability properties of the proposed manipulability tracking controller given the geometry of the underlying manifold. First of all, note that the dynamical system operated by the controller (38) corresponds to

$$\dot{\boldsymbol{M}} = k_{\boldsymbol{M}} \operatorname{Log}_{\boldsymbol{M}}(\hat{\boldsymbol{M}}), \tag{41}$$

where the controller gain is assumed to be a positive scalar value for sake of simplicity. Then, we select the Lyapunov function V as

$$V(\boldsymbol{M}) \;=\; \langle \boldsymbol{F}, \boldsymbol{F} \rangle_{\hat{\boldsymbol{M}}}, \tag{42}$$

where $\boldsymbol{F} = \operatorname{Log}_{\hat{\boldsymbol{M}}}(\boldsymbol{M})$ is a vector field composed of the initial velocities of all geodesics departing from the origin $\hat{\boldsymbol{M}}$, and $\langle \cdot, \cdot \rangle_{\hat{\boldsymbol{M}}}$ is the inner product (3). As proved in (Pait and Colón 2010), the function (42) is a Lyapunov function for a dynamical system $\dot{\boldsymbol{M}} = h(\boldsymbol{M})$ such that $h(\hat{\boldsymbol{M}}) = 0$ if the Lie derivative $\mathcal{L}_h V(\boldsymbol{M}) = 2\langle h, \boldsymbol{F} \rangle_{\hat{\boldsymbol{M}}}$ is negative everywhere except at the origin $\hat{\boldsymbol{M}}$. To verify this condition, we first express the velocity of the dynamical system (41) in the tangent space of $\hat{\boldsymbol{M}}$ using parallel transport as

$$\Gamma_{\boldsymbol{M} \to \hat{\boldsymbol{M}}}(\dot{\boldsymbol{M}}) = -k_{\hat{\boldsymbol{M}}} \operatorname{Log}_{\hat{\boldsymbol{M}}}(\boldsymbol{M}). \tag{43}$$

The Lie derivative $\mathcal{L}_h V$ of the proposed Lyapunov function for the dynamical system (43) is given by

$$\begin{aligned} \mathcal{L}_h V(\boldsymbol{M}) \;&=\; 2\langle -k_{\hat{\boldsymbol{M}}} \operatorname{Log}_{\hat{\boldsymbol{M}}}(\boldsymbol{M}), \operatorname{Log}_{\hat{\boldsymbol{M}}}(\boldsymbol{M}) \rangle_{\hat{\boldsymbol{M}}} \\ &= -2k_{\hat{\boldsymbol{M}}} \langle \operatorname{Log}_{\hat{\boldsymbol{M}}}(\boldsymbol{M}), \operatorname{Log}_{\hat{\boldsymbol{M}}}(\boldsymbol{M}) \rangle_{\hat{\boldsymbol{M}}} \\ &= -2k_{\hat{\boldsymbol{M}}} V. \end{aligned} \tag{44}$$

Therefore, we have

$$\begin{aligned} &V(\boldsymbol{M}) > 0, \quad \mathcal{L}_h V(\boldsymbol{M}) < 0 \qquad \forall \;\; \boldsymbol{M} \neq \hat{\boldsymbol{M}}, \\ &V(\boldsymbol{M}) = \mathcal{L}_h V(\boldsymbol{M}) = 0 \iff \boldsymbol{M} = \hat{\boldsymbol{M}}, \end{aligned}$$

so that the function (42) is a valid Lyapunov function. Moreover, by observing that $V(\boldsymbol{M}) = d^2(\boldsymbol{M}, \hat{\boldsymbol{M}})$ with $d(\cdot, \cdot)$ the affine-invariant distance (4), we have

$$\begin{aligned} &c_1 d^2(\boldsymbol{M}, \hat{\boldsymbol{M}}) \leq V(\boldsymbol{M}) \leq c_2 d^2(\boldsymbol{M}, \hat{\boldsymbol{M}}), \\ &\mathcal{L}_h V(\boldsymbol{M}) \leq -c_3 d^2(\boldsymbol{M}, \hat{\boldsymbol{M}}), \end{aligned}$$

where $0 < c_1 \leq 1$, $c_2 \geq 1$, $c_3 = 2k_{\hat{\boldsymbol{M}}} > 0$. This implies that the controller (38) is exponentially stable (see e.g., (Wu 2020)). It can be easily shown that this result holds with $c_3 = 2\lambda_{\min}(\boldsymbol{K}_{\hat{\boldsymbol{M}}})$ for a positive-definite controller gain matrix $\boldsymbol{K}_{\hat{\boldsymbol{M}}}$, where $\lambda_{\min}(\cdot)$ returns the minimum eigenvalue of the matrix.

Note that the Lyapunov function (42) is similar to the one usually defined to demonstrate the exponential stability of the classical inverse kinematic-based velocity controller $\dot{\boldsymbol{q}}_t = \boldsymbol{J}^{\dagger} \boldsymbol{K}_{\boldsymbol{x}} (\hat{\boldsymbol{x}}_t - \boldsymbol{x}_t)$. In that case, the Lyapunov function is defined as $V(\boldsymbol{x}) = (\hat{\boldsymbol{x}} - \boldsymbol{x})^{\mathsf{T}} (\hat{\boldsymbol{x}} - \boldsymbol{x})$, which is equivalent to the inner product $\langle \boldsymbol{e}, \boldsymbol{e} \rangle$ with the error $\boldsymbol{e} = \hat{\boldsymbol{x}} - \boldsymbol{x}$. In the case of manipulability tracking, the inner product $\langle \cdot, \cdot \rangle$ is defined in the SPD manifold and the error $\boldsymbol{e}$ is computed as $\operatorname{Log}_{\hat{\boldsymbol{M}}}(\boldsymbol{M})$. Finally, it is worth highlighting that when the manipulability tracking is assigned a secondary role, the controller (39) does not influence the stability of the main task of the robot as the manipulability-based redundancy resolution is carried out in the corresponding nullspace.

4.2.2 Acceleration-based controller Similarly to the velocity-based controller, we propose a geometry-aware acceleration-based controller that allows the computation of the joint accelerations $\ddot{\boldsymbol{q}}$ required to track a desired manipulability trajectory (i.e. desired manipulability and manipulability velocity profiles). The approach is inspired by the inverse kinematics formulation and its differential relationships used to compute the joint accelerations necessary to track desired end-effector positions and velocities.

To formalize the acceleration-based controller, let us first define the second-order time derivative of the manipulability ellipsoid computed from (33) by applying the product rule

$$\frac{\partial^2 \boldsymbol{M}(t)}{\partial t^2} = \boldsymbol{\mathcal{J}}(\boldsymbol{q}) \times_3 \ddot{\boldsymbol{q}}^{\mathsf{T}} + \dot{\boldsymbol{\mathcal{J}}}(\boldsymbol{q}) \times_3 \dot{\boldsymbol{q}}^{\mathsf{T}}, \tag{45}$$

(see Appendix D for details on the computation of $\dot{\boldsymbol{\mathcal{J}}}(\boldsymbol{q})$). So, by minimizing the ℓ^2-norm of the residuals, we can compute the required joint accelerations of the robot to track a desired trajectory of manipulability ellipsoids as its main task with

$$\ddot{\boldsymbol{q}} = (\boldsymbol{\mathcal{J}}^{\dagger}_{(3)})^{\mathsf{T}} \Big(\operatorname{vec}(\ddot{\boldsymbol{M}}) - \dot{\boldsymbol{\mathcal{J}}}^{\mathsf{T}}_{(3)} \dot{\boldsymbol{q}} \Big). \tag{46}$$

Similarly as in the classical acceleration-based controller that tracks a desired end-effector trajectory, we can define a controller to track a reference manipulability ellipsoid trajectory based on (46). To do so, we exploit the geometry of the SPD manifold to compute the difference between the current manipulability ellipsoid $\boldsymbol{M}_t$ and the desired one $\hat{\boldsymbol{M}}_t$, as previously specified for the velocity-based controller. Moreover, since the first-order time derivative of manipulability ellipsoids lies on the tangent space of the SPD manifold (i.e. the space of symmetric matrices Sym^D), the difference between the current manipulability velocity $\dot{\boldsymbol{M}}_t$ and the desired one $\widehat{\dot{\boldsymbol{M}}}_t$ is computed as a subtraction in the Euclidean space. Therefore, a reference manipulability acceleration command can be specified by

$$\operatorname{vec}(\ddot{\boldsymbol{M}}_t) = \boldsymbol{K}_p \operatorname{vec}\Big(\operatorname{Log}_{\boldsymbol{M}_t}(\hat{\boldsymbol{M}}_t) \Big) + \boldsymbol{K}_d \operatorname{vec}\big(\widehat{\dot{\boldsymbol{M}}}_t - \dot{\boldsymbol{M}}_t \big), \tag{47}$$

which resembles a proportional-derivative controller where $\boldsymbol{K}_p$ and $\boldsymbol{K}_d$ are gain matrices. Then, the reference joint acceleration $\ddot{\boldsymbol{q}}$ can be computed using (46) and (47). Note that this reference joint acceleration can correspond to a main task of the robot or to a secondary tracking objective. In the latter case, a manipulability-based redundancy resolution can also be implemented in a similar way as (39).

4.3 Actuators contribution

In many practical applications, the joint velocities of the robot are limited. The definition of manipulability ellipsoid can then be extended to include these actuation constraints, as shown in (Lee 1997). We here provide the definition of the force, velocity and dynamic manipulability ellipsoids and the corresponding manipulability Jacobians considering joint actuation constraints.

To include the joint velocity constraints of the robot in the definition of the velocity manipulability ellipsoid, we use the

following weighted forward kinematics formulation

$$\dot{\boldsymbol{x}} = \underbrace{(\boldsymbol{J}\boldsymbol{W}^{\dot{\boldsymbol{q}}})}_{\tilde{\boldsymbol{J}}} \underbrace{(\boldsymbol{W}^{\dot{\boldsymbol{q}}-1}\dot{\boldsymbol{q}})}_{\tilde{\dot{\boldsymbol{q}}}}, \tag{48}$$

where $\boldsymbol{W}^{\dot{\boldsymbol{q}}} = \text{diag}(\dot{q}_{1,\max}, \dots, \dot{q}_{n,\max})$ is a diagonal matrix whose elements correspond to the maximum joint velocities of the robot. Then, considering the set of joint velocities of constant unit norm $\|\tilde{\dot{\boldsymbol{q}}}\| = 1$ mapped into the Cartesian velocity space through

$$\|\tilde{\dot{\boldsymbol{q}}}\|^2 = \tilde{\dot{\boldsymbol{q}}}^{\mathsf{T}}\tilde{\dot{\boldsymbol{q}}} = \dot{\boldsymbol{x}}^{\mathsf{T}}(\tilde{\boldsymbol{J}}\tilde{\boldsymbol{J}}^{\mathsf{T}})^{-1}\dot{\boldsymbol{x}}, \tag{49}$$

the velocity manipulability ellipsoid is given by $\tilde{\boldsymbol{M}}^{\dot{\boldsymbol{x}}} = \tilde{\boldsymbol{J}}\tilde{\boldsymbol{J}}^{\mathsf{T}} = \boldsymbol{J}\boldsymbol{W}^{\dot{\boldsymbol{q}}}\boldsymbol{W}^{\dot{\boldsymbol{q}}\mathsf{T}}\boldsymbol{J}^{\mathsf{T}}$, which represents the flexibility of the manipulator in generating velocities in Cartesian space considering its maximum joint velocities as illustrated in Figure 6a. Note that the actuators contribution $\boldsymbol{W}^{\dot{\boldsymbol{q}}}\boldsymbol{W}^{\dot{\boldsymbol{q}}\mathsf{T}}$ also has a geometrical interpretation based on the fact that the robot joint position $\boldsymbol{q}$ lies on the flat n-torus manifold (Park 1995).

By following the methodology of Section 4.1, the change in the robot manipulability ellipsoid is related to the joint velocity via

$$\frac{\partial \tilde{\boldsymbol{M}}(t)}{\partial t} = \tilde{\boldsymbol{\mathcal{J}}}(\boldsymbol{q}) \times_3 \dot{\boldsymbol{q}}^{\mathsf{T}}. \tag{50}$$

Therefore, the velocity manipulability Jacobian including joint velocity limits is given by

$$\tilde{\boldsymbol{\mathcal{J}}}^{\dot{\boldsymbol{x}}} = \frac{\partial \boldsymbol{J}}{\partial \boldsymbol{q}} \times_2 \boldsymbol{J}\boldsymbol{W}^{\dot{\boldsymbol{q}}}\boldsymbol{W}^{\dot{\boldsymbol{q}}\mathsf{T}} + \frac{\partial \boldsymbol{J}^{\mathsf{T}}}{\partial \boldsymbol{q}} \times_1 \boldsymbol{J}\boldsymbol{W}^{\dot{\boldsymbol{q}}}\boldsymbol{W}^{\dot{\boldsymbol{q}}\mathsf{T}}. \tag{51}$$

Figure 6b shows the effect of including the actuator contribution when tracking a velocity manipulability ellipsoid. Notice that the robot joint q_1 significantly moves when given the highest velocity limit. In contrast, its influence on the manipulability tracking task is minimal when given the lowest velocity limit. This demonstrates the importance of considering the robot actuator specifications when tracking manipulability ellipsoids in real platforms.

