An Ergodic Approach to Robotic Surface Finishing with Learned Motion Preferences

Stefan Schneyer[1,2], Korbinian Nottensteiner[1], Alin Albu-Schäffer[1,2], Freek Stulp[1], and João Silvério[1]

***Abstract*—Surface finishing is a time-consuming, dangerous task, difficult to automate despite its necessity in many manufacturing processes. Its automation, particularly through robotics, increases productivity and relieves workers from health-critical tasks. However, challenges remain, as automated offline planning tools can result in certain areas being either neglected or overly processed. Ergodic control offers the possibility to cover target probability distributions in an online manner, by taking into account the observed coverage history. However, existing ergodic control approaches provide little flexibility in designing and adapting coverage strategies. Moreover, they come with simplifying assumptions, such as point-based dynamics, which are no longer valid for tasks where the robot is in contact with strongly varying curvatures on non-trivial surface geometries. In this work, we introduce a closed-form ergodic control framework that includes the tool imprint in the system modeling while simultaneously permitting the intuitive transfer of finish strategies, namely preferred motion directions. We build on the Spectral Multiscale Coverage (SMC) approach, augmenting it with a tool imprint model, as well as both target distributions and state-dependent movement directions extracted from human demonstrations. Through evaluations in a surface finishing task using a torque-controlled, 7-DoF, robot arm we show that our approach optimally covers surfaces according to the tool contact area, with robust error convergence.**

***Index Terms*—Ergodic Control, Learning from Demonstration, Motion and Path Planning, Intelligent and Flexible Manufacturing**

I. Introduction

Surface finishing operations, such as polishing, grinding, and sanding, are common in manufacturing and construction. These operations are often performed by human workers due to the difficulty of automating them [1], [2]. Since surface finishing is repetitive and can be hazardous to human health, there is a growing interest in developing robotic systems to perform these processes [3]–[8]. However, as the surface quality is challenging to quantify and the machining effect on the surface is difficult to predict, it is challenging to develop solutions that can achieve a desired surface quality [9], [10]. Furthermore, to establish a robotic system in a craft environment, the system must be easy for craftspeople to program. So, instead of programming the robot with a complex set of rules, the robot should be able to execute surface finishing processes from high-level task descriptions that can be easily specified manually and/or intuitively learned from demonstrations.

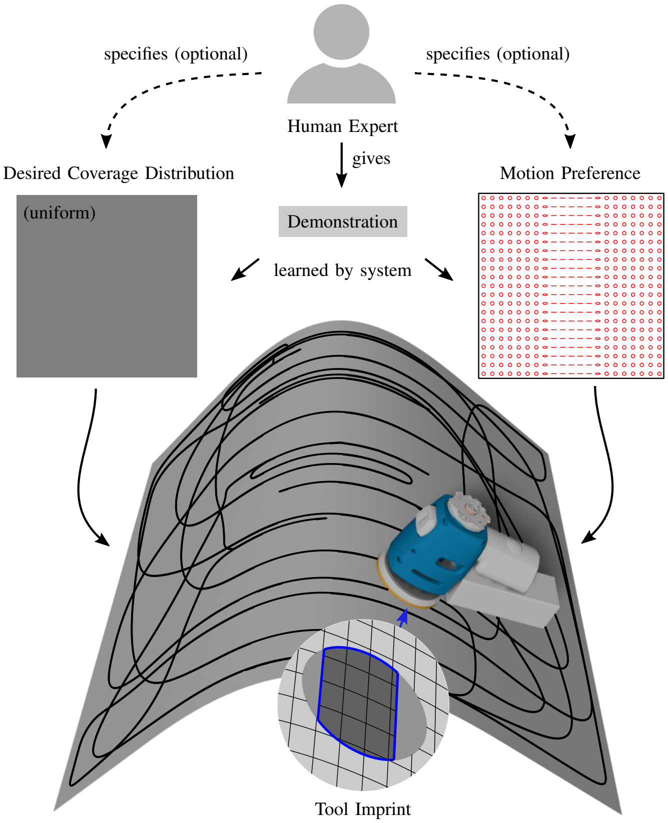


Fig. 1: Robotic surface finishing requires addressing challenges like coverage path planning, inconsistent tool contact on complex surfaces, limited real-time feedback, and material- or geometry-specific finishing requirements. We present an ergodic control framework that addresses these issues. It formulates surface finishing as the problem of covering a desired distribution while incorporating the tool contact area into process modeling and considering preferred finishing directions tailored to material or geometric features. The framework enables robots to learn high-level task descriptions, in the form of desired coverage distributions and motion preferences, from human demonstrations, which it uses to execute the task, enhancing adaptability and interaction in robotic surface finishing.

Problem Statement: In robot surface finishing, coverage planning is typically conducted offline, separating the planning stage from the execution phase. [5], [11], [12]. The robot initiates its task solely after completing the preliminary computations, which provides little flexibility and adaptation of coverage strategies. Moreover, certain areas of the surface may be neglected or overly processed due to the lack of online feedback. In addition, the contact area of the finishing tool varies significantly for complex surfaces. For flat surfaces, the entire tool surface is in contact, whereas for curved surfaces, only a portion makes contact. This makes it difficult to treat

[1] German Aerospace Center (DLR), Institute of Robotics and Mechatronics (RM), Münchener Str. 20, 82234 Weßling, Germany. name.surname@dlr.de

[2] Technical University of Munich (TUM); School of Computation, Information and Technology; Boltzmannstr. 3, 85748 Garching, Germany

a surface, for example, uniformly, since areas with a smaller finishing disk contact area have to be processed with smaller spacing between covering lines than those with a larger tool imprint. Furthermore, human operators may have preferred finishing directions for different materials or geometries. For instance, wood workpieces need to be sanded along the grain or cylindrical geometries along the direction of the main curvature (see Fig. 1) [13].

Contribution: To address the aforementioned limitations, we propose an ergodic-control-based approach for robot surface finishing that i) takes into account the contact area between finishing tool and surface to optimally cover desired distributions and ii) permits the learning of both desired distributions and preferred finishing directions from human demonstrations. Concretely, in this work, we:

- Formulate heavy-duty[1] surface processing tasks for industrial or craft applications as the problem of covering a desired probability distribution that can be learned from human demonstrations and achieved using ergodic control;
- Extend the Spectral Multiscale Coverage (SMC) ergodic control method [14] by incorporating a variable tool contact area that adapts to non-trivial surface geometries;
- Introduce a new approach to compute optimal ergodic control actions while accounting for preferred motion directions;
- Propose a solution to learn preferred motion directions from human demonstrations and incorporate them into ergodic controllers, ensuring that the robot imitates demonstrated behaviors while achieving the desired coverage.

We also show, in both illustrative examples and real robot experiments, that our approach allows for the direct specification of both desired distributions and preferred motions by human experts, with strong optimal coverage guarantees.

The remainder of this paper is structured as follows. Related work is presented in Section II. Section III provides background information on ergodic control and the SMC method. The proposed approach is described in Section IV. The experiments and results are presented in Section V. Section VI discusses the results and provides an outlook for future work. Finally, we conclude the paper in Section VII.

II. Related Work

A. Robotic Surface Finishing

Automating surface finishing is challenging due to the complexity of finding an optimal trajectory for the finishing tool to cover a surface efficiently while achieving a desired surface quality. Various approaches use offline planning to automate the task of covering a surface efficiently, where the process is usually split into two phases [15]–[17]: A preprocessing step, segmenting the surface, followed by a coverage path planning phase, to find an efficient trajectory for each individual partition. For segmentation, some approaches [15], [18] use geometry features like positional coordinates, normal vectors or curvature. Others [16], [17], [19]–[21] use topological simple and monotonic partitions. These approaches achieve resulting segments with easy-to-handle properties like convexity or structures without holes. By finding close-to-isometric two-dimensional surface parametrizations, simple patterns as rasters, Lissajous curves or spirals can be used to generate paths on the surface [12], [22]. This simplifies the complexity of finding an optimal path in which only the parameters of the pattern need to be determined. One notable limitation of offline planning methods is that they cannot easily handle non-uniform distributions of the desired coverage. Furthermore, these methods base their calculations solely on surface coverage modeling, preventing them from adapting to changes or deviations during execution. This limitation precludes the ability to react to unexpected events or modify the robot's behavior during task execution. In this work, we propose ergodic control as a principled approach to address these limitations by formulating coverage problems in heavy-duty surface processing as the tracking of probability distributions. To the best of our knowledge, this is the first work to introduce such a formulation [2].

One key aspect of coverage path planning for surface finishing is the tool imprint (see Fig. 1), which offline approaches already leverage to optimize path efficiency. Indeed, in offline approaches, starting from a raster path pattern, the path can be optimized by considering the tool imprint. As [5], [11], [24] show, the width between coverage path lines can be adjusted to optimize for remaining gaps and overlapping to achieve the largest possible coverage with minimum path length. In this context, the tool imprint model is essential for determining the optimal path width between coverage lines. In our previous work [5], we estimated the tool imprint by calculating the intersection area based on the exact tool geometry and the surface geometry, without relying on simplified assumptions such as constant curvature of the workpiece geometry. Building on this foundation, in this work we integrate the tool imprint model from [5] into an ergodic control formulation to optimize the coverage path online, for a varying tool contact. By combining the imprint modeling with the flexibility and adaptability of ergodic control, we allow for an accurate computation of the covered path, facilitating an efficient surface coverage.

Finally, as previously mentioned, traditional offline methods to surface finishing typically assume uniform desired coverage distributions, limiting their applicability in scenarios requiring non-uniform or more complex coverage patterns. In contrast, ergodic control naturally handles non-uniform desired coverage distributions, addressing this key limitation. With this in mind, we here propose to leverage Learning from Demonstration (LfD), particularly probabilistic solutions (see [25] for a comprehensive review), into our ergodic approach, enabling users to intuitively and flexibly demonstrate complex, non-trivial coverage behaviors as desired distributions, without explicit programming or extensive optimization. To our knowledge, this is the first work introducing ergodic LfD into heavy-

[1] Heavy-duty surface processing — mechanical processes like grinding, sanding, and finishing that remove significant material — relies on precise force, tool contact, and motion, as small variations can alter surface characteristics and overall finish quality.

[2] In [23] we presented preliminary results as a workshop contribution.

duty surface finishing tasks for industrial or craft applications, where we argue that adopting an ergodic approach plays a pivotal role.

B. Ergodic Control

Ergodic control is a technique used to guide the motion of dynamical systems so they exhibit *ergodicity* (see III for details), ensuring they cover reference probability distributions in an online manner. In this context, online refers to methods that calculate an agent's control actions in real-time while the robot is operating and adapt to changes in the goal distribution or system state dynamically, without requiring a dedicated replanning phase. In recent years, ergodic approaches have been applied across a variety of domains, including battery-powered robots [26], shared control for robotic arms in rehabilitation and training [27], search trajectories for mobile robots [28]–[31], active and imitation learning [32], [33], shape estimation using low-resolution sensors [34], higher-dimensional robotic manipulation like peg-in-hole insertion [35], [36], and painting artistic portraits [37]. A key advantage of ergodic control over offline methods in surface finishing is its ability to dynamically adapt to discrepancies between achieved and desired coverage caused by environmental perturbations, unmodeled contacts, or other uncertainties. Furthermore, heuristic coverage strategies, e.g., spiral motions [35] or random walk [33], are known for being inefficient, mainly due to not keeping track of observed coverage. In this work, we argue that ergodic control provides essential properties that address key limitations of existing robot surface finishing formulations, including those discussed in II-A. Particularly, it allows the system to consider the observed coverage history and adapt to changes in the environment, such as interruptions in task execution due to human intervention or changes in the robot's state. Additionally, the desired coverage of the task can be modified online, providing flexibility in task execution. Ergodic control can also handle non-uniform desired coverage distributions, ensuring that specific areas receive the appropriate amount of attention based on the task requirements.

The above-mentioned applications have been facilitated by recent advances in ergodic control methods. In this concern, the Spectral Multiscale Coverage (SMC) method [14], [38], [39] aims to achieve coverage by ensuring that trajectories of multiple agents are ergodic with respect to a prescribed measure. This approach leverages a cost function based on the frequency spectrum of a reference distribution to analytically compute optimal agent actions at each time step, enabling online planning to effectively cover a specified probability distribution. One of the main advantages of SMC is that it allows for prioritizing the coverage of different frequency components. Particularly, as shown, for instance, in [14], [35], [39], SMC prioritizes coverage of lower-frequency regions, effectively ensuring coverage of broader areas before addressing finer details, a relevant property for surface finishing tasks. This property extends even to multi-agent settings [30] where different agents can focus on other aspects, e.g., one agent on low and another on high frequency features, simultaneously.

Another online ergodic control approach is the Heat Equation Driven Area Coverage (HEDAC) method [40]. It introduces an ergodicity-based algorithm based on the stationary heat equation to generate a potential field, which directs agent movement through its gradient. However, compared to the SMC method, it does not allow prioritization in the frequency spectrum, which provides a flexible mechanism for parametrizing surface interaction strategies, such as emphasizing low- or high-frequency components. This added flexibility is especially valuable for surface finishing tasks, where control over the coverage behavior is fundamental for human operators to choose the strategy they prefer.

In [41], the authors propose a method that extends HEDAC for exploring a region of interest using the whole body of the robot, modeled as a collection of virtual agents constrained by the robot's kinematic chain. Control inputs are derived from a consensus among these agents. Moreover, in [42], a closed-loop control method is proposed based on HEDAC for tasks like cleaning curved surfaces. It evaluates actual coverage using vision and represents surfaces with point clouds. The method determines the robot's guiding control input by projecting the agent's position and nearby points from the point cloud onto the tangent plane, then computing the gradient of their potential field. This method, however, does not account for varying tool imprints, which is important in the context of surface finishing. