Impedance Control on Arbitrary Surfaces for Ultrasound Scanning Using Discrete Differential Geometry

Michael Dyck, Arne Sachtler, Julian Klodmann, and Alin Albu-Schäffer, *Fellow, IEEE*

***Abstract*—We propose an approach to robotic ultrasound scanning and interaction control with arbitrary surfaces using a passivity-based impedance control scheme. First, we introduce task coordinates depending on the geometry of the surface, which enable hands-on guidance of the robot along the surface, as well as teleoperated and autonomous ultrasound image acquisition. Our coordinates allow controlling the signed distance of the robot to the surface and alignment of the tool to the surface normal using classical impedance control. This corresponds to implicitly obtaining a foliation of parallel surfaces. By setting the desired signed distance negative, i.e., into the surface, we obtain passive contact forces and simultaneously provide an intuitive way to control the maximum penetration depth into the surface. We extend the approach to also incorporate coordinates allowing to control the specific point on the surface and, automatically, on all parallel surfaces. Finally, we demonstrate the performance of the controller on the seven degrees of freedom lightweight robot DLR MIRO: the robot tracks complex trajectories while accurately keeping the desired distance to the surface and applying an almost constant contact force. Finally, we compare the approach to the state of the art.**

***Index Terms*—Compliance and impedance control, computational geometry, medical robots and systems, telerobotics and teleoperation, physical human-robot interaction.**

I. INTRODUCTION

ULTRASOUND (US) has become one of the most commonly used diagnostic and therapeutic imaging modalities in medicine [1]. US provides major advantages over other medical imaging techniques, such as low cost, portability, non-invasiveness, has real-time capabilities and is radiation-free [1]. However, US image acquisition requires the presence of a skilled sonographer, capable of simultaneously adjusting probe orientation and contact force to improve image quality and maximize patient comfort. This induces a very high cognitive and physical workload, while quality and repeatability of the imaging process become strictly operator-dependent [2].

Robotic ultrasound (RUS) systems have been subject of extensive research efforts [3] recently, ranging from teleoperated image acquisition, to semi- and fully autonomous RUS systems. These systems show promising trends towards mitigating the operator-dependence, high cognitive workload and ergonomic deficits of physician-performed US, while increasing the reproducibility and accessibility of diagnostic US scanning. Li *et al.* identify contact force control as a major feature in RUS image acquisition [3]. Any excessive force leading to tissue deformation can lead to image artifacts, patient pain and tissue damage. Simultaneously, good acoustic coupling is necessary to achieve high image quality.

Impedance control [4] is a very popular approach to realize compliant behavior, allowing the definition of force ranges interacting between robot and environment. Direct and indirect force control compose the two main techniques for active interaction control [5]. Indirect force control methods, e.g., via a feed-forward force, suffer from unpredictable, non-passive behavior upon contact loss. Similar issues arise in the direct force control paradigm and methods such as hybrid position/force control [6], lacking in robustness and ability to deal with environmental uncertainties. The authors in [7] propose a method combining force tracking and impedance control for interaction with arbitrary analytic surfaces, while proving passivity and stability for arbitrary passive environments.

Several authors have investigated solutions to RUS tissue scanning in the past [8]–[10]. Usually, a feed-forward force controller realizes a desired contact force between US probe and tissue. These solutions generally rely on the built-in force-torque (F/T) sensing capabilities of the robotic system to estimate the external force at the end-effector. Virga *et al.* present a RUS system for screening of abdominal aortic aneurysms and use a standard Cartesian impedance controller [11] in x, y to move the US probe along a trajectory based on elastical registration of the patient to a generic MRI-based atlas [8]. The simplicity of this control scheme makes it unsuitable for US scanning on tissues of complex geometry. Similarly, [10] use Cartesian impedance control guiding the robotic manipulator along

Manuscript received 24 February 2022; accepted 1 June 2022. Date of publication 22 June 2022; date of current version 1 July 2022. This letter was recommended for publication by Associate Editor X. Li and Editor C. Gosselin upon evaluation of the reviewers' comments. This work was supported in part by the International Graduate School of Science and Engineering (IGSSE); TUM Graduate School. *(Corresponding author: Arne Sachtler.)*

Michael Dyck, Arne Sachtler, and Alin Albu-Schäffer are with the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), 82234 Wessling, Germany, and also with the Department of Informatics, Technical University of Munich, 80333 Munich, Germany (e-mail: michael.dyck@dlr.de; arne.sachtler@dlr.de; Alin.Albu-Schaeffer@dlr.de).

Julian Klodmann is with the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), 82234 Wessling, Germany (e-mail: julian.klodmann@dlr.de).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2022.3184800, provided by the authors.

Digital Object Identifier 10.1109/LRA.2022.3184800

a predefined trajectory on the human spine. Augmenting the impedance controller with a feed-forward force controller they perform RUS scanning, intended for application in neurosurgical scenarios. Jiang *et al.* [9] additionally employ a F/T-sensor to improve force control accuracy and to automatically adjust the orientation of the US probe perpendicular to the tissue surface. Huang *et al.* [12] use a depth camera to estimate the 3D surface of the target anatomy via a point cloud. The robot is controlled in a point-to-point mode, using force feedback of two F/T-sensors attached to either side of the probe to adjust in-plane rotation.

Another field of research that involves controlling a tool onto a surface is robotic manufacturing like polishing, grinding, or painting. Many publications deal with these kinds of problems, e.g. [7], [13], [14]. The classical setting in these scenarios is precomputing a Cartesian trajectory based on the object model and then executing it using a Cartesian impedance controller. Contact forces are usually generated by additional feed-forward force control. Ochoa and Cortesão capture polishing movements from humans and transform them to surfaces by shifting and rotating the movements in space [15]. These approaches precompute trajectories and cannot be used for hands-on modes. Kana *et al.* show a method to control the end-effector to an arbitrary surface by introducing a virtual proxy point that wanders around on the surface [16]. Their method allows for hands-on scenarios in human-robot co-manipulation. Additional contact forces are applied using active force control.

We present a novel approach to RUS scanning using a passivity-based impedance control scheme [17] on arbitrary surfaces, allowing for hands-on US probe guidance, as well as for teleoperated or intuitive two-dimensional (2D) path planning for autonomous execution of scanning paths. Our proposed control paradigm is based on representation of the environment as discretized, triangular mesh objects, thus providing high flexibility for usage in various applications. To this end, pre-operative imaging can be used to obtain a three-dimensional (3D) representation of the tissue surface, that can be seamlessly integrated with our controller. In contrast to [7]–[10], [13]–[16], we present a novel method for interaction with arbitrary surfaces, relying solely on classical impedance control augmented by special task coordinates, inherently guaranteeing stability and passivity. One of these coordinates is the signed distance to the surface. This coordinate can be used to continuously adjust the interaction force and realize probe-surface interaction with a desired impedance.

A. Contributions

(C1) Impedance control on arbitrary surfaces featuring continuous interaction force adjustment and control over the penetration depth even for soft structures;

(C2) Novel choice of task coordinates respecting the surface geometry and curvature, and intuitively applicable for hands-on, teleoperated and autonomous RUS scanning;

(C3) Real-time control on triangular meshes, efficiently calculating surface-specific quantities in a 3 kHz control loop.

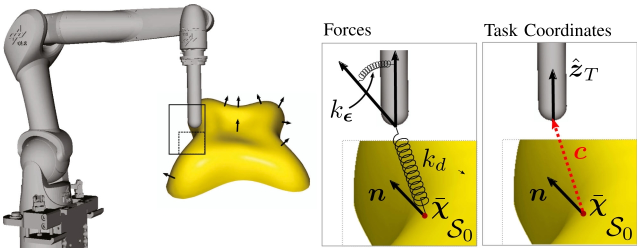


Fig. 1. Target surface and task coordinates of our impedance controller. Each point on the surface $\mathcal{S}_0$ can be assigned a unique, directed normal vector (left). The task coordinates result in a wrench pulling the tool to the surface and orienting it along the surface normal (middle). Calculation of the coordinates is based on the proxy point $\bar{\chi} \in \mathcal{S}_0$ on the surface, the corresponding surface normal $\boldsymbol{n}$ and the vector $\boldsymbol{c}$ between $\bar{\chi}$ and tool position $\boldsymbol{x}_t$.

