Geometry-aware Tracking of Manipulability Ellipsoids

Noémie Jaquier*†, Leonel Rozo*‡, Darwin G. Caldwell‡ and Sylvain Calinon†‡
*Both authors contributed equally to this work.
†Idiap Research Institute, Rue Marconi 19, PO Box 592, CH-1920 Martigny, Switzerland.
‡Department of Advanced Robotics, Istituto Italiano di Tecnologia (IIT), Via Morego 30, 16163 Genova, Italy.
Email: name.surname@idiap.ch, name.surname@iit.it.

Abstract—Body posture can greatly influence human performance when carrying out manipulation tasks. Adopting an appropriate pose helps us regulate our motion and strengthen our capability to achieve a given task. This effect is also observed in robotic manipulation where the robot joint configuration affects not only the ability to move freely in all directions in the workspace, but also the capability to generate forces along different axes. In this context, manipulability ellipsoids arise as a powerful descriptor to analyze, control and design the robot dexterity as a function of the articulatory joint configuration. This paper presents a new tracking control scheme in which the robot is requested to follow a desired profile of manipulability ellipsoids, either as its main task or as a secondary objective. The proposed formulation exploits tensor-based representations and takes into account that manipulability ellipsoids lie on the manifold of symmetric positive definite matrices. The proposed mathematical development is compatible with statistical methods providing 4th-order covariances, which are here exploited to reflect the tracking precision required by the task. Extensive evaluations in simulation and two experiments with a real redundant manipulator validate the feasibility of the approach, and show that this control formulation outperforms previously proposed approaches.

I. INTRODUCTION

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 [13], 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 [5]. For example, the human central nervous system plans arm movements considering its directional sensitivity, which is directly related to the arm posture [18]. 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 [6].

Robotics researchers have also explored 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 [4]. In this context, the so-called manipulability ellipsoid [23] 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 [4], and to improve minimum-time trajectory planning using a manipulability-aware inverse kinematics algorithm [2]. Vahrenkamp *et al.* [21] 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 [8], and task-level robot programming frameworks [19], to obtain singularity-free joint trajectories and high task-space dexterity. Nevertheless, as stated in [12], solely maximizing the ellipsoid volume to achieve high dexterity in motion may cause a reverse effect on the flexibility in force.

The aforementioned approaches do not specify a desired robot manipulability for the task. In contrast, Lee *et al.* [11] proposed an optimization method for finding 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 [12]. Note that despite authors in [11] and [12] predetermined task-dependent robot manipulability, their 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. For instance, Rozo *et al.* [17] showed that this geometric property improved a gradient-based redundancy resolution method aimed at varying the robot posture to match a desired manipulability ellipsoid as a secondary objective of a trajectory tracking task.

In this paper we address the problem of tracking robot manipulability ellipsoids from a novel geometry-aware control perspective. The proposed manipulability tracking formulation is 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 III-A. This relationship demands to consider that manipulability ellipsoids lie on the manifold of SPD matrices, which is here tackled by exploiting tensor-based representations and differential geometry (see Section II). This mathematical development is compatible with statistical methods providing 4th-order covariances [17], which are here exploited to reflect the manipulability tracking precision required by the task, as shown in Section III-B. This also allows our formulation to be easily combined with manipulability transfer frameworks, where desired manipulability ellipsoid profiles are obtained from demonstrations of a specific task performed by either a human or a robot.

Note that we focus on kineto-static manipulability ellipsoids, namely those obtained from differential kinematics and statics relationships (using the duality principle). However, the approach can be straightforwardly applied to other types of manipulability measures such as those proposed in [1, 3, 7]. This opens the door to manipulability tracking problems where different task requirements at kinematics and dynamics levels are needed, which may determine time-varying optimal directions for controlling a robot (e.g., velocity, force, acceleration) to perform successfully. We show that our approach outperforms previous gradient-based approaches [17] in terms of iterations needed to converge to a desired manipulability. The proposed controller is showcased in a pushing task and a peg-in-hole assembly using a 7-DoF robot (see Section IV).

II. BACKGROUND

A. Manipulability ellipsoids

Velocity and force manipulability ellipsoids introduced in [23] are kinetostatic performance measures of robotic platforms. They indicate the preferred directions in which force or velocity control commands may be performed at a given joint configuration. More specifically, the velocity manipulability ellipsoid describes the characteristics of feasible motion in Cartesian space corresponding to all the unit norm joint velocities. The velocity manipulability of an n-DoF robot can be found by using the kinematic relationship between task velocities $\dot{x}$ and joint velocities $\dot{q}$,

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

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

$$\|\dot{q}\|^2 = \dot{q}^\top \dot{q} = \dot{x}^\top (JJ^\top)^{-1} \dot{x}, \tag{2}$$

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

Note that the major axis of the velocity manipulability ellipsoid $M^{\dot{x}} = (JJ^\top)^{-1}$ 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 [4] for details).

Other forms of manipulability ellipsoids exist, such as the dynamic manipulability [22], 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 [3]. Recent works have extended this measure to analyze the robot capacity to accelerate its center of mass for locomotion stability [1, 7], showing the applicability of the aforementioned tools beyond robotic manipulation.

As mentioned previously, any manipulability ellipsoid 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 tracking formulation must consider this particular characteristic in order to properly track desired manipulability ellipsoids. We develop a geometry-aware formulation that takes inspiration from the classical inverse kinematics, by exploiting Riemannian manifolds and tensor representations, which are introduced next.

B. Riemannian manifold of $\mathcal{S}_{++}$

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 [15], and thus the use of classical Euclidean space methods for treating and analyzing these matrices is inadequate. A compelling solution is to endow these matrices with a Riemannian metric so that these form a Riemannian manifold.[3] This metric permits to define lengths of curves in the manifold. These curves, called geodesics, are the generalization of straight lines to Riemannian manifolds. Similarly to straight lines in Euclidean space, geodesics are the minimum length curves between two points on the manifold.

Intuitively, a Riemannian manifold $\mathcal{M}$ is a mathematical space for which each point locally resembles a Euclidean space. For each point $p \in \mathcal{M}$, there exists a tangent space $\mathcal{T}_p\mathcal{M}$ equipped with a positive definite inner product. In the case of the SPD manifold, the tangent space at any point $\Sigma \in \mathcal{S}_{++}^D$ is identified by the space of symmetric matrices Sym^D. The space of SPD matrices can be represented as the interior of a convex cone embedded in its tangent space Sym^D; see Fig. 1. To utilize these tangent spaces, we need mappings back and forth between $\mathcal{T}_p\mathcal{M}$ and $\mathcal{M}$, which are known as exponential and logarithmic maps.

The exponential map $\text{Exp}_{\Sigma} : \mathcal{T}_{\Sigma}\mathcal{M} \to \mathcal{M}$ maps a point L in the tangent space to a point Λ on the manifold, so that it lies

---

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

[2]Dually, the force manipulability ellipsoid can be computed from the static relationship between joint torques and Cartesian forces [23].

[3]The original cone of SPD matrices has been changed into a regular and complete (but curved) manifold with an infinite development in each of its $D(D+1)/2$ directions [15].
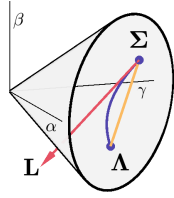
Fig. 1: Representation of the SPD manifold $\mathcal{S}_{++}^2$ embedded in its tangent space Sym^2. One point on the graph corresponds to a matrix $\begin{pmatrix} \alpha & \beta \\ \beta & \gamma \end{pmatrix} \in \text{Sym}^2$. Points inside the cone, such as $\boldsymbol{\Sigma}$ and $\boldsymbol{\Lambda}$, belong to the manifold. $\boldsymbol{L}$ 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 purple curve in the graph. Note that it does not correspond to the Euclidean path, depicted by the yellow line.

on the geodesic starting at $\boldsymbol{\Sigma}$ in the direction $\boldsymbol{L}$ and such that the distance between $\boldsymbol{\Sigma}$ and $\boldsymbol{\Lambda}$ is equal to the distance between $\boldsymbol{\Sigma}$ and $\boldsymbol{L}$. The inverse 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. 1.

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

$$d(\boldsymbol{\Lambda}, \boldsymbol{\Sigma}) = \| \log(\boldsymbol{\Sigma}^{-\frac{1}{2}} \boldsymbol{\Lambda} \boldsymbol{\Sigma}^{-\frac{1}{2}}) \|_{\text{F}}, \tag{3}$$

are computed as (see [15] for details)

$$\boldsymbol{\Lambda} = \text{Exp}_{\boldsymbol{\Sigma}}(\boldsymbol{L}) = \boldsymbol{\Sigma}^{\frac{1}{2}} \exp(\boldsymbol{\Sigma}^{-\frac{1}{2}} \boldsymbol{L} \boldsymbol{\Sigma}^{-\frac{1}{2}}) \boldsymbol{\Sigma}^{\frac{1}{2}}, \tag{4}$$

$$\boldsymbol{L} = \text{Log}_{\boldsymbol{\Sigma}}(\boldsymbol{\Lambda}) = \boldsymbol{\Sigma}^{\frac{1}{2}} \log(\boldsymbol{\Sigma}^{-\frac{1}{2}} \boldsymbol{\Lambda} \boldsymbol{\Sigma}^{-\frac{1}{2}}) \boldsymbol{\Sigma}^{\frac{1}{2}}. \tag{5}$$

In this paper, we exploit the Riemannian manifold framework to compute the difference between manipulability ellipsoids considering that these belong to the set $\mathcal{S}_{++}^D$. This geometry-aware approach proves to be crucial for tracking manipulability ellipsoids in terms of accuracy and convergence, beyond providing an appropriate mathematical treatment of the tracking problem. Note that Riemannian geometry is used in orientation tracking using unit quaternions [10] and has also been successfully exploited in applications such as robot motion optimization [16] and manipulability analysis of closed chains [14].