In a similar way, the force manipulability ellipsoid considering the maximum joint torques is defined as $\tilde{\boldsymbol{M}}^{\boldsymbol{F}} = (\boldsymbol{J}\boldsymbol{\Omega}^{\boldsymbol{\tau}}\boldsymbol{J}^{\mathsf{T}})^{-1}$, where $\boldsymbol{\Omega}^{\boldsymbol{\tau}} = (\boldsymbol{W}^{\boldsymbol{\tau}}\boldsymbol{W}^{\boldsymbol{\tau}\mathsf{T}})^{-1}$ and $\boldsymbol{W}^{\boldsymbol{\tau}} = \text{diag}(\tau_{1,\max}, \dots, \tau_{n,\max})$. Then, the corresponding manipulability Jacobian is given by

$$\tilde{\boldsymbol{\mathcal{J}}}^{\boldsymbol{F}} = -\left(\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{q}} \times_2 \boldsymbol{J}\boldsymbol{\Omega}^{\boldsymbol{\tau}} + \frac{\partial \boldsymbol{J}^{\mathsf{T}}}{\partial \boldsymbol{q}} \times_1 \boldsymbol{J}\boldsymbol{\Omega}^{\boldsymbol{\tau}}\right) \times_1 (\tilde{\boldsymbol{M}}^{\boldsymbol{F}})^{-1} \times_2 (\tilde{\boldsymbol{M}}^{\boldsymbol{F}})^{-1}.$$

Finally, the dynamic manipulability ellipsoid considering the maximum joint torques is $\tilde{\boldsymbol{M}}^{\ddot{\boldsymbol{x}}} = \boldsymbol{\Upsilon}\boldsymbol{\Omega}^{\boldsymbol{\tau}-1}\boldsymbol{\Upsilon}^{\mathsf{T}}$ with corresponding manipulability Jacobian defined as

$$\tilde{\boldsymbol{\mathcal{J}}}^{\ddot{\boldsymbol{x}}} = \frac{\partial \boldsymbol{\Upsilon}}{\partial \boldsymbol{q}} \times_2 \boldsymbol{\Upsilon}\boldsymbol{\Omega}^{\boldsymbol{\tau}-1} + \frac{\partial \boldsymbol{\Upsilon}^{\mathsf{T}}}{\partial \boldsymbol{q}} \times_1 \boldsymbol{\Upsilon}\boldsymbol{\Omega}^{\boldsymbol{\tau}-1}. \tag{52}$$

4.4 Exploiting 4th-order precision matrix as controller gain

An open problem regarding the proposed tracking approach is how to specify the values of the gain matrix $\boldsymbol{K}_M$, which basically determines how the manipulability tracking error affects the resulting joint velocities. In this sense, we propose to define $\boldsymbol{K}_M$ as a precision matrix, which describes how accurately the robot should track a desired manipulability ellipsoid. In learning from demonstration applications, such gain matrix would typically be set as proportional to the inverse of the observed covariance $\boldsymbol{\mathcal{S}}$ (see Section 3.2). This encapsulates variability information of the task to be learned. Our goal here is to exploit this information to demand the robot a high precision tracking for directions in which low variability is observed, and vice-versa.

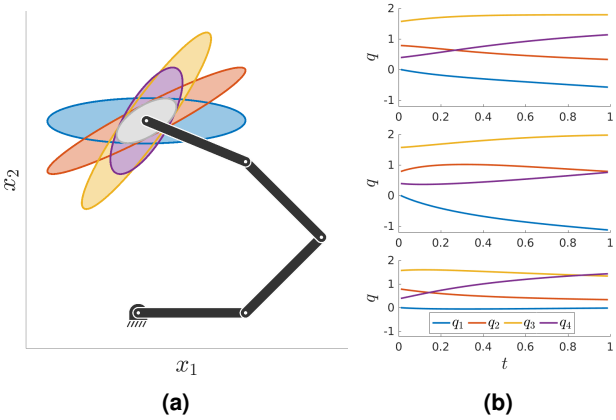
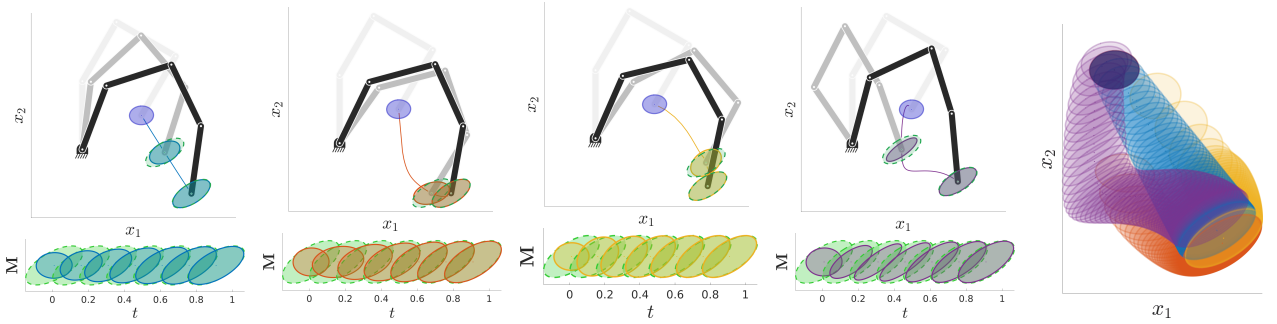


Figure 6. Illustration of actuators contribution. (*a*) Velocity manipulability ellipsoids obtained when setting a maximum joint velocity, for each joint, five times higher than the rest. The manipulability corresponding to equal maximum joint velocity is shown in gray. (*b*) Joint trajectories obtained with manipulability tracking (as in Fig. 5a) for equal maximum joint velocities (*top*), highest velocity limit for q_1 (*middle*), and lowest velocity limit for q_1 (*bottom*).

We therefore introduce the required precision $\boldsymbol{\mathcal{S}}^{-1}$ for a given manipulability tracking task into the controllers defined in Section 4.2. To do so, we define the gain matrix $\boldsymbol{K}_M$ as a function of the precision tensor. Specifically, we define the controller gain matrix as a full SPD matrix, which is computed from the matricization of the precision tensor $\boldsymbol{\mathcal{S}}^{-1}$ along its two first dimensions, with a proportion defined by

$$\boldsymbol{K}_M \propto \mathcal{S}^{-1}_{(1,2)}. \tag{53}$$

To show how precision matrices work as controller gains in our manipulability tracking problem, we tested different forms of $\boldsymbol{K}_M$ aimed at reproducing a given manipulability ellipsoid as a main task with a simulated 4-DoF planar robot. The robot is required to move its joints to track a desired manipulability ellipsoid, where the controller gain matrix $\boldsymbol{K}_M$ is a diagonal matrix with the diagonal elements of (53) to take into account the variation of each component of the manipulability ellipsoid. We tested four different precision tensors. First, equal variability for all components of the manipulability ellipsoid matrix is given. Then, the variability along the first or the second main axis of the manipulability ellipsoid, corresponding to the first and second diagonal elements of the gain matrix $\boldsymbol{K}_M$, is reduced. This means that the robot needs to prioritize the tracking of one of the ellipsoid main axes over the other. In the fourth test, the variability of the correlation between the two main axes of the manipulability ellipsoid is lowered. In this last case,



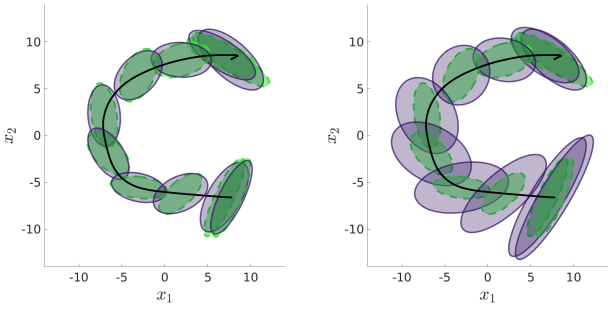
(a) Baseline **(b)** Precision along x_1 **(c)** Precision along x_2 **(d)** Diagonal precision **(e)** Trajectories

Figure 7. Manipulability tracking as main task with diagonal gain matrices defined from different precision tensors. The *top* plots depict the end-effector trajectory (solid colored line) and the posture of the robot along with the corresponding manipulability at time $t = 0,\ 0.25$ and 1s. The evolution of the manipulability along time is shown in the *bottom* plots. (*a*): equal tracking precision for all components. (*b*) and (*c*): tracking precision is $10:1$ higher for x_1 and x_2, respectively. (*d*): correlation between x_1 and x_2 axes is assigned a high tracking accuracy. (*e*) Evolution of the robot manipulability and end-effector trajectory for the gain matrices used in (*a*)-(*d*). The colors match those of the previous graphs. Initial and desired manipulability ellipsoids are depicted in dark blue and green on all graphs. Time t is in seconds.

the manipulability controller prioritizes the tracking of the ellipsoid orientation over the shape.

Figure 7 shows how the manipulator posture is adapted to track the desired manipulability ellipsoid with a priority on the component with the lowest variability. Note that when high tracking precision is required for one of the main axes of the ellipsoid, the robot initially seeks to fit the shape of the ellipsoid along that specific axis, and subsequently it matches the whole manipulability ellipsoid. In this case, the precision ratio between the prioritized and the rest of components of the gain matrix is $10:1$. When high tracking precision is assigned to the correlation of the ellipsoid axes, the robot first tries to align its manipulability with the orientation of the desired ellipsoid, and afterwards the whole manipulability is matched. In this case, the precision ratio between the prioritized correlation and the other components of the gain matrix is $3:1$. Notice that the precision tensor naturally affects the computed joint velocities required to track a given ellipsoid, which consequently influences the resulting motion of the end-effector as a function of the precision constraints, as shown in Fig. 7e. After convergence, the desired manipulability ellipsoid is successfully matched for all experiments. These results show that our geometry-aware tracking permits to take into account the variability information of a task to define the manipulability tracking precision.

Therefore, our manipulability tracking approach may be readily combined with the manipulability learning framework introduced in Section 3. In order to illustrate this, we show the reproduction phase of the experiment carried out in Section 3.2. The 5-DoF *student* robot was requested to track a desired Cartesian trajectory as main task, while varying its joint configuration for matching desired manipulability ellipsoids as secondary task. The student robot used the geometry-aware controller defined by (39), where $\boldsymbol{K}_M$ was defined either as a scalar value or as a diagonal matrix with the diagonal elements of (53) with the precision tensor being equal to the inverse of the covariance tensor $\hat{\boldsymbol{S}}^{\mathcal{O}\mathcal{O}}_{\mathcal{O}\mathcal{O}}$ retrieved by GMR (29). Our goal here was to exploit the learned variability information of the task to demand the robot a high precision tracking



(a) $\boldsymbol{K}_M$ as a scalar **(b)** $\boldsymbol{K}_M$ from GMR

Figure 8. Reproductions of a learned C-shape tracking task with desired manipulability ellipsoids. The end-effector trajectory is shown in black solid line, while the desired and reproduced manipulabilities are depicted in green and dark purple, respectively. (*a*) $\boldsymbol{K}_M$ is a scalar value, (*b*) $\boldsymbol{K}_M$ is the diagonal of the precision tensor retrieved by GMR. The required tracking precision is higher at the start and end of the task as a consequence of the low observed variability.

where low variability was observed in the demonstrations, and vice-versa. Successful reproductions of the demonstrated task using our manipulability-based redundancy resolution controller with scalar and variability-based matrix gains are shown in Figures 8a and 8b, respectively. Note that the variability-based matrix gain changes the required tracking precision, where higher precision is enforced only at the beginning and the end of the task, which results in lower control efforts in between. These results validate that the proposed approach allows the robot to reproduce reference profiles of desired manipulability ellipsoids while adapting the tracking precision according to the demonstrated requirements of the task.

4.5 Nullspace of the manipulability Jacobian

As traditionally done when designing redundancy resolution controllers, the nullspace of the manipulability Jacobian can also be exploited to fulfill secondary objectives when manipulability tracking is the main task. More specifically, a joint velocity $\dot{\boldsymbol{q}}_N$, aimed at fulfilling secondary objectives,

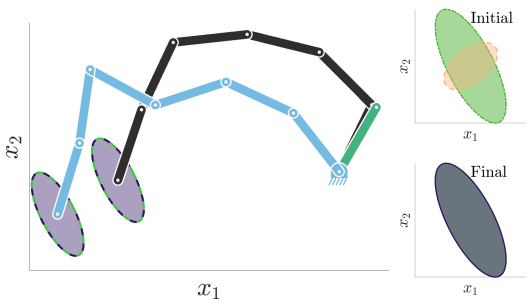


Figure 9. Use of the nullspace of the manipulability Jacobian. Two 6-DoF planar robots are required to track a desired manipulability ellipsoid as main task. The black robot also keeps its first joint at a fixed position (depicted by the green link), which is a secondary objective projected into the nullspace of the manipulability Jacobian. The final manipulability ellipsoids (in purple) fully overlap the desired ones (in green), showing a precise manipulability tracking. The initial manipulability ellipsoid is depicted in yellow.

can be projected into the nullspace of our manipulability tracking controller (38) using the nullspace operator $\left(\boldsymbol{I} - (\boldsymbol{\mathcal{J}}^{\dagger}_{(3)})^{\mathsf{T}}\boldsymbol{\mathcal{J}}^{\mathsf{T}}_{(3)}\right)$. Therefore, the resulting redundancy resolution controller is given by

$$\dot{\boldsymbol{q}}_t = (\boldsymbol{\mathcal{J}}^{\dagger}_{(3)})^{\mathsf{T}}\,\boldsymbol{K}_{\boldsymbol{M}}\,\mathrm{vec}\Big(\mathrm{Log}_{\boldsymbol{M}_t}(\hat{\boldsymbol{M}}_t)\Big) + \left(\boldsymbol{I} - (\boldsymbol{\mathcal{J}}^{\dagger}_{(3)})^{\mathsf{T}}\boldsymbol{\mathcal{J}}^{\mathsf{T}}_{(3)}\right)\dot{\boldsymbol{q}}_N. \quad (54)$$

In order to show the functionality of this nullspace operator, we carried out experiments with a simulated 6-DoF planar robot. The main task of the robot is to track a desired manipulability ellipsoid while keeping a desired pose for its first joint q_0, which is considered as secondary task. Thus, the nullspace velocity is defined as a simple proportional controller $\dot{\boldsymbol{q}}_N = \boldsymbol{K}^{\mathcal{P}}_{\boldsymbol{q}}(\hat{\boldsymbol{q}} - \boldsymbol{q}_t)$ where $\hat{\boldsymbol{q}}$ is the desired joint configuration and $\boldsymbol{K}^{\mathcal{P}}_{\boldsymbol{q}}$ is a matrix gain defined so that only joint position errors in the first joint are compensated. Figure 9 shows that the black manipulator configuration is adjusted to track the desired manipulability ellipsoid and keep, as accurately as possible, the desired joint position for q_0. Note that the black robot is able to find an alternative joint configuration that permits not only to closely track the desired manipulability, but also fulfill secondary objectives projected into its nullspace, in contrast to the blue robot which exclusively implements a manipulability tracking task. These results show that the nullspace of the manipulability Jacobian is suitable to carry out a secondary task along with manipulability tracking as main objective.