II. Impedance Control on Parallel Surfaces

Our aim is to design an impedance controller regulating the end-effector of a robot to a desired signed distance from an arbitrary surface (Fig. 1). At the same time the controller shall align the end-effector with the normal of the surface.

Consider a two-dimensional target surface $\mathcal{S}_0 \subset \mathbb{R}^3$ and suppose that the surface is oriented, i.e., it has an in- and an outside. Each point $\boldsymbol{\chi} \in \mathcal{S}_0$ has a unit normal vector $\boldsymbol{n}(\boldsymbol{\chi})$ such that $\boldsymbol{n} : \mathcal{S}_0 \to \mathbb{S}^2$ with the two-sphere $\mathbb{S}^2$ is smooth on $\mathcal{S}_0$. The normal vectors allow defining consistent signed distances. Fig. 1 sketches an example for a target surface in yellow and shows some normal vectors in black.

Impedance controllers can be visualized as virtual springs and dampers in task coordinates. Likewise, with our proposed task coordinate the control goal of regulating the end-effector to a signed distance d and aligning the tool to the surface normal is achieved by virtual springs and dampers (Fig. 1).

When controlling the distance coordinate to $d \neq 0$, we implicitly control the end-effector onto a parallel surface $\mathcal{S}_d$ obtained by shifting the base surface $\mathcal{S}_0$ along the normals

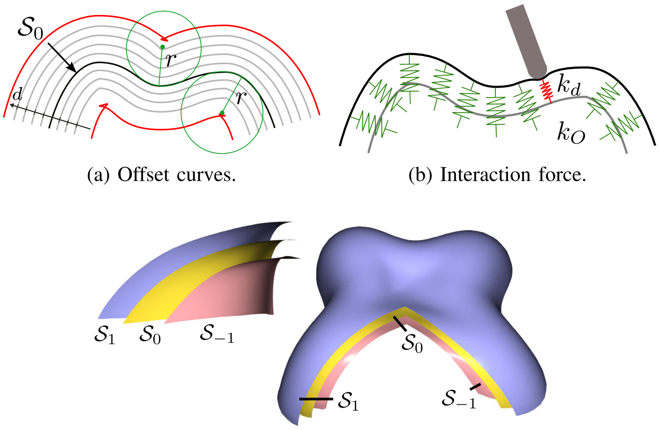
$$\mathcal{S}_d = \{\boldsymbol{\chi} + d\boldsymbol{n}(\boldsymbol{\chi}) \mid \boldsymbol{\chi} \in \mathcal{S}_0\}\,. \tag{1}$$

This family of surfaces generates a foliation, where the parallel surfaces and $\mathcal{S}_0$ are the leaves. We sketch the base leaf $\mathcal{S}_0$ and two parallel surfaces $\mathcal{S}_1$ and $\mathcal{S}_{-1}$ in Fig. 2(c).

Consider the example in two dimensions in Fig. 2(a) where a couple of offset surfaces as well as two osculating circles are shown. The parallel surfaces start self-intersecting as soon as the distance $|d|$ is larger than the smallest curvature radius $r_{\min}$ of the base surface. For $-r_{\min} < d < r_{\min}$ a smooth foliation is obtained and a one-to-one mapping between points on the parallel surfaces $\mathcal{S}_d$ and the base surface $\mathcal{S}_0$ can be obtained: we can find $\Psi_d : \mathcal{S}_0 \to \mathcal{S}_d$ such that Ψ is continuous and invertible. For parallel surfaces farther than $r_{\min}$ the offset surfaces self-intersect and no one-to-one mapping is possible. This generalizes to the three-dimensional case. We assume we stay in the allowed region for d from now on.

Because the map Ψ_d is invertible, we can transport functions $\varphi_0 : \mathcal{S}_0 \to \mathcal{Y}$ on $\mathcal{S}_0$ to $\mathcal{S}_d$

$$\varphi_d = \varphi_0 \circ \Psi_d^{-1}. \tag{2}$$



(a) Offset curves.

(b) Interaction force.

(c) Foliation by Parallel Surfaces

Fig. 2. Foliation view on the introduced task coordinates. (a) 2D example of offset curves with two representative osculating circles. Red curves: when going beyond the minimal curvature radius the map Ψ_d is no longer one-to-one. (b) Interaction force between tool and surface via a spring in the distance coordinate, pulling the tool towards its proxy point on $\mathcal{S}_0$. (c) Three leaves of the foliation as parallel surfaces given by $d = const.$

Setting $\varphi_0 = \boldsymbol{n}$ we can, for instance, pull the normals to all the other parallel surfaces $\mathcal{S}_d$. The same is true for any kind of chart on $\mathcal{S}_0$. Similar to the idea in previous work [18], we can first define coordinates on $\mathcal{S}_0$ and let them grow to other leaves of the foliation.

Commanding a desired distance $d < 0$, a passive interaction force can be realized (Fig. 2(b)). In contrast to classical approaches [7], no additional force control is required to be able to continuously adjust the interaction force, but the approach is rather based purely on passive impedance laws. Given the stiffness of the controlled distance spring k_d and the object stiffness k_O the required distance for a desired contact force f can be computed by $d = \frac{f}{k_d - k_O}$.

A. Task Coordinates and Impedance Controller

Let us define the surface-specific task coordinates $\boldsymbol{p} \in \mathbb{R}^4$

$$\boldsymbol{p}(\boldsymbol{q}) = \begin{bmatrix} d(\boldsymbol{q}) \\ \boldsymbol{\epsilon}(\boldsymbol{q}) \end{bmatrix}, \tag{3}$$

where $d(\boldsymbol{q})$ computes the signed-distance of the end-effector to the surface. Let $\bar{\boldsymbol{\chi}}(\boldsymbol{q})$ implement a map Ψ (2) between $\mathcal{S}_d$ and $\mathcal{S}_0$. As our manifold is oriented, we can look up the normal of the proxy point $\bar{\boldsymbol{\chi}}(\boldsymbol{q})$ and take the quaternion $\boldsymbol{h} = [\lambda,\, \boldsymbol{\epsilon}] \in \mathbb{H}$ between the end-effector orientation and the normal of the proxy point. We stack its vector part $\boldsymbol{\epsilon}$ into $\boldsymbol{p}(\boldsymbol{q})$ (3). $\boldsymbol{q} \in \mathcal{Q}$ are the generalized configuration coordinates of the manipulator.

Using the task coordinate function (3) we define an impedance control law [4], [11]

$$\boldsymbol{\tau} = \boldsymbol{J}_p^{\mathsf{T}}(\boldsymbol{q}) \left[\boldsymbol{K}(\boldsymbol{p}_d - \boldsymbol{p}(\boldsymbol{q})) + \boldsymbol{D}(\boldsymbol{q})(\dot{\boldsymbol{p}}_d - \dot{\boldsymbol{p}}) \right], \tag{4}$$

where $\boldsymbol{K}$ is a symmetric, positive definite (pd) stiffness matrix and $\boldsymbol{J}_p(\boldsymbol{q}) = {}^{\partial \boldsymbol{p}}/_{\partial \boldsymbol{q}}$ denotes the Jacobian of $\boldsymbol{p}(\boldsymbol{q})$. The symmetric and pd damping matrix $\boldsymbol{D}(\boldsymbol{q})$ is designed based on the double diagonalization design proposed in [19]. For $\boldsymbol{p}_d = \dot{\boldsymbol{p}}_d = \boldsymbol{0}$ the controller will pull the end-effector onto the surface and align its z-axis with the normal of the surface. Fig. 1 shows this case for the example surface representing the forces as springs. This choice of desired coordinates is particularly useful for interaction cases, e.g. when clinicians want to guide the robot in hands-on mode along the tissue.

Let us now design $\boldsymbol{p}(\boldsymbol{q})$. Assume the Cartesian forward kinematics $f_{\text{kin}} : \mathcal{Q} \to SE(3) \,|\, \boldsymbol{q} \mapsto \boldsymbol{x}$ is given and also the corresponding geometric Jacobian $\boldsymbol{J}_x(\boldsymbol{q})$

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix} = \boldsymbol{J}_x(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{5}$$

where $\boldsymbol{\omega}$ denotes the angular velocity vector and $\boldsymbol{v}$ translational velocities. We defer the computation of the proxy point on the surface $\bar{\boldsymbol{\chi}}(\boldsymbol{q})$ to a later section and assume it given for now. Let $\boldsymbol{c}(\boldsymbol{q}) = \boldsymbol{x}_t - \bar{\boldsymbol{\chi}}$, with $\boldsymbol{x}_t \in \mathbb{R}^{3\times 1}$ representing the translational component of $\boldsymbol{x} = f_{\text{kin}}(\boldsymbol{q})$. Then the signed distance is determined by (rightmost column Fig. 1)

$$d(\boldsymbol{q}) = \operatorname{sign}(\boldsymbol{n}(\bar{\boldsymbol{\chi}}(\boldsymbol{q})) \cdot \boldsymbol{c}(\boldsymbol{q}))\, \|\boldsymbol{c}(\boldsymbol{q})\|. \tag{6}$$

Next, let's design $\boldsymbol{\epsilon}(\boldsymbol{q})$ via the quaternion $\boldsymbol{h}(\boldsymbol{q})$ describing the relative rotation between the tool z-axis $\hat{\boldsymbol{z}}_T(\boldsymbol{q})$ and the unit normal $\boldsymbol{n}(\bar{\boldsymbol{\chi}})$. For any input vectors $\boldsymbol{n} \neq -\hat{\boldsymbol{z}}_T$ we use

$$\boldsymbol{h}(\boldsymbol{q}) = \left[\frac{\sqrt{2\,(1 + \hat{\boldsymbol{z}} \cdot \boldsymbol{n})}}{2},\; \underbrace{\frac{\hat{\boldsymbol{z}} \times \boldsymbol{n}}{\sqrt{2(1 + \hat{\boldsymbol{z}} \cdot \boldsymbol{n})}}}_{\boldsymbol{\epsilon}(\boldsymbol{q})} \right]. \tag{7}$$

and $\boldsymbol{h}(\boldsymbol{q}) = [0,\, \boldsymbol{n}^{\perp}]$ otherwise. The vector $\boldsymbol{n}^{\perp}$ is any unit vector orthogonal to $\boldsymbol{n}$. The combination of (6) and (7) yields the desired task coordinates $\boldsymbol{p}(\boldsymbol{q})$.

For the impedance controller we need the overall Jacobian $\boldsymbol{J}_p \in \mathbb{R}^{4\times n}$, where n denotes the number of joints. Apply the chain rule and write

$$\dot{\boldsymbol{p}} = \frac{\partial \boldsymbol{p}}{\partial t} = \frac{\partial \boldsymbol{p}(\boldsymbol{x})}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{x}(\boldsymbol{q})}{\partial \boldsymbol{q}} \frac{\partial \boldsymbol{q}(t)}{\partial t} = \boldsymbol{J}_{px}\boldsymbol{J}_x\dot{\boldsymbol{q}} = \boldsymbol{J}_{px}\dot{\boldsymbol{x}}. \tag{8}$$

We can express the desired Jacobian $\boldsymbol{J}_p$ as a product $\boldsymbol{J}_p = \boldsymbol{J}_{px}\boldsymbol{J}_x$ of the known geometrical Jacobian $\boldsymbol{J}_x$ and a new Jacobian $\boldsymbol{J}_{px}$ to be determined. Let's split it into blocks:

$$\dot{\boldsymbol{p}} = \boldsymbol{J}_{px}\dot{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{J}_{dv} & \boldsymbol{0} \\ \boldsymbol{J}_{\epsilon v} & \boldsymbol{J}_{\epsilon\omega} \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix}. \tag{9}$$

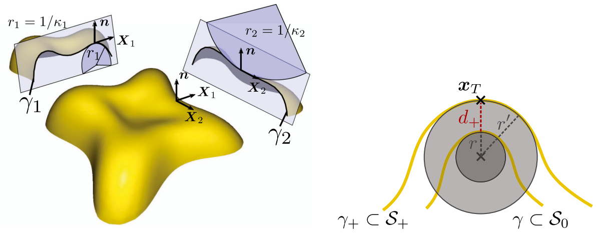
The first component $\boldsymbol{J}_{dv} \in \mathbb{R}^{1\times 3}$ describes the linear relation between translational Cartesian velocities $\boldsymbol{v}$ and the velocity in the distance coordinate $\dot{d}$. This relation is given by the orthogonal projection of $\boldsymbol{v}$ onto the normal vector between the proxy point $\bar{\boldsymbol{\chi}}$ and the end-effector position $\boldsymbol{x}_t$

$$\boldsymbol{J}_{dv} = \frac{(\boldsymbol{x}_t - \bar{\boldsymbol{\chi}})^{\mathsf{T}}}{|\boldsymbol{x}_t - \bar{\boldsymbol{\chi}}|}. \tag{10}$$

The mapping from angular velocities $\boldsymbol{\omega} \in \mathbb{R}^3$ of the end-effector onto velocities in $\boldsymbol{\epsilon} \in \mathbb{R}^3$ is given by [20]

$$\boldsymbol{J}_{\epsilon\omega} = \frac{1}{2} \begin{bmatrix} \lambda & \epsilon_3 & -\epsilon_2 \\ -\epsilon_3 & \lambda & \epsilon_1 \\ \epsilon_2 & -\epsilon_1 & \lambda \end{bmatrix}, \tag{11}$$

with λ, $\epsilon_{1,2,3}$ being the components of the quaternion $\boldsymbol{h}$.



(a) Normal and Principal Curvatures (b) Curvature Change

Fig. 3. (a) Normal curvature and principal curvatures. Intersecting the surface with a plane containing the normal and a tangent vector $\boldsymbol{X}$ yields a locally planar curve with normal curvature κ_n. (b) Parallel surfaces share principal directions of curvature, the curvature values change with the distance d.

Lastly, the component $\boldsymbol{J}_{\epsilon v} \in \mathbb{R}^{3\times 3}$ relates translational Cartesian velocities and velocities $\dot{\boldsymbol{\epsilon}}$ describing the rate of change of the relative orientation between surface normal and the end-effector. In other words, $\boldsymbol{J}_{\epsilon v}$ represents the change in $\boldsymbol{\epsilon}$ by the change in the surface normal $d\boldsymbol{n}$, arising from any tangential motion along the non-planar surface.

We briefly recap some central concepts of the differential geometry of surfaces that are relevant to finally derive $\boldsymbol{J}_{\epsilon v}$. The presented material can be found in any textbook like [21]; we recommend [22] for a visual and intuitive introduction. Consider again $\boldsymbol{n} : \mathcal{S}_0 \to \mathbb{S}^2$, the vector field mapping points on the manifold S_0 to their corresponding unit normal vector. The map $\boldsymbol{n}$ associates each point on $\mathcal{S}_0$ to a point on the unit sphere S^2. This is known as the *Gauß map*. Now, the differential $d\boldsymbol{n}$ of the Gauß map, referred to as the *Weingarten map*, tells how the surface normal changes while moving along the surface. Since $\boldsymbol{n}$ maps onto unit vectors, we know that any change of the surface normal can only be tangential to the surface itself. Hence, the Weingarten map $d\boldsymbol{n}$ in a tangential direction $\boldsymbol{X} \in \mathcal{T}_{\boldsymbol{\chi}}\mathcal{S}_0$ at a point $\boldsymbol{\chi} \in \mathcal{S}_0$ is a mapping from the tangent space $\mathcal{T}_{\boldsymbol{\chi}}\mathcal{S}_0$ to itself [21]. The Weingarten map $d\boldsymbol{n}$ relates velocities in the tangent space to changes in the normal, which is precisely what we need for the block $\boldsymbol{J}_{\epsilon v}$.