C. Tensor representation

Tensors are generalization of matrices to arrays of higher dimensions [9], where vectors and matrices may respectively be seen as 1st and 2nd-order tensors. Tensor representation permits to represent and exploit *a priori* data structure of multidimensional arrays. In this paper, such representation is mainly used 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. To do so, we first introduce the tensor operations needed for our mathematical treatment.

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

$$\mathcal{Y} = \mathcal{X} \times_n \boldsymbol{A} \iff \mathcal{Y}_{(n)} = \boldsymbol{A}\mathcal{X}_{(n)}, \tag{6}$$

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

2) 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 a vector variable $\boldsymbol{x}$ is a third order tensor $\frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{x}} \in \mathbb{R}^{I \times J \times K}$ such that

$$\left(\frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{x}}\right)_{ijk} = \frac{\partial y_{ij}}{\partial x_k}. \tag{7}$$

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

a) Left multiplication by a constant matrix:

$$\frac{\partial \boldsymbol{A}\boldsymbol{Y}}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{x}} \times_1 \boldsymbol{A} \tag{8}$$

Proof:

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

∎

b) Right multiplication by a constant matrix:

$$\frac{\partial \boldsymbol{Y}\boldsymbol{B}}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{x}} \times_2 \boldsymbol{B}^\top \tag{9}$$

Proof:

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

∎

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 third order tensor, namely

$$\frac{\partial \boldsymbol{Y}^{-1}}{\partial \boldsymbol{x}} = -\frac{\partial \boldsymbol{Y}^\top}{\partial \boldsymbol{x}} \times_1 \boldsymbol{Y}^{-1} \times_2 \boldsymbol{Y}^{-\top} \tag{10}$$

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

$$\frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{Y}^{-1}\boldsymbol{Y}) = \frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{I}) \implies \frac{\partial \boldsymbol{Y}^{-1}}{\partial \boldsymbol{x}} \times_2 \boldsymbol{Y}^\top + \frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{x}} \times_1 \boldsymbol{Y}^{-1} = \boldsymbol{0}.$$

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

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

∎

The proposed geometry-aware manipulability tracking, introduced in the next section, takes inspiration from the computation of the robot Jacobian, which is computed from the first-order time derivative of the robot forward kinematics. We use the tensor representation to similarly compute the first-order derivative of the function that describes the relationship between a manipulability ellipsoid $\boldsymbol{M}$ and the robot joint configuration $\boldsymbol{q}$.

III. Tracking Manipulability Ellipsoids

Several manipulation tasks in robotics may demand the robot to track a desired trajectory with a specific velocity profile, or apply forces along different task-related axes. These requirements are more easily achieved if the robot finds an appropriate posture that permits to apply the required velocity or force control commands. This problem can be viewed as matching a set of desired manipulability ellipsoids that are compatible with the task requirements, so that the robot can perform the task successfully. The desired manipulability ellipsoid profile can be manually set by an expert or alternatively learned from demonstration collected from either a human or a robot [17]. In this section, we introduce an approach that addresses this problem by exploiting the mathematical concepts presented in Section II.

A. Geometry-aware manipulability tracking formulation

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 desired robot joint values that lead the robot to match a desired manipulability ellipsoid.

First, the manipulability ellipsoid is expressed as a function of time as $M(t) = f\big(J(q(t))\big)$, for which we can compute the first-order time derivative by applying the chain rule as

$$\frac{\partial M(t)}{\partial t} = \frac{\partial f(J(q))}{\partial q} \times_3 \frac{\partial q(t)^\top}{\partial t} = \mathcal{J}(q) \times_3 \dot{q}^\top, \quad (11)$$

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

The derivation of the manipulability Jacobian $\mathcal{J}^F$ corresponding to the force manipulability ellipsoid $M^F = JJ^\top$ is straightforward by using (8) and (9) [4]

$$\mathcal{J}^F = \frac{\partial J}{\partial q} \times_2 J + \frac{\partial J^\top}{\partial q} \times_1 J. \quad (12)$$

Similarly, the manipulability Jacobian $\mathcal{J}^{\dot{x}}$ corresponding to the velocity manipulability ellipsoid $M^{\dot{x}} = (JJ^\top)^{-1}$ is obtained using (8), (9) and (10),

$$\mathcal{J}^{\dot{x}} = -\left(\frac{\partial J}{\partial q} \times_2 J + \frac{\partial J^\top}{\partial q} \times_1 J \right) \times_1 M^{\dot{x}} \times_2 M^{\dot{x}}. \quad (13)$$

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

[4] In the remainder of the paper we drop dependencies on q to simplify the notation.

use here a similar approach to compute the joint velocities $\dot{q}$ to track a desired manipulability profile. More specifically, by minimizing the L^2 norm of the residuals

$$\min_{\dot{q}} \|\dot{M} - \mathcal{J} \times_3 \dot{q}^\top\| = \min_{\dot{q}} \|\text{vec}(\dot{M}) - \mathcal{J}_{(3)}^\top \dot{q}\|,$$

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

$$\dot{q} = (\mathcal{J}_{(3)}^\dagger)^\top \text{vec}(\dot{M}), \quad (14)$$

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

Note that (14) 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 distance between the current manipulability ellipsoid M_t and the desired one $\hat{M}_t$ is minimum. Specifically, the difference between manipulability ellipsoids is computed using the logarithmic map (5) on the SPD manifold. Therefore, the corresponding controller is given by

$$\dot{q}_t = (\mathcal{J}_{(3)}^\dagger)^\top K_M \text{vec}\Big(\text{Log}_{M_t}(\hat{M}_t)\Big), \quad (15)$$

where K_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 manipulability ellipsoids profile is assigned a secondary role. Thus, the task objectives are to track the reference trajectories while exploiting the robot 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 null-space velocity that similarly exploits the geometry of the SPD manifold. Thus, the corresponding controller is given by

$$\dot{q}_t = J^\dagger K_x (\hat{x}_t - x_t)$$
$$+ (I - J^\dagger J)(\mathcal{J}_{(3)}^\dagger)^\top K_M \text{vec}\Big(\text{Log}_{M_t}(\hat{M}_t)\Big). \quad (16)$$

As in classical Jacobian-based methods, the use of the pseudo-inverse of the manipulability Jacobian in (14) typically requires to add a damping factor tuned as a function of the smallest singular values of $\mathcal{J}$, for numerical robustness. 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 M_t coincide with the desired one $\hat{M}$, 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. 2 shows how the manipulator configuration is successfully
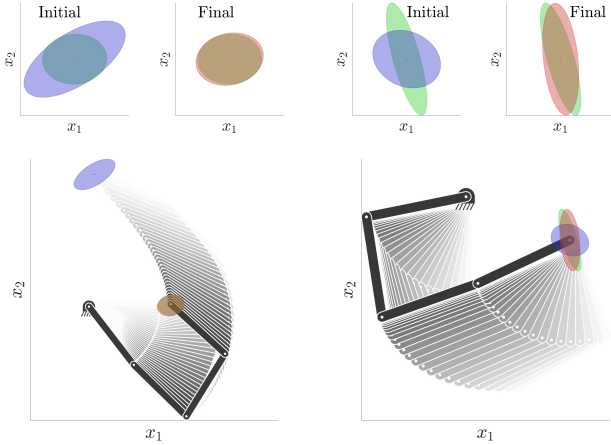
Fig. 2: Illustrations of the manipulability tracking as main task (*left*) and the manipulability-based redundancy resolution with Cartesian position control as main task (*right*). The robot color goes from light gray to black to show the evolution of the posture. Initial, final, and desired manipulability ellipsoids are respectively depicted in blue, red, and green. The *top* row shows close-up plots corresponding to the initial and final manipulability ellipsoids.

adjusted so that $M_t \simeq \hat{M}$ when the manipulability ellipsoid tracking is considered as the main task or as a secondary objective. These results show that the approach is suitable to solve these two manipulability ellipsoid tracking problems.

B. Exploiting 4th-order precision matrix as controller gain

An open problem regarding the proposed approach is how to specify the values of the gain matrix K_M, which basically determines how the manipulability tracking error affects the resulting joint velocities. In this sense, we propose to define 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. This encapsulates variability information of the task to be learned, and are represented as 4th-order tensors when the variability is computed over an SPD manifold [17]. 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.

As SPD matrices, or more broadly, any kind of matrix can be seen as 2nd-order tensor, the computation of covariance of matrices can be represented as a 4th-order covariance tensor. Thus, the covariance $\mathcal{S} \in \mathbb{R}^{D \times D \times D \times D}$ of N matrices $X_n \in \mathbb{R}^{D \times D}$ is defined as

$$\mathcal{S} = \frac{1}{N-1} \sum_{n=1}^{N} X_n \otimes X_n, \tag{17}$$

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

We therefore introduce the required precision $\mathcal{S}^{-1}$ for a given manipulability tracking task into the controllers defined in Section III-A. To do so, we define the gain matrix K_M as a function of the precision tensor. Specifically, in order to take

into account the variation of each component of the manipulability ellipsoid, we define K_M as diagonal components of the precision matrix $\mathcal{S}^{-1}$, with a proportion defined by

$$(K_M)_{ii} \propto (\mathcal{S}_{(1,2)}^{-1})_{ii}. \tag{18}$$

Alternatively, for cases in which the covariation of the manipulability ellipsoid components is relevant, we may define the controller gain matrix as a full SPD matrix, which is computed from the matricization of the precision tensor $\mathcal{S}^{-1}$ along its two first dimensions, with a proportion defined by

$$K_M \propto \mathcal{S}_{(1,2)}^{-1}. \tag{19}$$

To show how precision matrices work as controller gains in our manipulability tracking problem, we tested different forms of 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 K_M is diagonal as defined in (18). We tested four different precision tensors. In the first case, an 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 (18), 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 reduced. In this last case, the manipulability controller prioritizes the tracking of the ellipsoid orientation over the shape.

Figure 3 shows how the manipulator posture is adapted to match 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. 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 tries to match the whole manipulability. Note 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. 4. 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 formulation may be readily combined with manipulability learning frameworks such as [17].

IV. EXPERIMENTS

We first evaluate in simulation the proposed approach by comparing it to an Euclidean formulation. We then test the
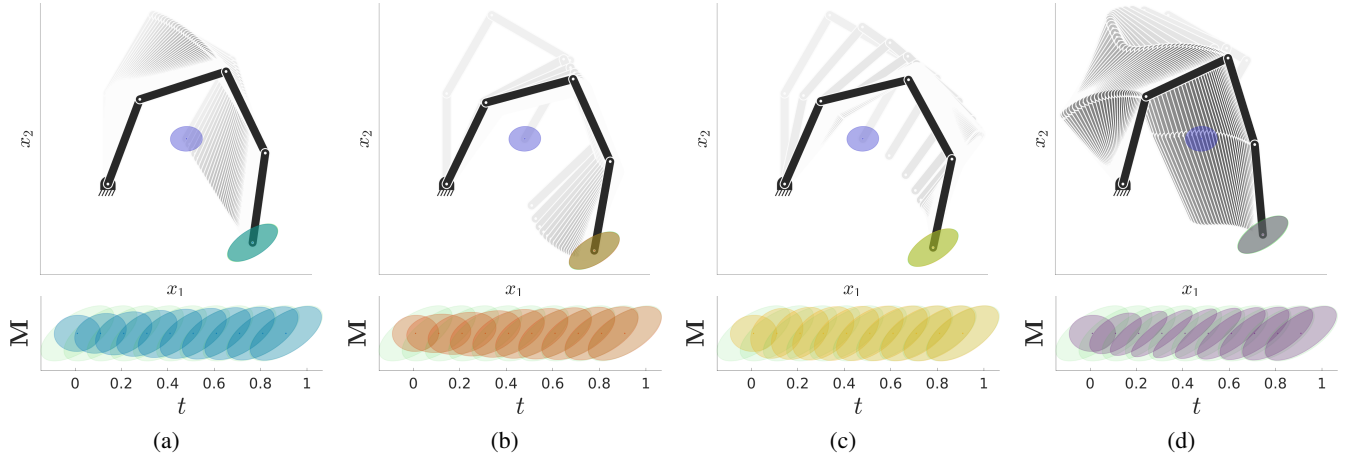
Fig. 3: Manipulability tracking as main task with diagonal gain matrices defined as a function of different precision tensors. Evolution of the robot configuration and corresponding manipulability ellipsoids are respectively shown in the top and bottom plots. (a): all components are given equal tracking precision. (b) and (c): tracking precision is higher for x_1 and x_2, respectively. The precision ratio between the prioritized and the rest of components of the gain matrix is $10:1$. (d): correlation between x_1 and x_2 axes is assigned a high tracking accuracy. The precision ratio between the prioritized correlation and the other components of the gain matrix is $3:1$. Initial and desired manipulability ellipsoids are depicted in dark blue and green on all graphs. Time t is given in seconds.
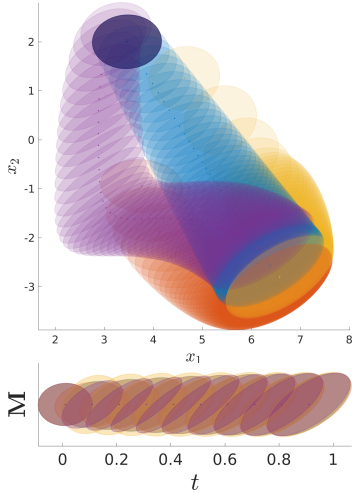

Fig. 4: Evolution of the robot manipulability and end-effector trajectory for different gain matrices when tracking a desired manipulability ellipsoid is the main task. *Top:* Trajectories of the end-effector for four different gain matrices along with the corresponding manipulability ellipsoids. The initial manipulability ellipsoid is depicted in dark blue. *Bottom:* Time evolution of the manipulability ellipsoids obtained with four different gain matrices. The gain matrices and manipulability ellipsoid colors correspond to those of Fig. 3. Position x and time t are given in centimeters and seconds, respectively.

approach with a Baxter robot controlled to track a desired manipulability ellipsoid as secondary objective.

A. Importance of geometry-awareness

The goal of this experiment is to evaluate the proposed tracking formulation compared to a controller ignoring the geometry of SPD matrices (i.e., treating the problem as Euclidean), and the gradient-based approach of [17]. To do so, we consider a planar 4-DoF robot that is required to keep its end-effector at a fixed Cartesian position $\hat{x}$ while minimizing

the distance between its current and desired manipulability ellipsoids M and $\hat{M}$. Thus, the manipulability tracking task becomes a secondary objective. The two following approaches are considered for comparison with the proposed formulation (16). Firstly, we analyze the corresponding Euclidean manipulability-tracking controller

$$\dot{q}_t = J^\dagger K_x(\hat{x}_t - x_t) + (I - J^\dagger J)(\mathcal{J}_{(3)}^\dagger)^\top K_M \text{vec}(\hat{M}_t - M_t),\tag{20}$$

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 evaluate the gradient-based approach of [17] that implements the controller

$$\dot{q}_t = J^\dagger K_x(\hat{x}_t - x_t) - (I - J^\dagger J)\alpha\nabla g_t(q),\tag{21}$$

where α is a scalar gain and

$$g_t(q) = \log\det\left(\frac{\hat{M}_t + M_t}{2}\right) - \frac{1}{2}\log\det\left(\hat{M}_t M_t\right)\tag{22}$$

is a cost function based on Stein divergence (a distance-like function on the SPD manifold [20]). The gain matrices K_M are fixed as identity matrices and the scalar gain is set to 1 for the comparison.

To alleviate the computational cost of the controllers using tensor representations, we defined matricization and vectorization operations using Mandel notation, such that

$$\mathcal{X}_{(3)} = \begin{pmatrix} \text{vec}\left(\boldsymbol{\mathcal{X}}_{:,:,1}\right)^\top \\ \vdots \\ \text{vec}\left(\boldsymbol{\mathcal{X}}_{:,:,K}\right)^\top \end{pmatrix} \text{ and } \text{vec}\left(\begin{pmatrix} \alpha & \beta \\ \beta & \gamma \end{pmatrix}\right) = \begin{pmatrix} \alpha \\ \gamma \\ \sqrt{2}\beta \end{pmatrix},\tag{23}$$

for $2\times2\times K$ third-order tensors and 2×2 matrices.
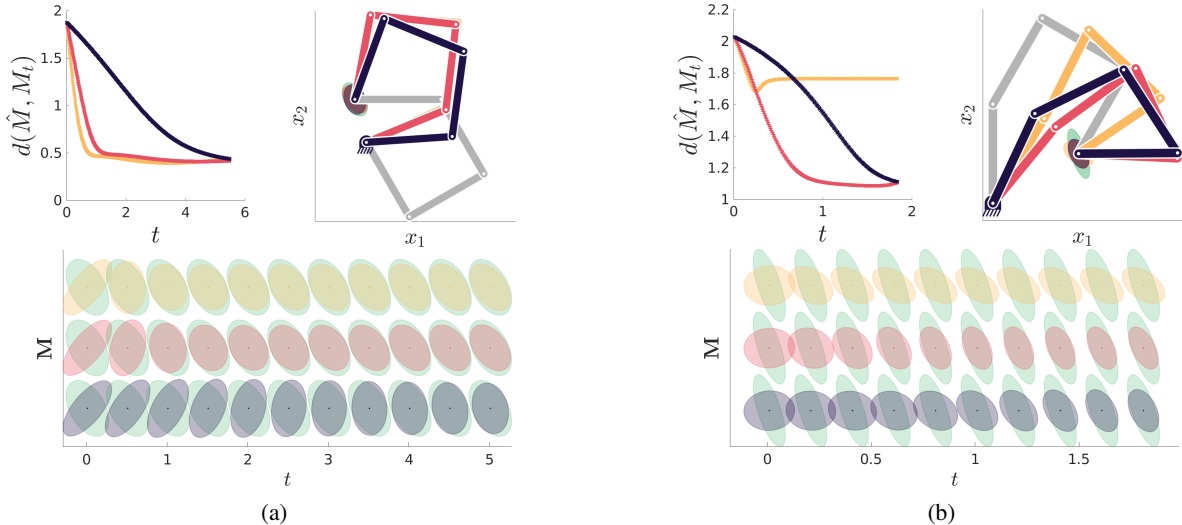
Fig. 5: Performance comparison of the different manipulability tracking formulations. Two cases are shown with varying initial robot configuration and desired manipulability. The *top left* graph shows the convergence of the affine invariant distance between the current and the desired manipulability ellipsoid over time. The distances for the Euclidean, geometry-aware and gradient-based approaches are respectively depicted in yellow, red, and purple. The *top 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. The *bottom row* shows the evolution of the manipulability ellipsoids over time for the different approaches.

Figure 5 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 geometry-aware and gradient-based approaches converge to a similar final robot configuration (see Fig. 5a, 5b-*top-right*), with similar values of the affine-invariant distance between the final and desired manipulability ellipsoids (see Fig. 5a, 5b-*top left*). 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 $\mathcal{J}(q)$.

Note that for some specific initial robot configurations and desired manipulability ellipsoids, the Euclidean manipulability-tracking controller (20) shows a slightly faster convergence rate than our method (see Fig. 5a). However, this Euclidean formulation leads to unstable behaviors in some configurations (see Fig. 5b), where the distance between the final and desired manipulability ellipsoids remains high compared to the other 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, the Euclidean manipulability-tracking controller is only effective if the current and desired ellipsoids are very similar.

The reported results not only showed that our approach outperforms the gradient-based method, but also supported our hypothesis that a geometry-aware manipulability controller results in good tracking performance while providing stable convergence regardless of the manipulability tracking error. 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 tracking precision required while matching 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.

B. Experiments with the Baxter robot

The performance of the proposed controller was tested in a pushing task and a peg-in-hole task (plugging an electric cable into an power socket), achieved by the 7-DoF arm of the Baxter robot. In the first experiment, the robot is required to match a desired manipulability ellipsoid aligned with a force that is perpendicularly applied to a wall, while the robot end-effector can freely move on the wall plane (see Fig. 6b). This task aims at emulating how humans vary their body posture when applying a force with known direction but unknown amplitude to successfully push an object. In this case, the robot controller is defined as (16), where the desired robot position $\hat{x}$ only considers the constraint of being on the wall plane, while $\hat{M}_t$ corresponds to a force manipulability whose main axis is orthogonal to the wall and kept constant over the course of the task.

Figure 6a shows the resulting manipulability using the redundancy resolution controller (16). As expected, the robot

modified its joint configuration in order to match, as accurately as possible, the desired force manipulability, therefore adopting a posture compatible with the force requirements of the pushing task. Note that the matching can in this experiment only be achieved partially, because the robot is also required to keep a position constraint at the level of its end-effector, and therefore the joints configuration to match the desired manipulability is restricted to avoid interference with the primary objective. In order to verify that the robot found an appropriate pose to match the desired manipulability while fulfilling the position constraint, we replaced the manipulability tracking term in (16) by joint velocity commands driven by Brownian noise. This test, carried out in simulation for ten minutes, allowed us to explore the space of possible poses satisfying the primary objective, for which corresponding manipulability ellipsoids were computed. The minimum distance between the computed and desired manipulability ellipsoids obtained in this test coincided with the distance achieved by our controller.

In the peg-in-hole scenario, the robot first needs to track a specific Cartesian trajectory to approach a hole and subsequently insert a peg into it, as shown in Fig. 6. Desired manipulability ellipsoids were defined according to the task requirements for the two different parts of the peg-in-hole process. Initially, a desired velocity manipulability is aligned with the direction of motion of the end-effector governed by the reference trajectory. Then, a desired force manipulability is set to be aligned with the force applied to insert the peg, which is executed at the end of the task. Note that both alignments refer to the major axis of the ellipsoids. Similarly as the pushing task, the robot used the redundancy resolution controller (16), where $\hat{x}$ was defined as the desired Cartesian trajectory to track, while $\hat{M}_t$ was set based on the aforementioned velocity and force manipulability ellipsoids required by the task.

In order to show the effects of the manipulability tracking controller on the robot posture over the course of the task, we also executed the peg-in-hole experiment with $K_M = 0$, which means that the manipulability tracking was fully disabled. A small difference between the robot postures can be observed during the approaching part, which shows how the manipulability controller influences the trajectory tracking phase. More notably, the robot significantly varied its posture when the insertion part took place, so that its manipulability coincided as accurately as possible with the desired force manipulability ellipsoid. This variation of the joint configuration consequently allows the robot to adopt a posture compatible with the control force required along the vertical direction of the task. Videos of the experiments accompany this paper (https://youtu.be/J4Ej4j6rhdY) and source codes are available at http://www.idiap.ch/software/pbdlib/[5].

V. CONCLUSIONS AND FUTURE WORK

This paper presented a novel approach to track robot manipulability ellipsoids. Our work extends the classical inverse
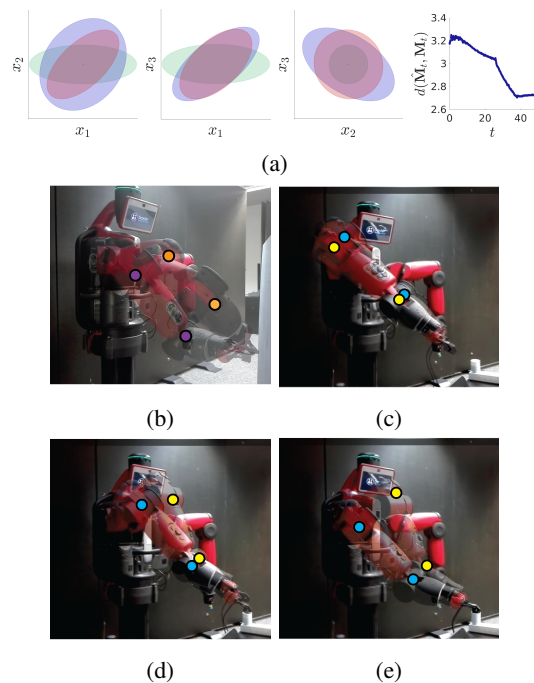
Fig. 6: (*a*) Pushing task: The *left* graphs show the initial, final, and desired manipulability ellipsoids respectively depicted in blue, red, and green. The *right* graph shows the evolution of the distance between the current and desired manipulability ellipsoids over time. (*b*) Pushing task: the initial and final pose of the robot are respectively depicted by purple and orange dots on the elbow and wrist bend joints. (*c-e*) Insertion task: the poses of the robot obtained with and without manipulability tracking are respectively depicted by yellow and blue dots on the elbow and wrist bend joints when the robot (*c*) approaches the hole, (*d*) prepares to insert the peg, (*e*) ends the insertion of the peg in the hole.

kinematics problem to manipulability ellipsoids, by establishing a mapping between a change of manipulability ellipsoid and the robot joint velocity. To do so, we exploited tensor representation and Riemannian manifolds to obtain a geometry-aware manipulability tracking controller. This enables the robot to modify its posture so that its manipulability ellipsoid matches a desired one, either as a main control task or as a redundancy resolution problem where the manipulability tracking is viewed as a secondary objective. We showed that the proposed formulation outperforms previous gradient-based approaches and provides a faster convergence rate. Furthermore, we showed that our approach is compatible with statistical methods providing 4th-order covariances, allowing us to exploit task variations to characterize the precision of the manipulability tracking problem, with stronger tracking along low variability directions. As future work, we plan to combine the proposed approach with learning from demonstration techniques, thus extending our work to manipulability transfer problems such as [17]. We will explore the use of our formulation in more complex tasks involving full 6D manipulability ellipsoids, and humanoid robot scenarios requiring to track a manipulability ellipsoid at the center of mass or zero-moment point [1, 7].

REFERENCES

[1] M. Azad, J. Babič, and M. Mistry. 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)*, pages 3484–3490, 2017.

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

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

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

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

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

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

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

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

[10] E. Kraft. A Quaternion-based Inscented Kalman Filter for Orientation Tracking. In *Proc. of the Intl. Conf. on Information Fusion*, pages 47–54, 2003.

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

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

[13] P. Morasso. Spatial control of arm movements. *Experimental Brain Research*, 42:223–227, 1981.

[14] F. C. Park and J. W. Kim. Manipulability of closed kinematic chains. *ASME Journal of Mechanical Design*, 120(4):542–548, 1998.

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

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

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

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

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

[20] S. Sra. A new metric on the manifold of kernel matrices with application to matrix geometric means. *Neural Information Processing Systems*, pages 144–152, 2012.

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

[22] T. Yoshikawa. Dynamic manipulability of robotic manipulators. *Journal of Robotic Systems*, 2:113–124, 1985.

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