5 Importance of geometry-awareness

In the previous sections we introduced a geometry-aware manipulability transfer framework composed of *(1)* a probabilistic model that encodes and retrieves manipulability ellipsoids, and *(2)* manipulability tracking controllers. In this section, we show that the geometry-awareness of our formulations is crucial for successfully learning and tracking manipulability ellipsoids in addition to providing an appropriate mathematical treatment of both problems.

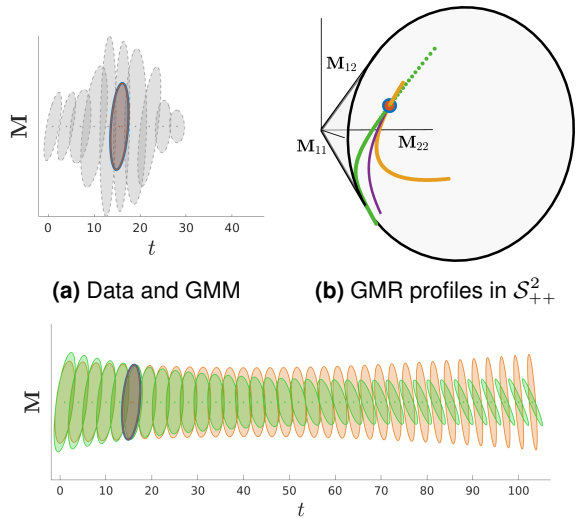
5.1 Learning

We first evaluate the proposed learning formulation compared to a framework that ignores that manipulability ellipsoids belong to the SPD manifold. To do so, we encode a distribution of manipulability ellipsoids with a GMM acting in the Euclidean space and we then retrieve desired manipulability ellipsoids via the corresponding GMR. To ensure the validity of the desired manipulability ellipsoids, GMM and GMR are performed on lower triangular matrices $\boldsymbol{L}$ obtained via Cholesky decomposition. Thus, the positive-definiteness of the desired manipulability ellipsoids computed as $\hat{\boldsymbol{M}} = \hat{\boldsymbol{L}}\hat{\boldsymbol{L}}^{\mathsf{T}}$ is guaranteed, where $\hat{\boldsymbol{L}}$ is the estimated GMR output. Note that this property is not guarantee in the case where GMM and GMR acting in the Euclidean space is applied directly to the manipulability ellipsoids $\boldsymbol{M}$. Therefore, we do not consider this approach in the comparison as the desired matrices $\hat{\boldsymbol{M}}$ may not be manipulability ellipsoids in some cases.

Figure 10 compares the proposed approach (Section 3) and the manipulability learning using GMM/GMR acting in Euclidean space. The demonstration consists of a time series of changing manipulability ellipsoids. For each approach, a 1-state GMM is trained and a reproduction is carried out for a longer time period than the demonstration using GMR. Both geometry-aware and Euclidean approaches obtain similar means of the GMM component (see Fig. 10a, 10b). This is due to the fact that the Euclidean mean computed using the Cholesky decomposition is a good approximation of the mean computed on $\mathcal{S}^{D}_{++}$ if the SPD data are close enough to each other. However, the covariances of the GMM components of both approaches are not equivalent. Indeed, the covariance of our geometry-aware approach is computed using the SPD data projected in the tangent space of the mean, while that of the Euclidean GMM corresponds to the covariance of the elements of the vectorized Cholesky decomposition, which ignores the geometry of the SPD manifold.

The manipulability ellipsoids profiles retrieved by the geometry-aware and Euclidean GMR are similar around the mean of the GMM component, but diverge when moving away from it (see Fig. 10c). This is because the estimated output in Euclidean space is only a valid approximation for input data lying close to the mean. In contrast, our approach is able to extrapolate the rotating behavior of the demonstrated manipulability ellipsoids as the recovered trajectory follows a geodesic on the SPD manifold (see Fig. 10b). Note that this is the equivalent to following a straight line in Euclidean space, which is the expected result of a trajectory computed via Gaussian conditioning. This behavior is obtained by parallel transporting the GMM covariances to the tangent space of the mean of the estimated conditional distribution of GMR (30). Therefore, the Euclidean GMR does not recover a trajectory following a geodesic on the manifold, leading to inconsistent extrapolated manipulability ellipsoids.

The reported results show that our geometry-aware approach accurately reproduces the behavior of the demonstrated data, and therefore provides a mathematically sound method for learning and retrieving manipulability ellipsoids in the SPD manifold. Note that similar behaviors are observed for GMM with any number of states, the



(a) Data and GMM **(b)** GMR profiles in $\mathcal{S}^2_{++}$

(c) GMM and retrieved GMR profiles

Figure 10. Importance of geometry in manipulability learning formulations. (*a*) Demonstrated data (depicted in light gray), and mean of the GMM component for the geometry-aware and Euclidean approaches (overlapping blue and red ellipsoids, respectively). (*c*) Manipulability profiles retrieved by the geometry-aware and Euclidean GMR, shown as green and orange ellipses, respectively. The time axis is shared with (*a*). (*b*) Mean of the GMM component and estimated profiles in the cone of SPD matrices. The manipulability profile obtained by our approach, shown in green, follows a geodesic. The profile obtained by the Euclidean framework is depicted by the orange curve and does not follow a geodesic on the manifold. The geodesic containing the mean of the Euclidean GMM, being a geometrically valid trajectory (depicted in purple), does not correspond to the trajectory obtained with the Euclidean framework. Thus, the Euclidean approach is geometrically flawed.

number $K = 1$ was chosen here to facilitate the visualization of the results.

5.2 Tracking

5.2.1 Comparisons with Euclidean tracking After showing the importance of geometry for learning manipulability ellipsoids, we compare the proposed tracking formulation against a controller ignoring the geometry of SPD matrices (i.e., treating the problem as Euclidean). Moreover, we evaluate our controller when the tracking of manipulability ellipsoids is assigned a secondary role. This evaluation compares our formulation against three Euclidean controllers, and the gradient-based approach in (Rozo et al. 2017). For the case in which the manipulability tracking is the main objective, we consider a 4-DoF planar robot that is required to track a desired manipulability ellipsoid by minimizing the error between its current and desired manipulability ellipsoids $\boldsymbol{M}$ and $\hat{\boldsymbol{M}}$. We first compare the proposed approach (38) with the following Euclidean manipulability tracking controller

$$\dot{\boldsymbol{q}}_t = (\boldsymbol{\mathcal{J}}^{\dagger}_{(3)})^{\mathsf{T}} \boldsymbol{K}_{\boldsymbol{M}} \text{vec}(\hat{\boldsymbol{M}}_t - \boldsymbol{M}_t), \tag{55}$$

where the difference between two manipulability ellipsoids is computed in Euclidean space, i.e., ignoring that manipulability ellipsoids belong to the set of SPD matrices. Secondly, we compare the proposed approach to the Cholesky-based Euclidean manipulability controller

$$\dot{\boldsymbol{q}}_t = (\boldsymbol{\mathcal{J}}^{\dagger}_{(3)})^{\mathsf{T}} \boldsymbol{K}_{\boldsymbol{M}} \text{vec}(\boldsymbol{\Delta L}_t \boldsymbol{\Delta L}_t^{\mathsf{T}}), \tag{56}$$

where $\boldsymbol{\Delta L} = \hat{\boldsymbol{L}} - \boldsymbol{L}$ and matrices $\boldsymbol{L}$ are obtained from the Cholesky decomposition such that $\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^{\mathsf{T}}$. This controller ensures that the difference between two manipulability ellipsoids is positive definite, but ignores that they belong to the SPD manifold. For completeness, we also compare our approach with the Cholesky-Jacobian-based Euclidean manipulability controller

$$\dot{\boldsymbol{q}}_t = (\mathcal{J}^{\dagger}_{\text{chol}(3)})^{\mathsf{T}} \boldsymbol{K}_{\boldsymbol{M}} \text{ vec}(\hat{\boldsymbol{L}} - \boldsymbol{L}), \tag{57}$$

where $\mathcal{J}_{\text{chol}} = \frac{\delta \boldsymbol{L}}{\delta \boldsymbol{q}} = \frac{\delta \boldsymbol{L}}{\delta \boldsymbol{M}} \boldsymbol{\mathcal{J}}$ is the Cholesky-based manipulability Jacobian, so that $\dot{\boldsymbol{L}} = \mathcal{J}_{\text{chol}} \times_3 \dot{\boldsymbol{q}}^{\mathsf{T}}$. This approach tracks a desired manipulability solely through its Cholesky decomposition with an adapted manipulability Jacobian. Similarly to (56), it ensures the positive-definiteness of manipulability ellipsoids, but ignores that they belong to the SPD manifold. For all the following comparisons, the gain matrices $\boldsymbol{K}_{\boldsymbol{M}}$ are identity matrices.

Figure 11 shows the convergence rate for the proposed geometry-aware controller, the Euclidean-based approach, the Cholesky-based Euclidean and Cholesky-Jacobian-based Euclidean formulations. Two tests were carried out by varying the initial configuration of the robot and the desired manipulability ellipsoid. In the first case, the Euclidean and geometry-aware formulations converge to similar robot joint configurations with a distance between the current and desired manipulability close to zero (see Fig. 11a-*left*, *middle* and Table 2). However, in the second test, the Euclidean formulation induces a sudden change in the joint configuration, resulting in an abrupt increase on the error measured between the current and desired manipulability ellipsoids (see Fig. 11b-*left*, *middle*). In real scenarios, such unstable robot behavior would certainly be harmful and unsafe. This erroneous tracking performance can be explained by the fact that the Euclidean path between two SPD matrices is a valid approximation of the geodesic only if these are close enough to each other, as shown in Fig. 11a-*right*. When this approximation is not valid (see Fig. 11b-*right*), the Euclidean controller outputs inconsistent reference joint velocities that destabilize the robotic system, therefore failing to track the desired manipulability. Note that the Cholesky-based Euclidean formulation does not converge in both cases (see Table 2) and induces a sudden change in joint configuration of the robot in the second scenario, similarly to the Euclidean formulation. This can be explained by the fact that the path induced by this method is not close to geodesics on the SPD manifold as shown by Fig. 11-*right*. As opposed to the two Euclidean formulations, the Cholesky-Jacobian-based Euclidean controller does not induce unstable robot behaviors and converges towards the desired manipulability ellipsoid for both cases. However, this method shows a poor convergence rate compared to our geometry-aware approach, as shown by Fig. 11-*left*. This can be explained by the fact that, although this approach generates curved paths on the SPD manifold, these paths do not resemble geodesics and tend to induce detours to reach the desired manipulability ellipsoid (see Fig. 11-*left*). This

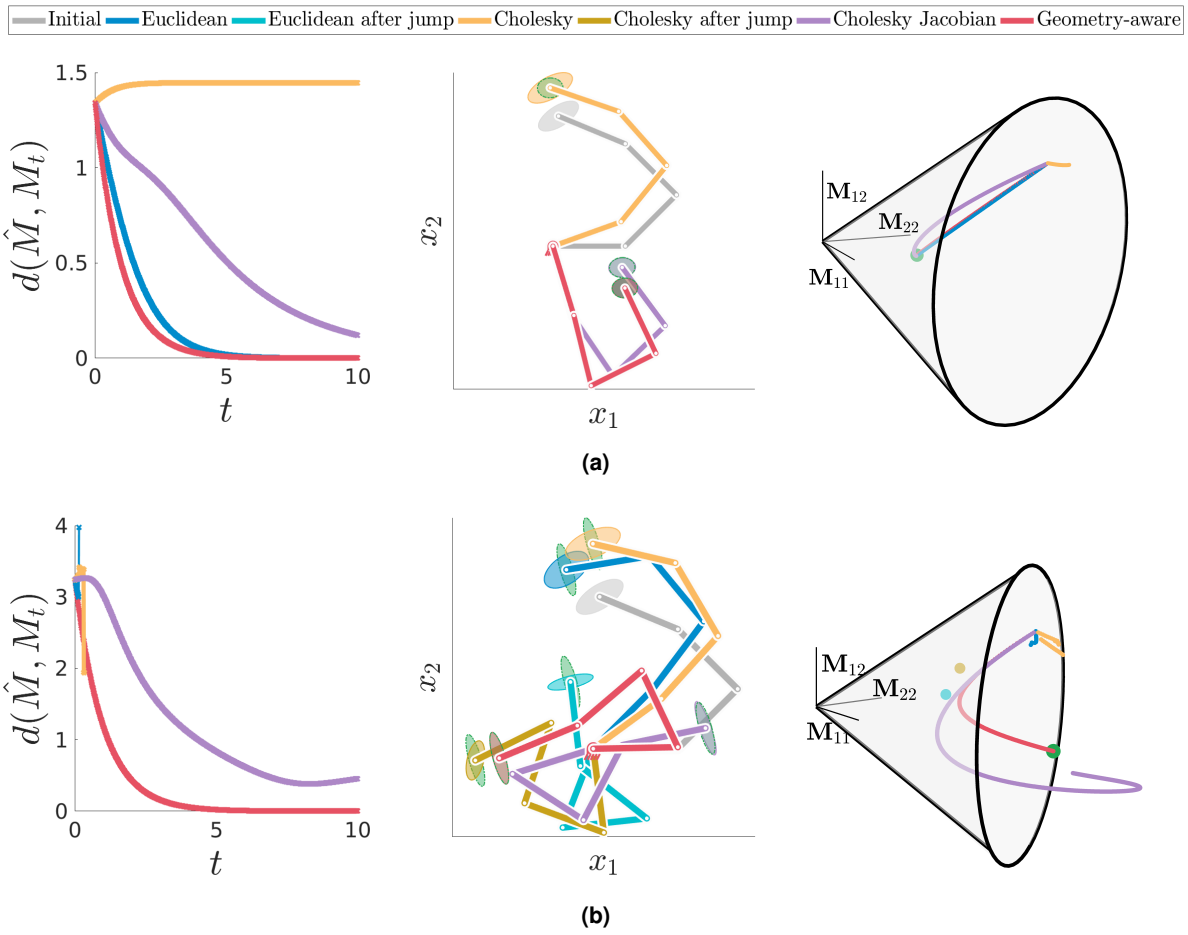


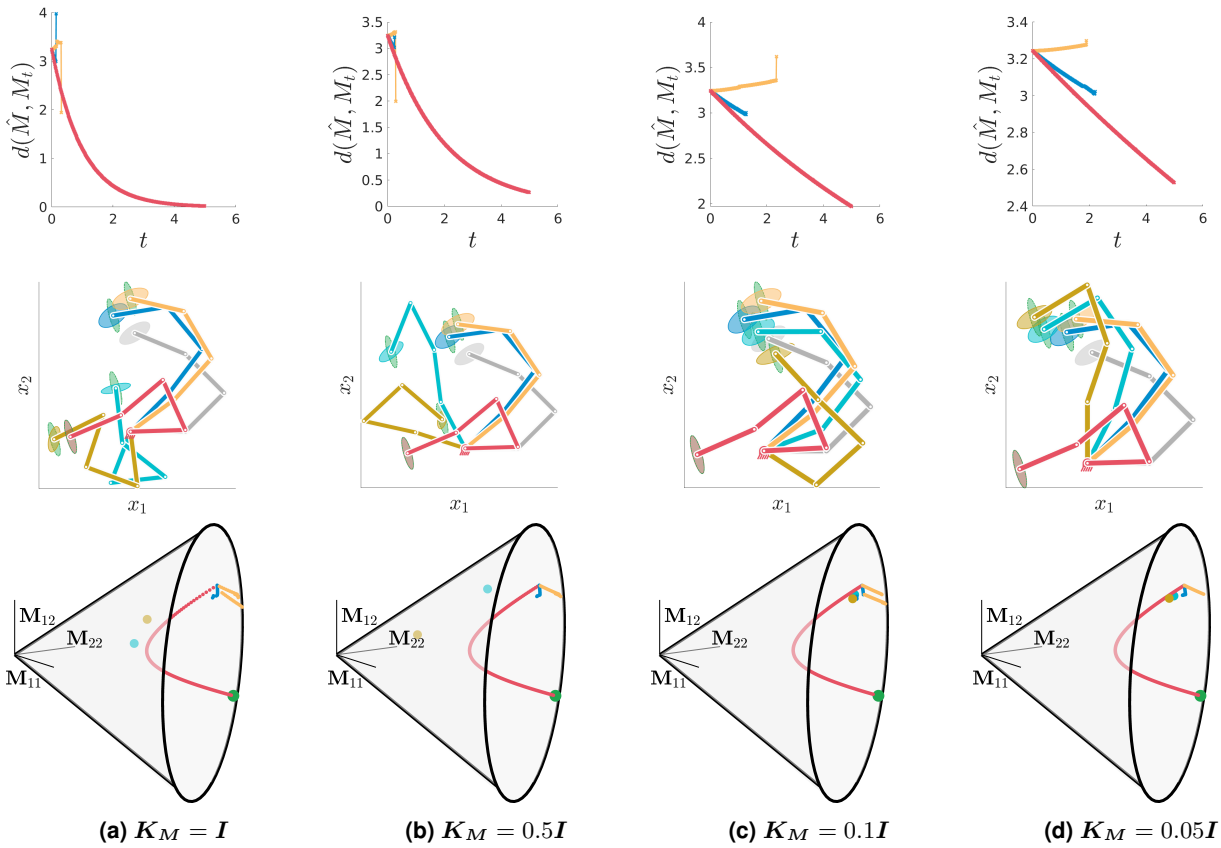
Figure 11. Performance of different manipulability tracking formulations. The *left* graphs show the affine-invariant distance between the current and desired manipulability ellipsoids over time. The distances for the Euclidean, Cholesky-based Euclidean, Cholesky-Jacobian-based Euclidean and geometry-aware approaches are respectively depicted in blue, yellow, lila and red. The *middle* graphs display the initial and final robot postures and the final manipulability ellipsoids. The initial posture is depicted in light gray, while the final posture and corresponding manipulability for the three methods are depicted in the same color as the distances. The desired manipulability is depicted in green. *Middle-(b)* also shows the sudden change in the robot posture for both Euclidean methods (55) and (56). The robot posture before and after the abrupt change is shown in blue and light blue, respectively for (55) and in yellow and olive, respectively for (56). The *right* graphs depict the evolution of the manipulability ellipsoids in the SPD manifold. The colors correspond to those of the previous graphs with the green dot representing the desired manipulability. The isolated light blue and olive dots in the *bottom-(b)* graph represent the manipulability ellipsoids after the abrupt changes in the robot joint configuration.

Table 2. Final distances $d(\hat{\boldsymbol{M}}, \boldsymbol{M}_t)$ between the current and desired manipulability ellipsoids for the performance comparison of the different manipulability tracking formulations.

Approach	Euclidean	(after jump)	Cholesky	(after jump)	Cholesky Jacobian	Geometry-aware
Fig. 11a	$1.3e^{-4}$	-	1.446	-	0.1204	$6e^{-5}$
Fig. 11b	2.997	3.977	3.385	1.944	0.455	$1.4e^{-4}$

is particularly visible for the second test, where the resulting joint configuration is farther from the initial pose of the robot compared to the joint configuration obtained by the proposed geometry-aware controller (see Fig. 11b-*middle*, *right*).

Previously, we hypothesized that the sudden changes in joint configuration when using the Euclidean and Cholesky-based Euclidean formulations in the second scenario are due to the path induced by the methods on the SPD manifold. In order to confirm this hypothesis, we reproduced the second test with lower gain values. Figure 12 shows the convergence of the proposed geometry-aware controller, the Euclidean-based approach and the Cholesky-based Euclidean formulation for gain matrices equal to $\boldsymbol{I}$, $0.5\boldsymbol{I}$, $0.1\boldsymbol{I}$ and $0.05\boldsymbol{I}$. We observe that, even for very low gains, both Euclidean and Cholesky-based Euclidean formulations lead to a sudden change in the joint configuration, resulting in an abrupt increase on the error measured between the current and desired manipulability ellipsoids (see Fig. 12-*top*, *middle*). Interestingly, the sudden changes occur at similar location along the path between the initial and desired manipulability ellipsoid independently of the gain value for both formulations (see Fig. 12-*bottom*), therefore confirming our above statement. This can also be seen by looking at the yellow and dark blue robots of Fig. 12-*middle* depicting the configurations before the jump, which are almost identical in all the graphs.



(a) $\boldsymbol{K}_M = \boldsymbol{I}$ **(b)** $\boldsymbol{K}_M = 0.5\boldsymbol{I}$ **(c)** $\boldsymbol{K}_M = 0.1\boldsymbol{I}$ **(d)** $\boldsymbol{K}_M = 0.05\boldsymbol{I}$

Figure 12. Comparison of the performance of different manipulability tracking formulations for different gains $\boldsymbol{K}_M$. The organization of the graphs and the colors are identical to Fig. 11. The Cholesky-Jacobian-based Euclidean formulation is not shown.

In the case in which the manipulability tracking task becomes a secondary objective, the 4-DoF planar robot is required to keep its end-effector at a fixed Cartesian position $\hat{\boldsymbol{x}}$ while minimizing the distance between its current and desired manipulability ellipsoids $\boldsymbol{M}$ and $\hat{\boldsymbol{M}}$. The four following approaches are considered for comparison with the proposed formulation (39). Firstly, we analyze the corresponding Euclidean manipulability-tracking controller

$$\dot{\boldsymbol{q}}_t = \boldsymbol{J}^\dagger \boldsymbol{K}_{\boldsymbol{x}}(\hat{\boldsymbol{x}}_t - \boldsymbol{x}_t) + (\boldsymbol{I} - \boldsymbol{J}^\dagger \boldsymbol{J})(\boldsymbol{\mathcal{J}}^\dagger_{(3)})^\mathsf{T} \boldsymbol{K}_M \mathrm{vec}(\hat{\boldsymbol{M}}_t - \boldsymbol{M}_t), \quad (58)$$

where the difference between two manipulability ellipsoids is computed in Euclidean space, i.e., ignoring that manipulability ellipsoids belong to the set of SPD matrices. Secondly, we implement the corresponding Cholesky-based Euclidean manipulability controller

$$\dot{\boldsymbol{q}}_t = \boldsymbol{J}^\dagger \boldsymbol{K}_{\boldsymbol{x}}(\hat{\boldsymbol{x}}_t - \boldsymbol{x}_t) + (\boldsymbol{I} - \boldsymbol{J}^\dagger \boldsymbol{J})(\boldsymbol{\mathcal{J}}^\dagger_{(3)})^\mathsf{T} \boldsymbol{K}_M \mathrm{vec}(\boldsymbol{\Delta L}_t \boldsymbol{\Delta L}_t^\mathsf{T}), \quad (59)$$

which ignores that manipulability ellipsoids lie on the SPD manifold but ensure a positive definite difference between two ellipsoids. Thirdly, we analyze the Cholesky-Jacobian-based Euclidean manipulability controller

$$\dot{\boldsymbol{q}}_t = \boldsymbol{J}^\dagger \boldsymbol{K}_{\boldsymbol{x}}(\hat{\boldsymbol{x}}_t - \boldsymbol{x}_t) + (\boldsymbol{I} - \boldsymbol{J}^\dagger \boldsymbol{J})(\boldsymbol{\mathcal{J}}^\dagger_{\mathrm{chol}(3)})^\mathsf{T} \boldsymbol{K}_M \mathrm{vec}(\hat{\boldsymbol{L}} - \boldsymbol{L}), \quad (60)$$

Table 3. Final distances $d(\hat{\boldsymbol{M}}, \boldsymbol{M}_t)$ between the current and desired manipulability ellipsoids for the performance comparison of the different manipulability-based redundancy resolution formulations.

Approach	Euclidean	Cholesky	Cholesky Jac.	Geometry-aware	Gradient-based
Fig. 13a	0.433	0.808	1.418	0.416	0.436
Fig. 13b	1.763	2.271	1.856	1.101	1.110

which tracks manipulability ellipsoids through their Cholesky decomposition. Fourthly, we evaluate the gradient-based approach of (Rozo et al. 2017) that implements the controller

$$\dot{\boldsymbol{q}}_t = \boldsymbol{J}^\dagger \boldsymbol{K}_{\boldsymbol{x}}(\hat{\boldsymbol{x}}_t - \boldsymbol{x}_t) - (\boldsymbol{I} - \boldsymbol{J}^\dagger \boldsymbol{J})\alpha \nabla g_t(\boldsymbol{q}), \quad (61)$$

where α is a scalar gain and

$$g_t(\boldsymbol{q}) = \log\det\left(\frac{\hat{\boldsymbol{M}}_t + \boldsymbol{M}_t}{2}\right) - \frac{1}{2}\log\det\left(\hat{\boldsymbol{M}}_t \boldsymbol{M}_t\right) \quad (62)$$

is a cost function based on Stein divergence (a distance-like function on the SPD manifold (Sra 2012)). The gain matrices $\boldsymbol{K}_M$ are fixed as identity matrices and the scalar gain is set to 1 for the comparison.

Figure 13 shows the convergence rate for the manipulability-based redundancy resolution of the aforementioned approaches. Two tests were carried out by varying the initial configuration of the robot and the desired manipulability ellipsoid. In both cases, both

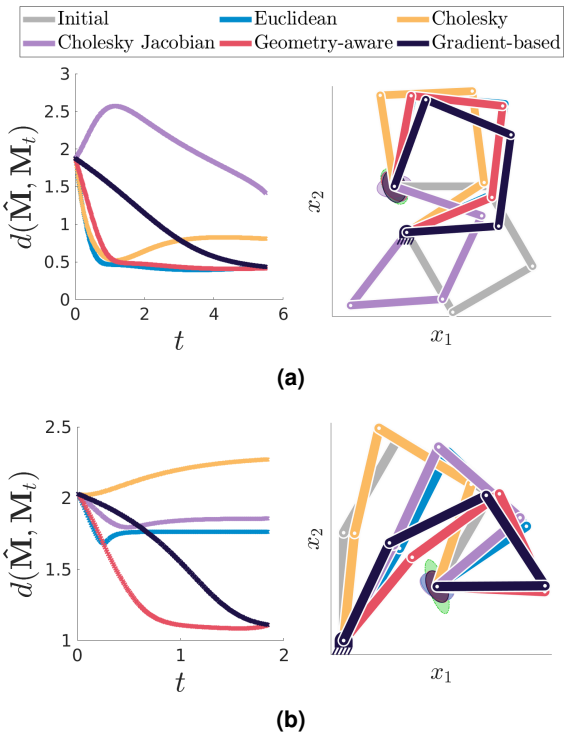


Figure 13. Performance comparison of the different manipulability-based redundancy resolution formulations. Two cases are shown with varying initial robot configuration and desired manipulability. The *left* graph shows the convergence of the affine invariant distance between the current and the desired manipulability ellipsoid over time. The distances for the Euclidean, Cholesky-based Euclidean, Cholesky-Jacobian-based Euclidean, geometry-aware and gradient-based approaches are respectively depicted in blue, yellow, lila, red, and purple. The *right* graph shows the initial and final posture of the robot along with the final manipulability ellipsoids. The initial posture of the robot is depicted in light gray. The final postures and the corresponding manipulability ellipsoids for the different methods are depicted in the same color as the distances. The desired manipulability ellipsoid is depicted in green.

geometry-aware and gradient-based approaches converge to a similar final robot configuration (see Fig. 13a, 13b-*right*), with similar values of the affine-invariant distance between the final and desired manipulability ellipsoids (see Fig. 13a, 13b-*left* and Table 3). More importantly, the proposed geometry-aware manipulability tracking approach shows a faster convergence than the gradient-based method, with a lower computational cost (3.5 ms and 4.2 ms per time step, with non-optimized Matlab code on a laptop with 2.7GHz CPU and 32 GB of RAM). This notable difference may be attributed to the fact that despite both methods take into account the geometry of manipulability ellipsoids, our approach is more informative about the kinematics of the robot through the use of the manipulability Jacobian $\mathcal{J}(\boldsymbol{q})$.

Note that for some specific initial robot configurations and desired manipulability ellipsoids, the Euclidean manipulability-tracking controller (58) shows a slightly faster convergence rate than our method (see Fig. 13a). However, this Euclidean formulation again leads to unstable behaviors in some configurations (see Fig. 13b), where the distance between the final and desired manipulability ellipsoids remains high compared to the two geometry-aware approaches. This poor tracking performance can be attributed to the fact that the Euclidean difference between two SPD matrices is an approximation that is only valid if the matrices are close enough to each other. Thus, similarly to Euclidean controller aimed at tracking manipulability ellipsoids as first task (55), the Euclidean manipulability-based redundancy resolution is only effective if the current and desired ellipsoids are very similar. Moreover, the distance between the final and desired manipulability ellipsoids remains higher than for the geometry-aware methods and the Euclidean controller by using the Cholesky-based Euclidean manipulability-based redundancy resolution. This tendency is similar to the observations made for the tracking of manipulability ellipsoids as main objective and is due to the fact that the controller (59) induces paths on the manifold that are not close to geodesics. Furthermore, the Cholesky-Jacobian-based Euclidean controller shows a poor tracking performance for the two considered scenarios. Notably, the distance between the current and desired ellipsoids is largely increased before decreasing slowly in the first case (see Fig. 13a). Moreover, in some configurations, the final distance remains high compared to the geometry-aware approaches as shown by Fig. 13b. These behaviors are due to the fact that the controller (60) does not follow geodesic paths on the SPD manifold.

The reported results supported our hypothesis that geometry-aware manipulability controllers result in good tracking performance while providing stable convergence regardless of the manipulability tracking error. This was observed when manipulability tracking was the main task and a secondary objective of the robot. Moreover, our manipulability-based redundancy resolution approach outperforms the gradient-based method. Furthermore, our controller permits to directly exploit the variability information of a task, given in the form of a 4th-order covariance tensor, through the gain matrix of the controller. This allows the robot to exploit the precision required while tracking a manipulability ellipsoid either as main or secondary objective. This operation is not available in the gradient-based method used for comparison, since the corresponding controller gain is a scalar.

5.2.2 Comparisons with manipulability-based optimization We compare our tracking approach against two state-of-the-art manipulability-based optimization methods widely used to improve robots posture for task execution. We first evaluate our geometry-aware controller against manipulability volume maximization. Then, we compare our controller to the compatibility index maximization (Chiu 1987), where the distance from the ellipsoid center to its surface is maximized along a specified direction. To do so, we consider two 8-DoF planar robots that are required to track a desired Cartesian velocity trajectory that leads to an L-shape path in the Cartesian space. In order to achieve high dexterity in motion, the first robot is requested to track a desired manipulability ellipsoid whose main axis is elongated along the direction of motion. The second robot varies its posture in order to maximize either the manipulability volume or the compatibility index along the direction of motion.

Fig. 14a shows the resulting joint configurations and manipulability ellipsoids of the two robots at different stages of the task where the second robot maximizes

the manipulability volume as secondary objective. We observe that the main axis of the manipulability ellipsoid obtained with the volume maximization approach is often perpendicular to the direction of motion, which often occurs as this method does not consider any geometric information about the desired manipulability ellipsoid. Also, since the resulting posture leads to ellipsoids that are not consistent with the task requirement (task velocity control directions) and degrade the robot capabilities, this becomes unstable when the gain of the velocity tracking controller is increased to achieve higher Cartesian velocities, as shown in Fig. 15. Conversely, the robot tracking a desired manipulability ellipsoid successfully completes the task when higher velocities are required.

The main advantage of maximizing the compatibility index over the volume is that the directions in which the ellipsoid should be elongated are specified. However, this approach favors robot configurations that may be close to singularities as the manipulability ellipsoids corresponding to these posture are flat ellipsoids that can be largely elongated (see Fig. 14c). This effect exacerbates when the compatibility index maximization is the main task of the robot, as this is not required to match a specific position in Cartesian space. Chiu (1988) extended the compatibility index optimization approach by defining the compatibility cost as a weighted sum, allowing the maximization or minimization of the ellipsoid along several directions. This method provides more flexibility on the resulting ellipsoid due to the weighted combination, at the cost of a laborious tuning. Moreover, the orientation and elongation of the main axes of the ellipsoid after the optimization are hard to infer from the cost weights.

In contrast to the considered manipulability-based optimization methods, the proposed geometry-aware controllers seeks to fit the full desired manipulability ellipsoid in all its directions. Singular configurations can therefore be easily avoided by defining appropriate desired manipulability ellipsoids. Moreover, our manipulability controller allows the tracking of any manipulability ellipsoid, including those providing a compromise between dexterity in motion and force exertion along any axis. This is not possible when using the compatibility index approach of (Chiu 1987) as it always favors the dexterity in motion over force or vice-versa. Although this compromise might be achievable using the compatibility index approach of (Chiu 1988), our method does not require a laborious tuning process. Manipulability tracking is also hard to achieve through manipulability volume maximization as there is no explicit control on the resulting ellipsoid main axes.

6 Experiments

Previously, in our former work (Jaquier et al. 2018), we showed the benefits of including the manipulability redundancy resolution controller in the nullspace of a position controller for a pushing and an insertion task. Contrary to the result obtained by the position controller alone, the posture of the robot significantly varied during the execution of the tasks to be compatible with their respective force requirements as a consequence of the force manipulability tracking.

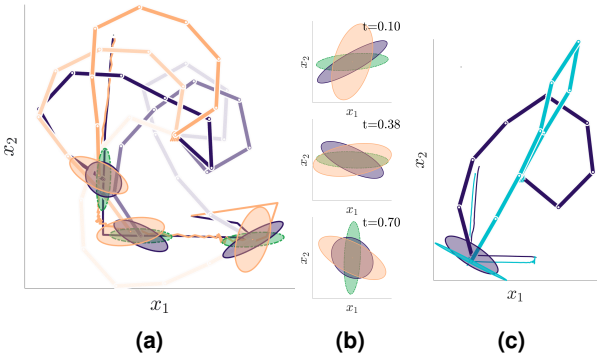


Figure 14. (*a*) Comparison of our manipulability tracking controller (in purple) with the manipulability volume maximization (in yellow). The main axis of the desired manipulability ellipsoids (in green) are aligned with the direction of motion in order to allow high velocities during the task execution. The robot colors become darker with the evolution of the movement. (*b*) Close-up plots of the manipulabilities represented in (*a*). (*c*) Comparison of our manipulability tracking controller (in purple) with the compatibility index maximization (in light blue).

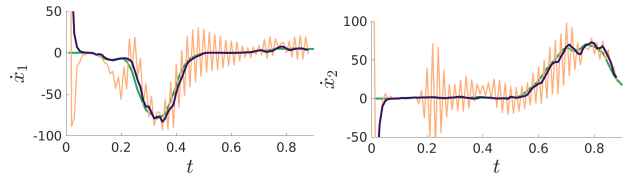


Figure 15. Cartesian velocities achieved with our manipulability tracking controller (purple) and the volume maximization approach (yellow) as secondary objective for a Cartesian velocity controller. The gain of the velocity controller are equal for both approaches. The desired velocities are shown in green.

In this section, we extensively evaluate the proposed tracking formulation with different robotic platforms and different types of manipulability ellipsoids in simulation. The approach is evaluated to track a desired manipulability for grasping with an Allegro hand (four 4-DoFs fingers) and to track a desired center of mass manipulability with NAO and Centauro robots (25- and 39-DoFs, respectively). We then illustrate and evaluate the proposed manipulability transfer approach in a bimanual task using a Baxter robot (two 7-DoFs arms) and a couple of Franka Emika Panda robots (7-DoFs).

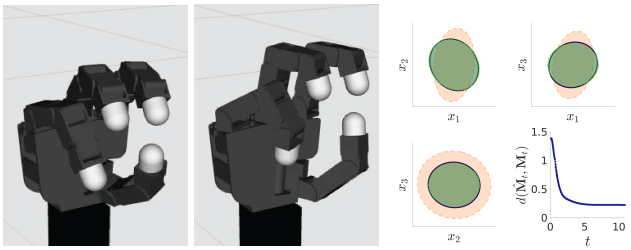
6.1 Manipulability tracking for a robotic hand

In the context of robotic hands, manipulability ellipsoids have been used to analyze their performances in grasping tasks (Prattichizzo et al. 2012). In this experiment, we aim at modifying the posture of a robotic hand to match a desired manipulability ellipsoid while grasping an object.

For the case of multiple arm systems, the set of joint velocities of constant unit norm $\|\dot{\boldsymbol{q}}_a\| = \|(\dot{\boldsymbol{q}}_1^\mathsf{T}, \dots, \dot{\boldsymbol{q}}_C^\mathsf{T})^\mathsf{T}\| = 1$ is mapped to the Cartesian velocity space $\dot{\boldsymbol{x}}_a = (\dot{\boldsymbol{x}}_1^\mathsf{T}, \dots, \dot{\boldsymbol{x}}_C^\mathsf{T})^\mathsf{T}$ through

$$\|\dot{\boldsymbol{q}}_a\|^2 = \dot{\boldsymbol{q}}_a^\mathsf{T}\dot{\boldsymbol{q}}_a = \dot{\boldsymbol{x}}_a^\mathsf{T}(\boldsymbol{G}_a^{\dagger\mathsf{T}}\boldsymbol{J}_a\boldsymbol{J}_a^\mathsf{T}\boldsymbol{G}_a^\dagger)^{-1}\dot{\boldsymbol{x}}_a, \qquad (63)$$

with the Jacobian $\boldsymbol{J}_a = \mathrm{diag}(\boldsymbol{J}_1, \dots, \boldsymbol{J}_C)$, the grasp matrix $\boldsymbol{G}_a = (\boldsymbol{G}_1, \dots, \boldsymbol{G}_C)$ and C the number of arms. Therefore, the velocity manipulability ellipsoid of the C-arms system



(a) Initial **(b)** Final **(c)**

Figure 16. Manipulability tracking for grasping tasks with the Allegro hand in simulation. *(a)* and *(b)* show the initial and final pose of the robot, respectively. *(c)* Initial, final and desired manipulability ellipsoids respectively depicted in yellow, dark purple and green. The *bottom-right* graph shows the evolution of the distance between the current and desired manipulability ellipsoid over time (in seconds).

is given by $\boldsymbol{M}^{\dot{\boldsymbol{x}}_a} = \boldsymbol{G}_a^{\dagger\mathsf{T}} \boldsymbol{J}_a \boldsymbol{J}_a^{\mathsf{T}} \boldsymbol{G}_a^{\dagger}$ (Chiacchio et al. 1991a). Note that the system is modeled under assumptions that the arms are holding a rigid object with a tight grasp.

In this first experiment, the Allegro hand was required to track a desired manipulability, while maintaining relative positions between the different fingers. This experiment aims at emulating how humans adapt their finger configuration to the task at hand while grasping an object. In this experiment, the desired velocity manipulability ellipsoid was designed by the experimenter to be a medium-size isotropic ellipsoid. The purpose of this design is to provide the hand with the capability to perform a displacement of the object while being resistant to external perturbations in all the directions. For example, in the case where the hand is holding a pen, it is desirable that the pen can be moved with dexterity, while the hand should resist to perturbations due to the pen-surface contacts.

The fingers were controlled according to a leader-follower strategy (Luh and Zheng 1987). Therefore the thumb joints were moved to track the desired manipulability ellipsoid using the controller (38) and the other fingers were required to maintain constant relative end-effector positions with respect to the thumb end-effector, while tracking the manipulability as secondary objective with the redundancy controller (39). The center of the object was considered as the central position between the four fingers of the hand and the contact points were assumed to be at the finger tips.

Figures 16a and 16b show an example of adaptation of the posture of the hand to track a desired velocity manipulability ellipsoid for a grasp defined by the user. As expected, the robot modified its joint configuration in order to match, as accurately as possible, the desired velocity manipulability (see Fig. 16c). Note that the manipulability tracking in this experiment can only be achieved partially, because the robotic hand is also required to maintain the initial grasp. Nevertheless, this tracking may be further improved if the dimensionality of the nullspace of the main task is higher (e.g. not all the finger tips are position-constrained), or using a higher DoF robotic hand.

6.2 Manipulability tracking for a humanoid center of mass

An interesting use of manipulability ellipsoids arises when these are defined at the center of mass (CoM) of humanoid robots, which permits to analyze their capabilities to accelerate the CoM in locomotion (Azad et al. 2017; Gu et al. 2015), or to evaluate how resistant they can be to external perturbations using the force manipulability at a specific humanoid posture. With the goal of getting some insights on the role of CoM manipulability ellipsoids in legged robots, we designed manipulability tracking experiments using two different floating-base robots in simulation, namely, the humanoid NAO and the Centauro robot (Baccelliere et al. 2017) within the Pyrobolearn framework (Delhaisse et al. 2019).

Specifically, we required the robots to track a desired manipulability ellipsoid defined at its CoM while keeping balance. We assumed a strict hierarchy of tasks that gave the highest priority to the task of maintaining the CoM position over the support polygon and zero velocity at all contact points with the floor, while the manipulability tracking was considered a secondary task. Under the aforementioned assumptions, we implemented the inverse kinematics-based controller for floating-base robots proposed in (Mistry et al. 2008), which we briefly introduce here. First, let us define the Jacobian for the primary task as

$$\boldsymbol{J}_b = \begin{bmatrix} \boldsymbol{J}_{\text{feet}} \\ \boldsymbol{J}_{\text{CoM,xy}} \end{bmatrix}, \tag{64}$$

where $\boldsymbol{J}_{\text{feet}}$ represents the Jacobians for the position/orientation of the robot feet while $\boldsymbol{J}_{\text{CoM,xy}}$ is the Jacobian for the projection of the CoM onto the (x, y) plane (assuming the gravity vector is in the z direction). Next, we define the vector of primary desired velocities $\boldsymbol{x}_b$ (i.e. velocities of the robot feet and CoM), noting that all the robot feet velocities must equal zero in order to maintain constraints, therefore

$$\dot{\boldsymbol{x}}_b = \begin{bmatrix} \boldsymbol{0} \\ \dot{\boldsymbol{x}}_{\text{CoM}} \end{bmatrix}, \tag{65}$$

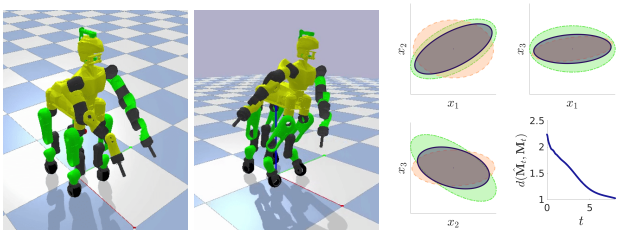
where $\dot{\boldsymbol{x}}_{\text{CoM}}$ is the velocity at the robot CoM so that it lies in the support polygon.

Regarding the secondary task, that is, the manipulability tracking at the robot CoM, we first compute the Jacobian at the CoM $\boldsymbol{J}_{\text{CoM}}$ for floating-base robots as in (Mistry et al. 2008), which allows us to calculate manipulability ellipsoids of the types introduced in Section 4. Depending on which type of manipulability we require the robot to track, we can use any of the manipulability Jacobians (34), (35) or (36) to compute the desired joint velocities $\dot{\boldsymbol{q}}$ for the manipulability tracking task using (38). So, the full joint velocity controller for legged robots required to keep balance while tracking a desired manipulability ellipsoid at their CoM is defined as

$$\dot{\boldsymbol{q}} = \begin{bmatrix} \boldsymbol{I}_{n\times n} \\ \boldsymbol{0}_{6\times n} \end{bmatrix}^{\mathsf{T}} \left(\boldsymbol{J}_b^{\dagger} \dot{\boldsymbol{x}}_b + \boldsymbol{N}_b \left(\boldsymbol{\mathcal{J}}_{(3)}^{\dagger}\right)^{\mathsf{T}} \boldsymbol{K}_M \operatorname{vec}\left(\operatorname{Log}_{M_t}(\hat{\boldsymbol{M}}_t)\right) \right), \tag{66}$$

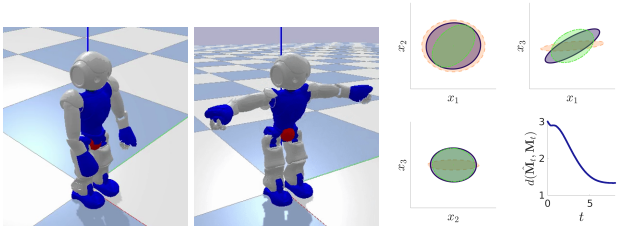
where the first term is included in order to account for the virtual joints of legged robots, n is the number of DoF of the robot, and $\boldsymbol{N}_b$ is the nullspace of the Jacobian (64).

We ran several experiments for testing the manipulability tracking at the CoM of the Centauro (Fig. 17) and NAO



(a) Initial **(b)** Final **(c)**

Figure 17. Tracking of the COM manipulability with the Centauro robot in simulation. *(a)* and *(b)* show the initial and final pose of the robot, respectively. *(c)* Initial, final and desired manipulability ellipsoids respectively depicted in yellow, dark purple and green. The *bottom-right* graph shows the evolution of the distance between the current and desired manipulability ellipsoid over time, given in seconds.



(a) Initial **(b)** Final **(c)**

Figure 18. COM manipulability tracking with NAO in simulation. *(a)* and *(b)* show the initial and final pose of NAO, respectively. The CoM of the robot is depicted by a red sphere. *(c)* Initial, final and desired manipulability ellipsoids respectively depicted in yellow, dark purple and green. The *bottom-right* graph shows the distance between the current and desired manipulability over time (given in seconds).

(Fig. 18) robots using the controller (66). The tests consisted of manually setting a desired manipulability ellipsoid to be tracked at the CoM of the robot, and running a joint velocity controller given the reference provided by (66). Notably, both Centauro and NAO tracked the desired manipulability as precisely as possible without compromising the balancing task. Figures 17c and 18c show the distance between the desired and current CoM manipulability, which decreases over time as the robot adapts its posture to carry out a good tracking while keeping its balance. An interesting aspect about defining and tracking CoM manipulability ellipsoids is the final posture that the robots achieve. Figure 17b shows the final posture achieved by Centauro when tracking a CoM manipulability whose projection on the (x_1, x_2) plane is a tilted ellipse, which makes the robot adopt a posture where the front legs and torso rotate on the same plane (which corresponds to the floor in the virtual environment). The final posture of NAO displayed in Fig. 18b shows that both arms are completely extended along the humanoid frontal axis, in an attempt to align them with one of the main axis of the CoM manipulability ellipsoid. However, both the balancing task and the lower number of DoF constrain NAO to closely match the desired manipulability.

6.3 Manipulability transfer between robots for a bimanual task

The performance of the proposed manipulability transfer framework was tested in a bimanual unplugging of an electric cable from a power socket. The central idea is to teach different dual-arm robots to execute a task requiring a specific manipulability profile via kinesthetic teaching provided only to one of the bimanual robots.

In the first part of the experiment, the two 7-DoF arms of a Baxter robot are kinesthetically guided to provide demonstrations (see Fig. 19a). The posture of the arms is modified by the user so that the main axis of the dual force manipulability ellipsoid of the system $\boldsymbol{M}^{\boldsymbol{F}_a} = (\boldsymbol{G}_a^{\dagger\mathsf{T}} \boldsymbol{J}_a \boldsymbol{J}_a^{\mathsf{T}} \boldsymbol{G}_a^{\dagger})^{-1}$ is aligned with the direction of extraction. Then, the arms are moved in opposite directions to unplug the electric cable from the socket. We extracted both the relative position $\boldsymbol{\Delta x}_t$ between the end-effectors of both arms and the force manipulability ellipsoid of the system $\boldsymbol{M}_{a,t}^{\boldsymbol{F}}$. The collected data were time-aligned and split in two datasets of time-driven trajectories, namely relative Cartesian positions and manipulability. We trained a classical GMM over the time-driven relative positions and a geometry-aware GMM over the time-driven manipulability ellipsoids. The number of components of each model ($K = 4$) was selected by the experimenter.

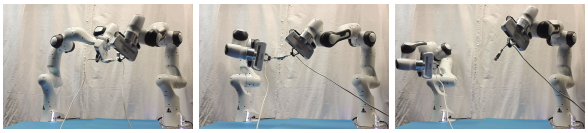
In the second part of the experiment, the unplugging task is reproduced by both the Baxter robot and a pair of Franka Emika Panda robots (see Fig. 19b, 19c). For both reproductions, the relative position between the end-effectors and the desired manipulability of the system were computed at each time step by a classical GMR as $\hat{\boldsymbol{\Delta x}}_t \sim p(\boldsymbol{\Delta x}|t)$ and a geometry-aware GMR as $\hat{\boldsymbol{M}}_{a,t}^{\boldsymbol{F}} \sim P(\boldsymbol{M}_a^{\boldsymbol{F}}|t)$. In both cases, the left robotic arm was required to move its joints to track the desired manipulability ellipsoid (38), while the right arm was required to maintain the desired relative Cartesian position with respect to the left arm, while tracking the desired manipulability as secondary objective (39). Note that the actuation contribution of each robot was taken into account to compute the manipulability ellipsoids through the whole experiment.

Figure 20a displays the two demonstrations recorded by kinesthetically guiding the Baxter robot along with the components of the GMM encoding $\boldsymbol{\Delta x}_t$ and the centers of the components of the geometry-aware GMM encoding $\boldsymbol{M}_a^{\boldsymbol{F}}$. The first and third dimensions of $\boldsymbol{\Delta x}_t$ are not represented as they do not vary significantly during the experiment. Figure 20b shows the relative Cartesian position and manipulability ellipsoid profile to be tracked and the reproduction results when the Baxter robot executed the task. Baxter successfully tracked the desired manipulability ellipsoid while maintaining the required relative distance between its end-effectors.

Figure 20c shows the relative Cartesian position between the arms and the manipulability ellipsoid profile obtained during the reproduction of the task by the two Panda robots. These successfully achieved the required task and tracked the desired manipulability ellipsoid profile obtained from model trained with the data recorded on the Baxter robot. Note the manipulability matching is not exact in this case due to the differences between Baxter and the Panda robots. Indeed,

(a) Demonstrations provided by the user on the Baxter robot

(b) Reproduction by the Baxter robot



(c) Reproduction by the two Franka Emika Panda robots

Figure 19. Unplugging task. The robots pose at the beginning of the task, before and after the extraction of the cable from the socket are respectively shown in the *left*, *middle* and *right* column.

even if the actuation capabilities of each robot are taken into account in our manipulability transfer framework, the capabilities of the two dual-arm system differ due to other physical specificities, e.g. the relative position of the bases of the arms.

7 Discussion

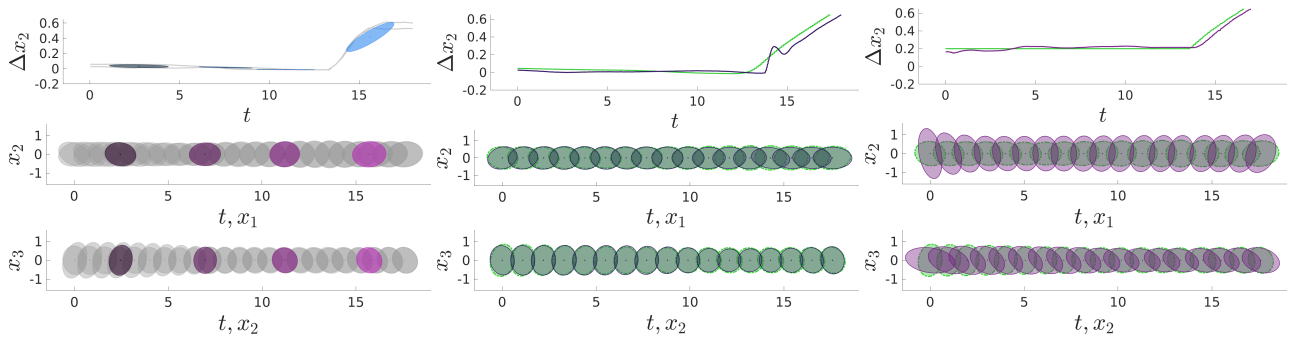
Our tracking formulation enables robots to modify their posture in an exponentially stable way so that desired manipulability ellipsoids are tracked, either as a main control task or as a redundancy resolution problem where the manipulability tracking is considered a secondary objective. Compared to state-of-the-art manipulability-based optimization schemes, our tracking formulation allows the reproduction of any manipulability ellipsoid beyond the maximization of manipulability parameters. The proposed tracking approach covers different manipulability ellipsoids proposed in the literature, such as velocity, force and dynamic manipulability ellipsoids (Doty et al. 1995). A relevant aspect about our approach is their generic structure, which means that we can track manipulability ellipsoids for a large variety of robots, as reported in the previous section, where a robotic hand, a Centauro robot, a humanoid and two different bimanual setups were used to test our tracking approach. This shows that our approach can be used in a large variety of contexts and that many further applications can be considered.

The manipulability transfer results reported in Section 6.3 showed the effectiveness of the proposed approach for transferring manipulability ellipsoids between robots that differ in their kinematic structure, which has remained a challenge in the robot learning community. Our learning 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. In addition, the proposed framework extends the robot learning capabilities beyond the transfer of trajectory, force and impedance.

It is important to emphasize the fact that the manipulability tracking precision strongly depends on the number of DoFs when the task is considered a secondary objective, as the higher it is, the more capable the robot is to perform more than one task simultaneously. Note that, in the case of legged robots (which are often characterized by a high number of DoFs), the manipulability tracking may still be slightly compromised because of the set of constraints imposed by the balancing task, as observed in Section 6.2. However, if these robots are provided with the possibility of modifying their feet position while keeping balance, then the manipulability tracking may be further improved. This clearly requires more sophisticated balancing controllers, but gives robots more freedom to adapt their posture and achieve better manipulability tracking. Notice that in the case of robotic hands, a similar behavior arises when the finger tips are constrained according to some grasping requirements, which might affect the manipulability tracking when projected into the nullspace of the primary task.

It is important to notice that the proposed manipulability tracking approach is a local method in the sense that the solution depends on the current configuration of the robot expressed through the Jacobian. This makes the tracking convergence dependent on the current configuration of the robot, which sometimes may limit the tracking performance. However, the robot may achieve a better tracking performance if it is allowed to look for other initial postures. As an example, the robot may not track precisely the desired manipulability ellipsoids for a given initial posture, due for instance to its joint limits. However, if the robot slightly modifies its initial posture, it may find a better starting configuration to subsequently minimize the error between the desired and current manipulability ellipsoids in a larger proportion, even if the new initial posture initially increases this error.

Overall, the proposed manipulability transfer framework may be exploited in a large variety of applications, where the posture of the robot may have an impact on its performance while executing the task. In addition to varying the robot posture for task compatibility, tracking a desired manipulability profile as a secondary task may typically complement a main control task to avoid singularity, handle perturbations during task execution, optimize the execution time or minimize the energy consumption (Kim et al. 2010). In particular, manipulability transfer may be utilized from a motion planning point of view. To do so, the robot may first track a desired manipulability as main control task in a planning phase, where the robot adapts its posture in order to anticipate the next action. Following this planning phase, the robot executes the desired action with a posture adapted to the task requirements. In this phase, the desired manipulability is tracked as a secondary task. Moreover, in the context of rehabilitation and assistance, the proposed learning and tracking formulations may be exploited in control strategies for exoskeletons. In (Petrič et al. 2019), the exoskeleton posture is optimized to achieve an isotropic manipulability by sensing the human muscular manipulability. In this setting, a varying



(a) Demonstrations and GMM **(b)** Reproduction with Baxter **(c)** Reproduction with the Panda robots

Figure 20. (*a*) Demonstrations and GMM encoding the unplugging task. The *top* graph shows the demonstrated relative end-effectors position for the Baxter robot (in gray) and components of the 4-states GMM (in blue). Only the most representative dimension is displayed. The distance between the two arms increases when the cable is unplugged from the socket. The *middle-bottom* graphs show the demonstrated force manipulability profile (in gray) and centers of the 4-states GMM in the SPD manifold over time (in purple). **(*b*) Reproduction of the unplugging task with Baxter.** The desired and reproduced trajectories are represented in green and dark blue respectively. The *top* graph shows the desired and reproduced relative position between the end-effectors along the second dimension. The *middle-bottom* graphs show the desired and reproduced (overlapping) manipulability ellipsoids. **(*c*) Reproduction of the unplugging task with the two Panda robots.** The desired and reproduced trajectories are represented in green and purple respectively. The *top* graph shows the desired and reproduced relative position between the end-effectors along the second dimension. The *middle-bottom* graphs shows the desired and reproduced manipulability ellipsoids. The position x and time t are given in meters and seconds.

exoskeleton manipulability profile may be retrieved using GMR as a function of the sensed muscular manipulability.

From a mathematical point of view, it is worth highlighting the importance of considering the structure of the data we work with. While alternative solutions to handle SPD matrices are present in literature (e.g. those using Cholesky decomposition), we showed that Euclidean manipulability-tracking controllers lead to unstable behaviors in contrast to the stable behavior displayed by our geometry-aware controller. Equally important, the manipulability ellipsoids profiles retrieved by the geometry-aware and Euclidean GMR were similar only around the mean of the GMM components, but diverged when moving away from it. This is because the estimated output in Euclidean space is only a valid approximation for input data lying close to the mean, as reported in Section 5. Therefore, geometry-awareness is crucial for successful learning and tracking of manipulability ellipsoids.

8 Conclusions and Future Work

This article presented a novel framework for transferring manipulability ellipsoids to robots. The proposed approach is built on a probabilistic learning model that allows the encoding and retrieval manipulability ellipsoids, and on the extension of the classical inverse kinematics problem to manipulability ellipsoids, by establishing a mapping between a change of manipulability ellipsoid and the robot joint velocity. We exploited tensor representation and Riemannian manifolds to build a geometry-aware learning framework and exponentially stable tracking controllers and showed the importance of geometry-awareness for manipulability transfer. We then showed that our manipulability transfer framework allows the exploitation of task variations recovered by the learning approach to characterize the precision of the manipulability tracking problem. 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. Furthermore, it allows manipulability transfer between agents of different embodiments, while taking into account their individual characteristics and is adapted to complex scenarios involving any manipulability ellipsoid shape and various types of robots.

Future work will explore manipulability transfer between humans and robots. Following this research direction, we recently proposed a statistical analysis of single and dual-arm manipulability ellipsoids for human movements, accompanied by two human-to-robot manipulability transfer experiments. The corresponding results will be detailed in a forthcoming publication. We will also investigate manipulability transfer strategies where the desired manipulability would be optimized in function of the robot. The objective would be to adapt the manipulability ellipsoid to exploit the capabilities of the learner in situations in which this learner can reach a better manipulability than the teacher for the task at hand.

Acknowledgements

This work was supported by the Swiss National Science Foundation (SNSF/DFG project TACT-HAND).

References

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

Azad M, Babič J and Mistry M (2017) Dynamic manipulability of the center of mass: A tool to study, analyse and measure physical ability of robots. In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. pp. 3484–3490.

Baccelliere L, Kashiri N, Muratore L, Laurenzi A, Kamedula M, Margan A, Cordasco S, Malzahn J and Tsagarakis NG (2017) Development of a human size and strength compliant bi-manual platform for realistic heavy manipulation tasks. In:

IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS). Vancouver, Canada, pp. 5594–5601.

Basser P and Pajevic S (2007) Spectral decomposition of a 4th-order covariance tensor: Applications to diffusion tensor MRI. *Signal Processing* 87(2): 220–236.

Bruyninckx H and De Schutter J (1996) Symbolic differentiation of the velocity mapping for a serial kinematic chain. *Mechanism and Machine Theory* 31(2): 135–148.

Chiacchio P (1990) Exploiting redundancy in minimum-time path following robot control. In: *American Control Conference*. pp. 2313–2318.

Chiacchio P, BouffardVercelli Y and Pierrot F (1997) Force polytope and force ellipsoid for redundant manipulators. *Journal of Robotic Systems* 14(8): 613–620.

Chiacchio P, Chiaverini S, Sciavicco L and Siciliano B (1991a) Global Task Space Manipulability Ellipsoids for Multiple-Arm Systems. *IEEE Trans. on Robotics and Automation* 7(5): 678–685.

Chiacchio P, Chiaverini S, Sciavicco L and Siciliano B (1991b) Reformulation of dynamic manipulability ellipsoid for robotic manipulators. In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. pp. 2192–2197.

Chiu SL (1987) Control of redundant manipulators for task compatibility. In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. pp. 1718–1724.

Chiu SL (1988) Task compatibility of manipulator postures. *Intl. Journal of Robotics Research* 7(5): 13–21.

Cos I, Bélanger N and Cisek P (2011) The influence of predicted arm biomechanics on decision making. *Journal of Neurophysiology* 105: 3022–3033.

Delhaisse B, Rozo L and Caldwell DG (2019) Pyrobolearn: A Python framework for robot learning practitioners. In: *Conference on Robot Learning (CoRL)*.

do Carmo M (1992) *Riemannian Geometry*. Birkhäuser Basel.

Doty K, Schwartz E, Melchiorri C and Bonivento C (1995) Robot manipulability. *IEEE Transactions on Robotics and Automation* 11(3): 462–468.

Dounskaia N, Wang W, Sainburg RL and Przybyla A (2014) Preferred directions of arm movements are independent of visual perception of spatial directions. *Experimental Brain Research* 232(2): 575–586.

Dubbelman G (2011) *Intrinsic Statistical Techniques for Robust Pose Estimation*. PhD thesis, University of Amsterdam, Netherlands.

Freifeld O, Hauberg S and Black MJ (2014) Model transport: Towards scalable transfer learning on manifolds. In: *Conf. on Computer Vision and Pattern Recognition (CVPR)*. Columbus, OH, USA, pp. 1378–1385.

Gu Y, Lee G and Yao B (2015) Feasible center of mass dynamic manipulability of humanoid robots. In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. pp. 5082–5087.

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

Jaquier N and Calinon S (2017) 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, pp. 59–64.

Jaquier N, Rozo L, Caldwell DG and Calinon S (2018) Geometry-aware tracking of manipulability ellipsoids. In: *Robotics: Science and Systems (R:SS)*. Pittsburgh, USA.

Kim W, Lee S, Kang M, Han J and Han C (2010) Energy-efficient gait pattern generation of the powered robotic exoskeleton using DME. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. pp. 2475–2480.

Kolda T and Bader B (2009) Tensor decompositions and applications. *SIAM Review* 51(3): 455–500.

Lee I and Oh J (2016) Humanoid posture selection for reaching motion and a cooperative balancing controller. *Journal of Intelligent and Robotics Systems* 8(3-4): 301–316.

Lee J (1997) A Study on the Manipulability Measures for Robot manipulators. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. pp. 1458–1465.

Lee J (2012) *Introduction to Smooth Manifolds*, *Graduate Texts in Mathematics*, volume 218. Springer, New York.

Lee S (1989) Dual redundant arm configuration optimization with task-oriented dual arm manipulability. *IEEE Transactions on Robotics and Automation* 5(1): 78–97.

Luh JYS and Zheng YF (1987) Constrained Relations Between Two Coordinated Industrial Robots for Motion Control. *Intl. Journal of Robotics Research* 6(3): 60–70.

Mistry M, Nakanishi J, Cheng G and Schaal S (2008) Inverse kinematics with floating base and constraints for full body humanoid robot control. In: *IEEE/RAS Intl. Conf. on Humanoid Robots (Humanoids)*. Daejeon, South Korea, pp. 22–27.

Morasso P (1981) Spatial control of arm movements. *Experimental Brain Research* 42: 223–227.

Murray RM, Li Z and Shankar Sastry S (1994) *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc.

Pait F and Colón D (2010) Some properties of the Riemannian distance function and the position vector X, with applications to the construction of Lyapunov functions. In: *IEEE Conference on Decision and Control (CDC)*. Atlanta, USA, pp. 6277–6280.

Park FC (1995) Optimal Robot Design and Differential Geometry. *Journal of Mechanical Design* 117(B): 87–92.

Park FC and Kim JW (1998) Manipulability of closed kinematic chains. *ASME Journal of Mechanical Design* 120(4): 542–548.

Pennec X, Fillard P and Ayache N (2006) A Riemannian framework for tensor computing. *Intl. Journal on Computer Vision* 66(1): 41–66.

Petrič T, Peternel L, Morimoto J and Babič J (2019) Assistive arm-exoskeleton control based on human muscular manipulability. *Frontiers in Neurorobotics* 13.

Prattichizzo D, Malvezzi M, Gabiccini M and Bicchi A (2012) On the Manipulability Ellipsoids of Underactuated Robotic Hands with Compliance. *Robotics and Autonomous Systems* 60(3): 337–346.

Ratliff N, Toussaint M and Schaal S (2015) Understanding the geometry of workspace obstacles in motion optimization. In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. pp. 4202–4209.

Rozo L, Calinon S, Caldwell DG, Jiménez P and Torras C (2016) Learning physical collaborative robot behaviors from human demonstrations. *IEEE Trans. on Robotics* 32(3): 513–527.

Rozo L, Jaquier N, Calinon S and Caldwell DG (2017) Learning manipulability ellipsoids for task compatibility in robot manipulation. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. pp. 3183–3189.

Sabes P and Jordan M (1997) Obstacle avoidance and a perturbation sensitivity model for motor planning. *Journal of Neuroscience* 17: 7119–7128.

Simo-Serra E, Torras C and Moreno-Noguer F (2017) 3D human pose tracking priors using geodesic mixture models. *Intl. Journal on Computer Vision* 122(2): 388–408.

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

Sra S (2012) A new metric on the manifold of kernel matrices with application to matrix geometric means. *Neural Information Processing Systems* : 144–152.

Sra S and Hosseini R (2015) Conic geometric optimization on the manifold of positive definite matrices. *SIAM Journal on Optimization* 25(1): 713–739.

Tyagi A and Davis JW (2008) A recursive filter for linear systems on Riemannian manifolds. In: *Conf. on Computer Vision and Pattern Recognition (CVPR)*. Anchorage, Alaska, pp. 1–8.

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

Wu D (2020) Intrinsic construction of Lyapunov functions on Riemannian manifold. *arXiv preprint 2002.11384* .

Yoshikawa T (1985a) Dynamic manipulability of robotic manipulators. *Journal of Robotic Systems* 2: 113–124.

Yoshikawa T (1985b) Manipulability of robotic mechanisms. *Intl. Journal of Robotics Research* 4(2): 3–9.

Zeestraten MJA, Havoutis I, Silvério J, Calinon S and Caldwell DG (2017) An approach for imitation learning on Riemannian manifolds. *IEEE Robotics and Automation Letters* 2(3): 1240–1247.

Appendices

A Derivative of a matrix w.r.t. a vector

Left multiplication by a constant matrix (Eq. (14))

$$\frac{\partial \boldsymbol{A}\boldsymbol{Y}}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{x}} \times_1 \boldsymbol{A}$$

Proof.

$$\left(\frac{\partial \boldsymbol{A}\boldsymbol{Y}}{\partial \boldsymbol{x}}\right)_{ljk} = \frac{\partial}{\partial x_k}\sum_i a_{li}y_{ij} = \sum_i a_{li}\frac{\partial y_{ij}}{\partial x_k}$$

Right multiplication by a constant matrix (Eq. (15))

$$\frac{\partial \boldsymbol{Y}\boldsymbol{B}}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{x}} \times_2 \boldsymbol{B}^{\mathsf{T}}$$

Proof.

$$\left(\frac{\partial \boldsymbol{Y}\boldsymbol{B}}{\partial \boldsymbol{x}}\right)_{ilk} = \frac{\partial}{\partial x_k}\sum_i y_{ij}b_{jl} = \sum_j b_{jl}\frac{\partial y_{ij}}{\partial x_k}$$

Derivative of the inverse of a matrix (Eq. (16))

$$\frac{\partial \boldsymbol{Y}^{-1}}{\partial \boldsymbol{x}} = -\frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{x}}^{\mathsf{T}} \times_1 \boldsymbol{Y}^{-1} \times_2 \boldsymbol{Y}^{-\mathsf{T}}$$

Proof. We compute the derivative of the definition of the inverse $\boldsymbol{Y}^{-1}\boldsymbol{Y} = \boldsymbol{I}$ as

$$\frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{Y}^{-1}\boldsymbol{Y}) = \frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{I}),$$

$$\frac{\partial \boldsymbol{Y}^{-1}}{\partial \boldsymbol{x}} \times_2 \boldsymbol{Y}^{\mathsf{T}} + \frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{x}} \times_1 \boldsymbol{Y}^{-1} = \boldsymbol{0}.$$

Then, by isolating $\frac{\partial \boldsymbol{Y}^{-1}}{\partial \boldsymbol{x}}$, we obtain

$$\frac{\partial \boldsymbol{Y}^{-1}}{\partial \boldsymbol{x}} = -\frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{x}}^{\mathsf{T}} \times_1 \boldsymbol{Y}^{-1} \times_2 \boldsymbol{Y}^{-\mathsf{T}}.$$

B Symbolic manipulability Jacobian for a serial kinematic chain

The computation of the manipulability Jacobian involves computing the derivative of the robot Jacobian w.r.t. the joint angles. Those derivatives can be computed in a symbolic form as shown in (Bruyninckx and De Schutter 1996). We remind here the symbolic derivative for the hybrid representation of the Jacobian $\boldsymbol{J} \in \mathbb{R}^{6\times n}$ that is used in the computation of the manipulability Jacobian $\boldsymbol{\mathcal{J}}$.

The i-th column of the Jacobian is denoted by

$$\boldsymbol{J}^i = \begin{pmatrix} \boldsymbol{w}^i \\ \boldsymbol{v}^i \end{pmatrix}, \tag{67}$$

with $\boldsymbol{w}^i \in \mathbb{R}^3$ and $\boldsymbol{v}^i \in \mathbb{R}^3$ the rotational and translational components of the Jacobian.

The derivative of the Jacobian w.r.t. the joint angles is a third order tensor $\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{q}} \in \mathbb{R}^{6\times n\times n}$ with mode-1 fibers or columns

$$\left(\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{q}}\right)_{:ij} = \frac{\partial \boldsymbol{J}^i}{\partial q^j} = \begin{cases} \boldsymbol{P}_\Delta(\boldsymbol{J}^j)\boldsymbol{J}^i & \text{if } j \leq i \\ -\boldsymbol{M}_\Delta(\boldsymbol{J}^j)\boldsymbol{J}^i & \text{if } j > i \end{cases}, \tag{68}$$

where

$$\boldsymbol{P}_\Delta(\boldsymbol{J}^j) = \begin{pmatrix} [\boldsymbol{w}^j\times] & \boldsymbol{0}_{3\times 3} \\ \boldsymbol{0}_{3\times 3} & [\boldsymbol{w}^j\times] \end{pmatrix}, \tag{69}$$

$$\boldsymbol{M}_\Delta(\boldsymbol{J}^j) = \begin{pmatrix} \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 3} \\ [\boldsymbol{v}^j\times] & \boldsymbol{0}_{3\times 3} \end{pmatrix}, \tag{70}$$

and $\times$ the cross product between two vectors. The notation $[\boldsymbol{w}^j\times]$ in a matrix denotes that the corresponding component of the result of the right-multiplication of the matrix by a vector is equal to the cross product between $\boldsymbol{w}^j$ and the corresponding vector component, e.g. $\boldsymbol{P}_\Delta(\boldsymbol{J}^j)\boldsymbol{J}^i = \begin{pmatrix} \boldsymbol{w}^j \times \boldsymbol{w}^i \\ \boldsymbol{w}^j \times \boldsymbol{v}^i \end{pmatrix}$.

Note that the time derivative of the Jacobian can therefore be computed as

$$\frac{d\boldsymbol{J}}{dt} = \sum_{j=1}^{n} \frac{\partial \boldsymbol{J}}{\partial q^j}\dot{q}_j. \tag{71}$$

C Symbolic dynamic manipulability Jacobian for a serial kinematic chain

The derivative of the robot inertia matrix w.r.t. joint angles is necessary for the computation of the dynamic manipulability Jacobian. It can be computed in closed form as follows.

The inertia matrix $\boldsymbol{\Lambda}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ can be written as

$$\boldsymbol{\Lambda}(\boldsymbol{q}) = \sum_{i=1}^{n} \boldsymbol{J}_i^{\mathsf{T}} \begin{pmatrix} \boldsymbol{\Lambda}_i & \boldsymbol{0} \\ \boldsymbol{0} & m_i \boldsymbol{I} \end{pmatrix} \boldsymbol{J}_i, \tag{72}$$

where $\boldsymbol{J}_i$, $\boldsymbol{\Lambda}_i$ and m_i are the Jacobian, inertia matrix and mass of link i, respectively (Park 1995; Murray et al. 1994).

The derivative of the inertia matrix is the third order tensor $\frac{\partial \boldsymbol{\Lambda}}{\partial \boldsymbol{q}} \in \mathbb{R}^{n\times n\times n}$ computed as

$$\frac{\partial \boldsymbol{\Lambda}}{\partial \boldsymbol{q}} = \sum_{i=1}^{n} \frac{\partial \boldsymbol{J}_i^{\mathsf{T}}}{\partial \boldsymbol{q}} \times_2 \boldsymbol{J}_i^{\mathsf{T}} \boldsymbol{M}_i + \frac{\partial \boldsymbol{J}_i}{\partial \boldsymbol{q}} \times_1 \boldsymbol{J}_i^{\mathsf{T}} \boldsymbol{M}_i, \tag{73}$$

where $\boldsymbol{M}_i = \begin{pmatrix} \boldsymbol{\Lambda}_i & \boldsymbol{0} \\ \boldsymbol{0} & m_i \boldsymbol{I} \end{pmatrix}$ and $\frac{\partial \boldsymbol{J}_i}{\partial \boldsymbol{q}}$ is computed with Eq. (68).

D Symbolic derivative of the manipulability Jacobian for a serial kinematic chain

In some cases, e.g. in the acceleration tracking controller, the time derivative of the manipulability Jacobian is required. This time derivative can be computed symbolically by exploiting the first and second derivative of the Jacobian w.r.t. the joint angles.

The time derivative of the velocity manipulability Jacobian $\boldsymbol{\mathcal{J}}^{\dot{\boldsymbol{x}}} \in \mathbb{R}^{6\times n\times n}$ defined as

$$\boldsymbol{\mathcal{J}}^{\dot{\boldsymbol{x}}} = \frac{\partial \boldsymbol{J}}{\partial \boldsymbol{q}} \times_2 \boldsymbol{J} + \frac{\partial \boldsymbol{J}^{\mathsf{T}}}{\partial \boldsymbol{q}} \times_1 \boldsymbol{J}, \tag{74}$$

is obtained by exploiting the chain rule as

$$\frac{\partial \boldsymbol{\mathcal{J}}^{\dot{\boldsymbol{x}}}}{\partial t} = \frac{\partial}{\partial t}\left(\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{q}} \times_2 \boldsymbol{J} + \frac{\partial \boldsymbol{J}^{\mathsf{T}}}{\partial \boldsymbol{q}} \times_1 \boldsymbol{J} \right) \tag{75}$$

$$= \frac{\partial^2 \boldsymbol{J}}{\partial t \partial \boldsymbol{q}} \times_2 \boldsymbol{J} + \frac{\partial \boldsymbol{J}}{\partial \boldsymbol{q}} \times_2 \frac{\partial \boldsymbol{J}}{\partial t} + \frac{\partial^2 \boldsymbol{J}^{\mathsf{T}}}{\partial t \partial \boldsymbol{q}} \times_1 \boldsymbol{J} + \frac{\partial \boldsymbol{J}^{\mathsf{T}}}{\partial \boldsymbol{q}} \times_1 \frac{\partial \boldsymbol{J}}{\partial t}. \tag{76}$$

The time derivative of the Jacobian is given by Eq. (71) and the time derivative of the derivative of the Jacobian w.r.t. joint angles is given by

$$\frac{\partial^2 \boldsymbol{J}}{\partial t \partial q_j} = \sum_{k=1}^{n} \frac{\partial^2 \boldsymbol{J}}{\partial q_k \partial q_j} \dot{q}_k, \tag{77}$$

where the second derivative of the Jacobian w.r.t. the joint angles is a fourth order tensor $\frac{\partial^2 \boldsymbol{J}}{\partial \boldsymbol{q}^2} \in \mathbb{R}^{6\times n\times n\times n}$ with mode-1 fibers or columns

$$\left(\frac{\partial^2 \boldsymbol{J}}{\partial \boldsymbol{q}^2}\right)_{:ijk} = \frac{\partial^2 \boldsymbol{J}^i}{\partial q^k \partial q^j} =$$

$$\begin{cases} \big(\boldsymbol{P}_\triangle(\boldsymbol{J}^j)\boldsymbol{P}_\triangle(\boldsymbol{J}^k)\big)\boldsymbol{J}^i + \boldsymbol{P}_\triangle(\boldsymbol{J}^j)\big(\boldsymbol{P}_\triangle(\boldsymbol{J}^k)\boldsymbol{J}^i\big) & \text{if } k \le j \le i \\ \boldsymbol{P}_\triangle(\boldsymbol{J}^j)\big(\boldsymbol{P}_\triangle(\boldsymbol{J}^k)\boldsymbol{J}^i\big) & \text{if } j \le k \le i \\ -\boldsymbol{P}_\triangle(\boldsymbol{J}^j)\big(\boldsymbol{M}_\triangle(\boldsymbol{J}^k)\boldsymbol{J}^i\big) & \text{if } j \le i < k \\ -\big(\boldsymbol{P}_\triangle(\boldsymbol{J}^k)\boldsymbol{M}_\triangle(\boldsymbol{J}^j)\big)\boldsymbol{J}^i - \boldsymbol{M}_\triangle(\boldsymbol{J}^j)\big(\boldsymbol{P}_\triangle(\boldsymbol{J}^k)\boldsymbol{J}^i\big) & \text{if } k \le i < j \\ -\big(\boldsymbol{P}_\triangle(\boldsymbol{J}^k)\boldsymbol{M}_\triangle(\boldsymbol{J}^j)\big)\boldsymbol{J}^i & \text{if } i < k < j \\ -\big(\boldsymbol{P}_\triangle(\boldsymbol{J}^j)\boldsymbol{M}_\triangle(\boldsymbol{J}^k)\big)\boldsymbol{J}^i & \text{if } i < j \le k \end{cases} \tag{78}$$

where $\boldsymbol{P}_\triangle(\boldsymbol{J}^j)$ and $\boldsymbol{M}_\triangle(\boldsymbol{J}^k)$ are defined as in (69) and (70), respectively. The time derivative of the force manipulability Jacobian $\boldsymbol{\mathcal{J}}^{\boldsymbol{F}}$ and the manipulability Jacobian $\boldsymbol{\mathcal{J}}^{\ddot{\boldsymbol{x}}}$ corresponding to the dynamic manipulability ellipsoid can be computed symbolically in a similar way using Eqs. (71) and (77). Moreover, their w.r.t. joint angles can be computed symbolically using the chain rules, Eqs. (68) and (78).