Before deriving $\boldsymbol{J}_{\epsilon v}$, let us briefly recap normal curvature and principal curvature with help of the example in Fig. 3(a). The normal curvature κ_n at a point $\boldsymbol{\chi} \in \mathcal{S}_0$ in direction $\boldsymbol{X} \in \mathcal{T}_{\boldsymbol{\chi}}\mathcal{S}_0$ is the curvature of the planar curve $\gamma \subset \mathcal{S}_0$ obtained by intersecting the surface with the plane spanned by $\boldsymbol{X}$ and the normal $\boldsymbol{n}(\boldsymbol{\chi})$. The blue planes in Fig. 3(a) show two special cases of such planes. We take the tangential directions $\boldsymbol{X}_1$, $\boldsymbol{X}_2$ where the surface bends the most and least. These directions are called principal directions of curvature and their corresponding curvature values are called principal curvatures κ_1, κ_2. Fig. 3(a) shows the resulting planar curves $\gamma_{1,2}$ and the corresponding osculating circles with radius $1/\kappa_{1,2}$. Principal directions are always orthogonal [21].

We use the directions of principal curvature at each point to determine local coordinates of the manifold $\mathcal{S}_0$. In these coordinates the second fundamental form $I\!I$ is particularly simple, being a diagonal matrix of the principal curvatures. The Weingarten map can be computed projecting a vector $\boldsymbol{v}$ onto the tangent plane spanned by $\boldsymbol{X}_1$, $\boldsymbol{X}_2$ and applying the second fundamental form [23]

$$d\boldsymbol{n}(\boldsymbol{v}) = \boldsymbol{P}^{\mathsf{T}} \boldsymbol{I\!I} \boldsymbol{P} \boldsymbol{v} = \boldsymbol{P}^{\mathsf{T}} \begin{bmatrix} \kappa_1 & 0 \\ 0 & \kappa_2 \end{bmatrix} \boldsymbol{P} \boldsymbol{v}, \tag{12}$$

where $\boldsymbol{P} \in \mathbb{R}^{2\times 3}$ contains $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ as rows.

Using these concepts we derive an expression for our Jacobian component $\boldsymbol{J}_{\epsilon v} \in \mathbb{R}^{3\times 3}$, relating translational Cartesian velocities of the tool to velocities in the orientational task coordinates $\boldsymbol{\epsilon}$

$$\boldsymbol{J}_{\epsilon v} = \boldsymbol{J}_{\epsilon \omega} [\boldsymbol{n}]_\times \boldsymbol{P}^{\mathsf{T}} \boldsymbol{I\!I} \boldsymbol{P}, \tag{13}$$

where $[\boldsymbol{n}]_\times$ denotes the skew-symmetric cross product matrix of the vector $\boldsymbol{n}$. The cross-product matrix is introduced to transform the change in the normal to angular velocities.

Consider a case where the robot is at a distance $d_+ > 0$ and assume the robot is moving along the surface keeping d_+ constant. We travel along a parallel surface $\mathcal{S}_+$. Fig. 3(b) depicts a local section of the planar curves $\gamma \subset \mathcal{S}_0$ and $\gamma_+ \subset \mathcal{S}_+$. Both surfaces $\mathcal{S}_0$ and $\mathcal{S}_+$ have the same principal directions [21]. However, the radius $r' = r + d_+$ of the osculating circle of $\mathcal{S}_+$differs from the radius r of the osculating circle and thus the curvature of $\mathcal{S}_0$. This change in curvature for parallel surfaces with an offset d is encoded in the second fundamental form in principal coordinates by

$$\tilde{\boldsymbol{I\!I}} = \begin{bmatrix} \kappa_1(1 - d\kappa_1) & 0 \\ 0 & \kappa_2(1 - d\kappa_2) \end{bmatrix}. \tag{14}$$

This generalizes to $d = 0$ and we consequently implement (13) always using $\tilde{\boldsymbol{I\!I}}$.

B. Optional Surface Parametrization

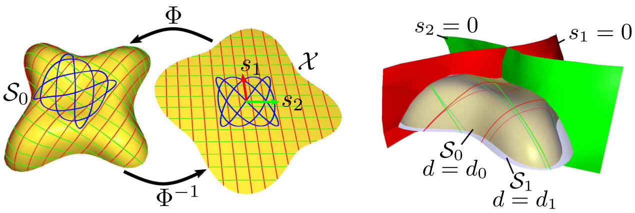
The concept in the previous section allows controlling the robot to a distance to the surface and aligning the surface normal and the tool. This is especially helpful for interaction cases when the position on the surface shall not be constrained.

For a variety of tasks, such as teleoperation or autonomous path scanning (cmp. Section I) it is also required to stabilize and control the particular position on the surface. In that case we require additional coordinates to specify the position of the proxy point $\bar{\boldsymbol{\chi}}$ on the surface. We consider 2D surfaces here, therefore two additional coordinates are required.

We need the inverse of a parametrization $\Phi : \mathcal{X} \to \mathcal{S}_0$ of a surface where $\mathcal{X}$ is called the parameter domain [21]. The function $\Phi(\boldsymbol{s})$, with $\boldsymbol{s} = [s_1, s_2]^{\mathsf{T}}$, maps a point $\boldsymbol{s} \in \mathcal{X}$ onto the immersed surface $\mathcal{S}_0$. Provided that we find a $\Phi(\boldsymbol{s})$ for our target surface $\mathcal{S}_0$, we can use its inverse $\boldsymbol{s} = \Phi^{-1}(\boldsymbol{\chi})$ as global coordinates to specify where we are on the surface. We augment the previously chosen task coordinates:

$$\boldsymbol{\rho}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{s}(\bar{\boldsymbol{\chi}}(\boldsymbol{q})) \\ \boldsymbol{p}(\boldsymbol{q}) \end{bmatrix} = \begin{bmatrix} s_1(\boldsymbol{q}) \\ s_2(\boldsymbol{q}) \\ d(\boldsymbol{q}) \\ \boldsymbol{\epsilon}(\boldsymbol{q}) \end{bmatrix}. \tag{15}$$

Any non-singular parametrization Φ can be used. We have defined coordinates on the base manifold $\mathcal{S}_0$ which we can propagate to all leaves $\mathcal{S}_d$ of the foliation using Ψ_d as in [18]. In our case Ψ_d is implemented by the proxy point function $\bar{\boldsymbol{\chi}}$, i.e., on $\mathcal{S}_d$ we use $\boldsymbol{s} = \Phi^{-1} \circ \bar{\boldsymbol{\chi}}$. Thus, we implicitly have consistent



(a) Parametrization (b) Orthogonal Foliations

Fig. 4. (a) Parametrization Φ of the surface $\mathcal{S}_0 \subset \mathbb{R}^3$ with coordinates s_1, s_2 in the parameter domain $\mathcal{X}$. Red and green lines represent isolines of the coordinates s_1, s_2 in $\mathcal{X}$ and $\mathcal{S}_0$, respectively. The blue trajectory shows an exemplary lissajous curve in $\mathcal{X}$ and its corresponding image on $\mathcal{S}_0$. (b) Orthogonal foliations represented by $s_1 =$ const. (red), $s_2 =$const. (green).

coordinates $\boldsymbol{s}$ on all leaves $\mathcal{S}_d$ and obtain also foliations for $s_1 =$ const. and $s_2 =$ const. (Fig. 4(b)).

In order to incorporate these parametrization coordinates into the impedance control, we extend our Jacobian $\boldsymbol{J}_{px}$ to include the mapping from translational velocities $\boldsymbol{v}$ to velocities $\dot{\boldsymbol{s}}$

$$\dot{\boldsymbol{\rho}} = \boldsymbol{J}_{\rho x}(\boldsymbol{x})\dot{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{J}_{sv} & \boldsymbol{0} \\ \boldsymbol{J}_{dv} & \boldsymbol{0} \\ \boldsymbol{J}_{\epsilon v} & \boldsymbol{J}_{\epsilon\omega} \end{bmatrix} \dot{\boldsymbol{x}}. \tag{16}$$

For any point on the surface the gradients ∇s_1 and ∇s_2 determine the directional derivatives of s_1 and s_2 for tangent vectors to the surface. However, the gradients ∇s_1 and ∇s_2 cannot be computed for points $\boldsymbol{x} \notin \mathcal{S}_0$. Therefore, we again use the proxy point $\bar{\boldsymbol{\chi}}$ and need to adapt the gradients to compensate for the change of curvature (see Fig. 3(b)). Intuitively, for curved surfaces, the proxy and query point do not travel with the same speed. Consider again the local coordinates given by the principal directions $\boldsymbol{P} \in \mathbb{R}^{2\times 3}$ used in (12). Then, the first fundamental form on a parallel surface at given distance d to $\mathcal{S}_0$ in local principal coordinates is given by

$$\tilde{\boldsymbol{I}} = \begin{bmatrix} (1 - d\kappa_1)^2 & 0 \\ 0 & (1 - d\kappa_2)^2 \end{bmatrix}. \tag{17}$$

To include the influence of d into our Jacobian, we set

$$\boldsymbol{J}_{sx} = \begin{bmatrix} \nabla^{\mathsf{T}} s_1(\bar{\boldsymbol{\chi}}) \\ \nabla^{\mathsf{T}} s_2(\bar{\boldsymbol{\chi}}) \end{bmatrix} \boldsymbol{P}^{\mathsf{T}} \begin{bmatrix} (1 - d\kappa_1)^2 & 0 \\ 0 & (1 - d\kappa_2)^2 \end{bmatrix} \boldsymbol{P}. \tag{18}$$

We calculate the total Jacobian $\boldsymbol{J}_\rho(\boldsymbol{q}) \in \mathbb{R}^{6\times n}$ relating joint velocities $\dot{\boldsymbol{q}}$ to velocities in the task coordinates $\dot{\boldsymbol{\rho}}$ via

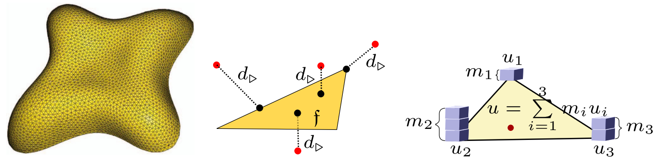
$$\dot{\boldsymbol{\rho}} = \boldsymbol{J}_\rho(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{J}_{\rho x}(\boldsymbol{x})\boldsymbol{J}_x(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{19}$$

and implement an impedance controller in $\boldsymbol{\rho}$ by

$$\boldsymbol{\tau} = \boldsymbol{J}_\rho^{\mathsf{T}}(\boldsymbol{q})\left[\boldsymbol{K}_\rho(\boldsymbol{\rho}_d - \boldsymbol{\rho}(\boldsymbol{q})) + \boldsymbol{D}_\rho(\boldsymbol{q})(\dot{\boldsymbol{\rho}}_d - \dot{\boldsymbol{\rho}})\right]. \tag{20}$$

C. Discrete Differential Geometry

Instead of using analytic methods to describe the target surface $\mathcal{S}_0$ as in e.g. [7], [24], we use a discrete representation in terms of a triangular mesh. Triangular meshes provide a flexible interface to load surfaces into the controller and are generated e.g. by CAD software, 3D scanning or computer vision approaches [25]. Additionally, the discrete version can handle basically arbitrary surfaces including complicated topologies and sharp corners. In contrast to the smooth setting, where a surface with sharp corners and edges is not manifold, discrete differential geometry offers techniques to handle edges and corners by smoothing out the normal map.



(a) Discretized surface. (b) Closest point. (c) Barycentric coordinates.

Fig. 5. Discretization of the smooth surface $\mathcal{S}_0$. (a) Wireframe of the discretized surface. (b) Closest point query between tool position and surface. (c) Barycentric coordinates used for interpolation of various per-vertex quantities defined on the triangular mesh.

Let the target surface $\mathcal{S}_0$ be described by a triangular mesh (see Fig. 5(a)). This is encoded by a matrix of N vertices $\boldsymbol{V} \in \mathbb{R}^{N\times 3}$ and M faces $\boldsymbol{F} \in \mathbb{N}^{M\times 3}$, where the entries in $\boldsymbol{F}$ are the vertex indices. Further, we assume a list of normals $\boldsymbol{N} \in \mathbb{R}^{N\times 3}$, where the i-th row is the unit normal at the i-th vertex.

For the task coordinates the proxy point query $\bar{\boldsymbol{\chi}}(\boldsymbol{q})$ must be implemented. We first use the Cartesian forward kinematics to determine the position of the end-effector $\boldsymbol{x}_t(\boldsymbol{q})$. Then, we find the closest triangle

$$\bar{\mathfrak{f}}(\boldsymbol{q}) = \min_{j\in\{1..M\}} d_\triangleright(\mathfrak{f}_j, \boldsymbol{x}_t(\boldsymbol{q})), \tag{21}$$

where $d_\triangleright(\mathfrak{f}, \boldsymbol{x})$ computes the distance of a point $\boldsymbol{x} \in \mathbb{R}^3$ to the face $\mathfrak{f}$ (Fig. 5(b)). In order to perform the query (21) efficiently we use an axis-aligned bounding box (AABB) tree [26], which can be constructed before execution. The AABB tree allows query in $\mathcal{O}(\log M)$ time and thus enables us to use (21) in real-time applications also for large meshes. The construction time of the tree is in the order of milliseconds to seconds for meshes with up to a million triangles.

Given the closest triangle $\bar{\mathfrak{f}}$ we compute the corresponding closest point on it and take the proxy point $\bar{\boldsymbol{\chi}}$ to be that point. We have taken the term *proxy point* from Kana *et al.* [16]. They project the connection vector into the tangent space of the surface at the proxy point and then describe its motion using first order dynamics. On the one hand this ensures a continuous trajectory of the proxy point; on the other hand the additional dynamics make the task forces depend not only on $\boldsymbol{q}$, but also on their history: one cannot integrate the task forces to potential fields and there are no coordinates, such that their proxy point position is a function of $\boldsymbol{q}$ only. In contrast, we directly take the closest point query to obtain the proxy point: our coordinates only depend on $\boldsymbol{q}$, not on any dynamics. We ensure continuity by constraining the maximally allowed distance to r_{min} as before. In this region the map Ψ_d^{-1} between parallel surfaces $\mathcal{S}_d$ and $\mathcal{S}_0$ is one-to-one and continuous; it is implemented by the closest point query.

In discrete meshes many properties like the normals, curvatures and later the parametrization $\boldsymbol{s}(\boldsymbol{\chi})$ are only known on the vertices. Therefore, we apply the principle of barycentric interpolation. Given a point $\boldsymbol{\chi}$ on the triangle we assign masses m_i to the three vertices of the triangle such that $\sum_i m_i = 1$. The barycenter of the triangle with the new vertex masses will be at $\boldsymbol{\chi}$. Suppose we have a scalar function $u : \mathcal{S}_0 \to \mathbb{R}$ on the surface, which we only know on the vertices. Using the barycentric coordinates we can approximate $u(\boldsymbol{\chi})$ at any point on the surface by (see Fig. 5(c))

$$u(\boldsymbol{\chi}) = \sum_{i=1}^{3} m_i(\boldsymbol{\chi}) u(\mathfrak{v}_i). \tag{22}$$

We use barycentric interpolation on the vertex normals to obtain the normal at the proxy point $\boldsymbol{n}(\bar{\boldsymbol{\chi}}(\boldsymbol{q}))$.

In order compute the Jacobian (9) also the principal curvatures and principal curvature directions are required. We use quadric fitting by Panozzo *et al.* [27] to estimate the principal curvatures and their directions at every vertex and again use barycentric interpolation to enable computing the them everywhere. The principal curvatures and their directions are precomputed and stored for every vertex.

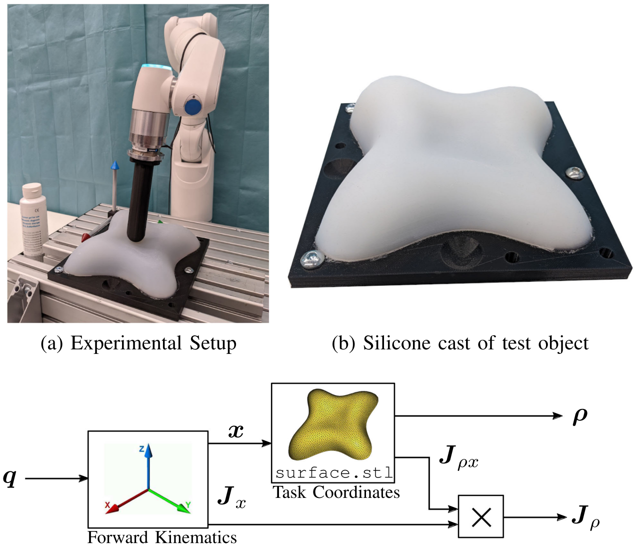
Next we add the parametrization $\boldsymbol{\chi} = \Phi(\boldsymbol{s})$ for the discrete mesh. Here we assume that the surface is of disk-topology, i.e., two-dimensional, with no holes and a single boundary. To our advantage this is a well studied problem in the field of computer graphics and various approaches are available. Sheffer *et al.* [28] provide an overview on these techniques. We use the *as-rigid-as-possible* (ARAP) parametrization by Liu *et al.* [29], which optimizes for isometry. We have shown the result obtained by this algorithm already in Fig. 4(a).

The rationale behind using the ARAP parametrization becomes clear when considering the telemanipulation case: using a joystick or arrow-keys to command velocities in the parameter domain $\mathcal{X}$, we desire that these velocities map to the same velocities on the surface. Additionally, isometry implies conformality, i.e, arrow keys in orthogonal directions will also correspond to right angles on the surface.

The output of the ARAP parametrization algorithm is a matrix $\boldsymbol{S} \in \mathbb{R}^{N \times 2}$, where the i-th row $\boldsymbol{s}(\mathfrak{v}_i) \in \mathcal{X}$ represents the coordinates of the i-th vertex in the parametrization domain $\mathcal{X}$. Using barycentric interpolation we compute $\boldsymbol{s}(\boldsymbol{\chi})$. Finally, the gradients ∇s_k are required, which we obtain calculating the discrete gradient operator for the piecewise linear functions s_k at each mesh vertex and again interpolating barycentrically. Now, we have everything to implement the concept from the smooth setting on the discrete mesh.

III. Experiments

We demonstrate our approach in our experimental setup with the seven degrees of freedom (DoF) robot arm DLR MIRO [30]. It is equipped with torque sensors allowing to directly command the joint torques computed by our impedance controller (4) or (20) to the joint-level torque controllers. Fig. 6(a) shows our experimental setup. To simulate soft tissue, we have cast a model of our target surface using *Dragon Skin*$^{\text{TM}}$ silicone with Shore 10 A hardness (on the ASTM D-2240 scale) with a wall thickness of approximately 1 cm and solid material underneath (Fig. 6(b)). From that we can estimate the object stiffness k_O and thus specify d_d and control gain k_d to realize intended passive interaction forces. Further, we take a dummy tool consisting of a cylinder and a spherical tip and a total length of 14 cm. We use MIRO's built-in F/T-sensors (10 DoF) to estimate the interaction force on the tool tip.


(a) Experimental Setup

(b) Silicone cast of test object

(c) Interface to our implementation

Fig. 6. Experimental setup, silicone phantom and implementation.

We implemented our impedance controller (20) in MATLAB/Simulink, and designed a custom Simulink block in C++ using the *libigl* library [31] for geometry processing, calculating our task coordinates $\boldsymbol{\rho}(\boldsymbol{x})$ and Jacobian $\boldsymbol{J}_{\rho x}(\boldsymbol{x})$ (Fig. 6(c)). This can be easily connected to a block computing the forward kinematics of any robot. We output the task coordinates and Jacobian, which is multiplied to the geometric Jacobian of the robot to obtain $\boldsymbol{J}_\rho$ as required by the impedance controller (20). The target surface $\mathcal{S}_0$ is specified by a triangular mesh in terms of an STL or Wavefront file.

Our test surface consists of $N = 13367$ vertices and $M = 26360$ faces. It has a maximum curvature of $\kappa_{\text{max}} = 65.77$ and we set our allowed distance cautiously to $d_{\text{max}} = 15$ mm. For this surface, the construction of the AABB tree takes $t_{\text{init}} = 18$ ms. One cycle of querying the tree and computing all quantities required for $\boldsymbol{\rho}$ and $\boldsymbol{J}_{\rho x}$ takes $t_{\text{call}} = 11.91$ ms. We use our controller within a 3 kHz real-time control loop.

The seven DoF robot is kinematically redundant under the given choice task coordinates $\boldsymbol{\rho}(\boldsymbol{q})$ and has an inherent nullspace at the elbow joint. We control this nullspace and apply an additional joint spring in the seventh joint and project them by a dynamically consistent nullspace projection [32].

We provide a video of all experiments in the multimedia material attached to the letter. Additionally, we perform experiments on teleoperation mode and hands-on guidance and show results in the video.

A. Commanding a Trajectory in Our Task Coordinates

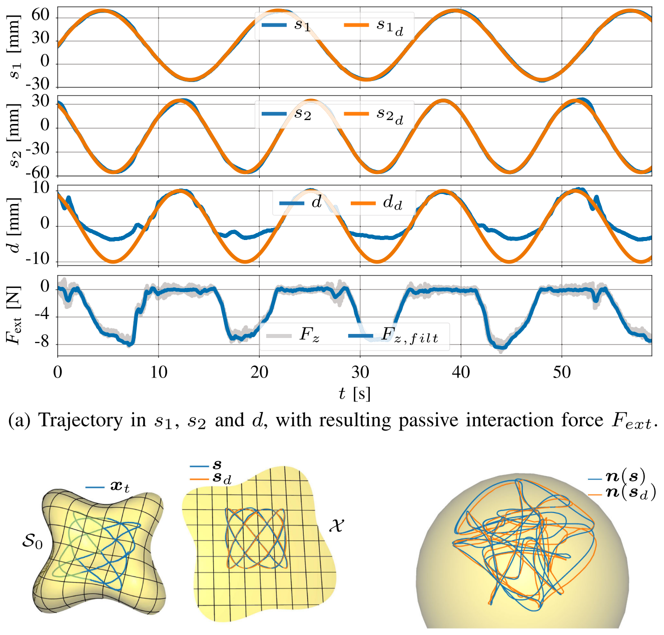
For the first experiment, we command a desired trajectory in the surface coordinates $\boldsymbol{s}$ and d by defining a Lissajous curve

TABLE I
STIFFNESS PARAMETERS FOR THE EXPERIMENTS

	k_{s_1}	k_{s_2}	k_d	k_{ϵ_1}	k_{ϵ_2}	k_{ϵ_3}
unit	N/m	N/m	N/m	Nm/rad	Nm/rad	Nm/rad
III-A	2500	2500	1000	50	50	50
III-B2	2500	2500	0	50	50	50
III-B3	2500	2500	250	50	50	50
	k_x	k_y	k_z	k_α	k_β	k_γ
unit	N/m	N/m	N/m	Nm/rad	Nm/rad	Nm/rad
III-B1	2500	2500	0	50	50	50

TABLE II
MEAN AND STD OF INTERACTION FORCE IN SECTION III-B

	III-B1		III-B2		III-B3	
	mean	std	mean	std	mean	std
F_z $[N]$	-2.178	0.521	-2.183	0.495	-2.403	0.498



(a) Trajectory in s_1, s_2 and d, with resulting passive interaction force F_{ext}.

(b) Lissajous trajectory on $\mathcal{X}$ and $\mathcal{S}_O$.

(c) Image of Gauß map on $\boldsymbol{s}_d$.

Fig. 7. Results of commanding a Lissajous trajectory in our task coordinates.

(cmp. Fig. 4(a)) in $\boldsymbol{s}$ and a sine trajectory in d

$$\begin{bmatrix} s_{1,d}(t) \\ s_{2,d}(t) \\ d_d(t) \end{bmatrix} = 45 \text{ mm} \begin{bmatrix} \sin(3\omega_0 t) \\ \sin(4\omega_0 t) \\ \sin(4\omega_0 t) \end{bmatrix} \quad \text{for } \omega_0 = 0.12\ s^{-1}. \tag{23}$$

Further, we command $\boldsymbol{\epsilon}_d = \mathbf{0}$, i.e., the tool's normal shall follow the surface normal. We set the stiffness matrix $\boldsymbol{K}_\rho$ to a diagonal matrix with the values given in Table I (row III-A) and specify a damping ratio of $\zeta = 0.7$. The selected values rest on heuristic experimental examination and the known stiffness of our silicone object. Since we deploy an almost isometric ARAP parametrization, we note that the units of the stiffnesses k_{s_1}, k_{s_2} will only be approximately N/m. We use the impedance control law (20) and achieve the results shown in Fig. 7. The evolutions of the three task coordinates s_1, s_2 and d are shown in Fig. 7(a) together with the external force F_{ext} in z-direction measured at the end-effector. Additionally, we show the achieved and desired trajectory in the parametrization domain in Fig. 7(b) and the resulting trajectory of the tool tip on the surface. Finally, Fig. 7(c) presents the image of the Gauß map, i.e., the path unit vectors trace out on the unit sphere, for the desired normal $\boldsymbol{n}$ on $\boldsymbol{s}_d$ and the actual orientation of the tool axis $\hat{\boldsymbol{z}}_T$. Overall we achieve a good tracking performance, with a root-mean-square error (RMSE) in the parametrization coordinates of $\text{RMSE}_{\boldsymbol{s}_1} = 0.917$ mm, $\text{RMSE}_{\boldsymbol{s}_2} = 1.465$ mm. Commanding a sine trajectory in the distance coordinate we show consistency of the parametrization coordinates $\boldsymbol{s}$ along all leaves of the foliation, and demonstrate the effect of controlling the tool at different distances to the surface. For sections where $d_d > 0$ mm we achieve a $\text{RMSE}_d = 0.871$ mm. As expected, for $d_d < 0$ mm a discrepancy between actual and desired distance remains. As a result, the tool exerts forces onto the surface, whose magnitude depends on object and controller stiffnesses k_O, k_d, and the magnitude of $d_d < 0$ mm. We demonstrate how controlling the distance can be used to realize passive interaction forces and, more importantly, continuously adjust these forces without the need of explicit force control. At the same time, our control method provides a straightforward way of limiting the penetration depth of the US tool into the tissue. The images of the Gauß map along the commanded and actual trajectories $(\boldsymbol{n}(\boldsymbol{s}_d), \boldsymbol{n}(\boldsymbol{s}))$ in Fig. 7(c) illustrate the high curvature of our test surface and the capability of our controller to accurately realize a tool orientation normal to the surface.

B. Comparison of Contact Force Control Strategies

We show three additional experiments, comparing different strategies for contact force control, which is of major importance for US scanning. The stiffness parameters for all three experiments are listed in Table I.

1) Cartesian Impedance Control With Feed-Forward Force: For comparison to the state of the art we use classical 6D Cartesian impedance control to move the robot along a pre-planned trajectory on the surface. We use the same trajectory (23) in $\boldsymbol{s}$ and map each point on the Lissajous curve to 6D Cartesian poses $\boldsymbol{x} \in SE(3)$. We set the stiffness in the z-coordinate to $K_z = 0$ N/m and add a feed-forward force term $\boldsymbol{F}_d \in \mathbb{R}^6$ with z-component $F_{z,d} = -2$ N and zero elsewhere

$$\boldsymbol{\tau} = \boldsymbol{J}_{\boldsymbol{x}}^{\mathsf{T}}(\boldsymbol{q}) \left[\boldsymbol{F}_d + \boldsymbol{K}_{\boldsymbol{x}}(\boldsymbol{x}_d - \boldsymbol{x}) + \boldsymbol{D}_{\boldsymbol{x}}(\dot{\boldsymbol{x}}_d - \dot{\boldsymbol{x}}) \right]. \tag{24}$$

We show the results in the left column in Fig. 8. The upper row shows the distance of the tool tip to the phantom surface, the lower figure depicts the measured external force at the end-effector in z-direction. Mean and std of the interaction force F_z are given in Table II, which shows constant and accurate interaction force overall. Observe that the tool is not constantly in contact with the silicone surface. The high peak in the distance plot at $t \approx 23$ sec represents loss of contact and results in a subsequent control input arising from the feed-forward force term $\boldsymbol{F}_d$, forcing the tool back onto the surface with an impact (negative peak in measured force).

2) Mesh Impedance Control With Feed-Forward Force: We extend our controller (20) adding an additional feed-forward

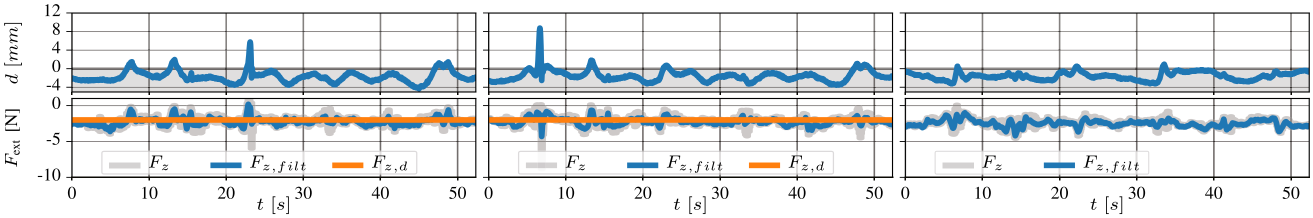


Fig. 8. Left column, experiment III-B1: Cartesian impedance control with feedforward force. The upper row shows the distance d in mm of the tool tip to the surface of the silicon phantom, the lower figure depicts the measured external force F_{ext} in N in z-direction. Middle column, experiment III-B2: Mesh impedance control with feedforward force. Right column, experiment III-B3: Mesh impedance control with negative distance.

force term $\boldsymbol{F}_d \in \mathbb{R}^6$ as in (24) and control the tool to the same Lissajous curve. We set the stiffness in the d-coordinate to zero and command $F_{d,d} = -2$ N. The middle column in Fig. 8 depicts the results. Again, overall the contact force of 2 N is realized (see Table II); however, the tool clearly looses contact to the surface at $t \approx 7$ sec, leading to a subsequent impact collision due to the feed-forward force control.

3) Mesh Impedance Control With Negative Distance: Finally, we use our presented impedance controller (20) and command a negative desired distance of $d_d = -10$ mm, i.e., we control the tool onto the parallel surface $\mathcal{S}_{-10\ \mathrm{mm}}$. We select a lower stiffness k_d for the distance coordinate for this experiment (Table I) and follow the same Lissajous curve (23) (see Fig. 8, right column). In contrast to the previous force control strategies, we show that our control results in almost continuous contact between tool and phantom. At the same time, the controller realizes an interaction force comparable to the two force control experiments (see Table II), without the issue of contact loss and subsequent impact collision.

IV. Discussion and Conclusions

We presented a novel approach to impedance control on arbitrary surfaces, which can be used for robotic ultrasound applications. Our experiments show strong performance in tracking of a complex Lissajous trajectory. We demonstrate our choice of task coordinates (3) allows controlling the robot along complex surfaces including patches of high curvature, where the controller generates high velocities in the robot's joints. By quantifying the relationship between curvature and joint velocities, it may be possible to predict and limit situations of high joint velocities to increase safety in medical scenarios. There were no issues regarding real-time capabilities despite using a complex surface with many faces. The AABB tree structure helps to obtain fast queries. No discretization artefacts were apparent during the experiments. Barycentric interpolation locally smoothens properties on the mesh. Thus, we can utilize discretized representations of arbitrary surfaces, which clearly distinguishes us from previous work on impedance control on analytic surfaces, see e.g. [7].

Furthermore, our results prove the expected behavior when controlling the tool at a negative distance to the surface, i.e. when applying an interaction force to the surface. The right column in Fig. 8 depicts the evolution of the interaction force during tracking of the Lissajous trajectory. Overall, we see an interaction force between tool and silicone phantom which stays in the vicinity of a constant force, with a standard deviation of 0.498 N comparable to classical approaches with feed-forward force control (Table II). Our proposed control method allows to continuously adjust the interaction force by simply adjusting stiffness and set point of our impedance control. This is in contrast to current methods deploying indirect, direct or hybrid position/force control to achieve a desired interaction force [6], [7], which inherently suffer from passivity and stability issues upon contact loss.

Our choice of task coordinates form an integrable system, which can be seen in the results of Section III-A and Fig. 7(a). The parametrization coordinates $\boldsymbol{s}$ are valid on all leaves of the foliation and can be controlled independently from the distance coordinate d. This is an immediate consequence of the foliation structure of the parallel surfaces. This enables us to continuously switch within the foliation, i.e. adjust the interaction force or increase the distance to the surface, while using consistent parametrization coordinates $\boldsymbol{s}$.

Overall, our proposed control scheme is capable of continuous adjustment of the interaction force, and shows comparable performance to classical Cartesian impedance control with an additional feed-forward force when it comes to realization of constant forces (Table II). We bypass some of the downsides inherent to force control, providing a novel way of contact force control for robotic US scanning. The inherent passivity and stability of our controller leads to deterministic and safe behavior on contact loss and guarantees that no uncontrolled and undesired impact onto the tissue surface occurs. Our task coordinates allow to intuitively control the maximum penetration depth of the US probe into the tissue, thus making it possible to limit tissue deformation during US scanning.

A. Future Work

We will further evaluate our controller in more realistic clinically relevant settings for RUS, conducting more in-vitro experiments and consulting clinicians. From a technological point of view, we would like to extent our controller by an outer cascade, adjusting the impedance and its desired task coordinates based on visual servoing to enhance the US image quality. Besides that, using a F/T-sensor at the tool could be utilized for stiffness estimation of the object at the current point of interaction, making it possible to locally update the mesh structure representing the surface, and thus providing new means for dealing with deformable tissue. Additionally, we consider extending our method to also realize out-of-plane rotation around the transducer axis. Furthermore, we will investigate how the presented control can be extended to be applicable as prohibited zone virtual fixture in robotic surgery. Lastly, we plan on enhancing our control methodology by considering solutions for the in Fig. 2(a) introduced focal points of parallel surfaces, and generalizing to arbitrary foliations.

References

[1] K. K. Shung, "Diagnostic ultrasound: Past, present, and future," *J. Med. Biol. Eng.*, vol. 31, no. 6, pp. 371–374, 2011.

[2] W. A. Berg, J. D. Blume, J. B. Cormack, and E. B. Mendelson, "Operator dependence of physician-performed whole-breast US: Lesion detection and characterization," *Radiol.*, vol. 241, no. 2, pp. 355–365, 2006.

[3] K. Li, Y. Xu, and M. Meng, "An overview of systems and techniques for autonomous robotic ultrasound acquisitions," *IEEE Trans. Med. Robot. Bionics*, vol. 3, no. 2, pp. 510–524, May 2021.

[4] N. Hogan, "Impedance control: An approach to manipulation: Part I-theory; Part II-implementation; Part III-applications," *J. Dyn. Syst., Meas. Control*, vol. 107, no. 1, pp. 1–24, 1985.

[5] L. Villani and J. De Schutter, "Force control," in *Springer Handbook of Robotics*. Berlin, Germany: Springer, 2016, pp. 195–220.

[6] M. H. Raibert and J. J. Craig, "Hybrid position/force control of manipulators," *J. Dyn. Syst., Meas., Control*, vol. 103, no. 2, pp. 126–133, 6 1981.

[7] C. Schindlbeck and S. Haddadin, "Unified passivity-based Cartesian force/impedance control for rigid and flexible joint robots via task-energy tanks," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 440–447.

[8] S. Virga *et al.*, "Automatic force-compliant robotic ultrasound screening of abdominal aortic aneurysms," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 508–513.

[9] Z. Jiang, M. Grimm, M. Zhou, Y. Hu, J. Esteban, and N. Navab, "Automatic force-based probe positioning for precise robotic ultrasound acquisition," *IEEE Trans. Ind. Electron.*, vol. 68, no. 11, pp. 11 200–11 211, Nov. 2021.

[10] O. Zettinig *et al.*, "3D ultrasound registration-based visual servoing for neurosurgical navigation," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 12, no. 9, pp. 1607–1619, 2017.

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

[12] Q. Huang, J. Lan, and X. Li, "Robotic arm based automatic ultrasound scanning for three-dimensional imaging," *IEEE Trans. Ind. Informat.*, vol. 15, no. 2, pp. 1173–1182, Feb. 2019.

[13] I. Bonilla *et al.*, "A vision-based, impedance control strategy for industrial robot manipulators," in *Proc. IEEE Int. Conf. Automat. Sci. Eng.*, 2010, pp. 216–221.

[14] D. Song, T. Lee, and Y. J. Kim, "Artistic pen drawing on an arbitrary surface using an impedance-controlled robot," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 4085–4090.

[15] H. Ochoa and R. Cortesão, "Impedance control architecture for robotic-assisted mold polishing based on human demonstration," *IEEE Trans. Ind. Electron.*, vol. 69, no. 4, pp. 3822–3830, Apr. 2022.

[16] S. Kana, K.-P. Tee, and D. Campolo, "Human–robot co-manipulation during surface tooling: A general framework based on impedance control, haptic rendering and discrete geometry," *Robot. Comput.- Integr. Manuf.*, vol. 67, 2021, Art. no. 102033.

[17] A. Albu-Schäffer, C. Ott, and G. Hirzinger, "A unified passivity-based control framework for position, torque and impedance control of flexible joint robots," *Int. J. Robot. Res.*, vol. 26, no. 1, pp. 23–39, 2007.

[18] A. Albu-Schäffer and A. Sachtler, "Quasi-orthogonal foliations of the configuration space: A redundancy resolution approach at position level," 2022, *arXiv:2202.09869*. [Online]. Available: https://arxiv.org/abs/2202.09869

[19] A. Albu-Schäffer, C. Ott, U. Frese, and G. Hirzinger, "Cartesian impedance control of redundant robots: Recent results with the DLR-light-weight-arms," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2003, pp. 3704–3709.

[20] B. Siciliano, *Differential Kinematics and Statics*. London, U.K.:Springer, 2009, pp. 105–160.

[21] M. P. Do Carmo, *Differential Geometry of Curves and Surfaces: Revised and Updated*, 2nd ed., New York, NY, USA: Dover, 2016.

[22] T. Needham, *Visual Differential Geometry and Forms: A Mathematical Drama in Five Acts*. Princeton, NJ, USA: Princeton Univ. Press, 2021.

[23] S. Rusinkiewicz, "Estimating curvatures and their derivatives on triangle meshes," in *Proc. 2nd Int. Symp. 3D Data Process., Visualization Transmiss.*, 2004, pp. 486–493.

[24] T. Bahls, F. A. Fröhlich, A. Hellings, B. Deutschmann, and A. O. Albu-Schäffer, "Extending the capability of using a waterjet in surgical interventions by the use of robotics," *IEEE Trans. Biomed. Eng.*, vol. 64, no. 2, pp. 284–294, Feb. 2017.

[25] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Lévy, *Polygon Mesh Processing*. Boca Raton, FL, USA: CRC Press, 2010.

[26] G. van den Bergen, "Efficient collision detection of complex deformable models using AABB trees," *J. Graph. Tools*, vol. 2, no. 4, pp. 1–13, Jan. 1998.

[27] D. Panozzo, E. Puppo, and L. Rocca, "Efficient multi-scale curvature and crease estimation," in *Proc. Comput. Graph., Comput. Vis. Math.*, 2010, pp. 9–16.

[28] A. Sheffer, E. Praun, and K. Rose, "Mesh parameterization methods and their applications," *Found. Trends Comput. Graph. Vis.*, vol. 2, no. 2, pp. 105–171, 2006.

[29] L. Liu, L. Zhang, Y. Xu, C. Gotsman, and S. J. Gortler, "A local/global approach to mesh parameterization," *Comput. Graph. Forum*, vol. 27, no. 5, 2008, pp. 1495–1504.

[30] U. Seibold, B. Kübler, T. Bahls, R. Haslinger, and F. Steidle, "The DLR Mirosurge surgical robotic robotic demonstrator," in *The Encyclopedia of Medical Robotics*. Singapore: World Scientific, Aug. 2018, pp. 111–142.

[31] A. Jacobson *et al.*, "libigl: A simple C geometry processing library," 2018. [Online]. Available: https://libigl.github.io/

[32] A. Dietrich, C. Ott, and A. Albu-Schäffer, "An overview of null space projections for redundant, torque-controlled robots," *Int. J. Robot. Res.*, vol. 34, no. 11, pp. 1385–1400, 2015.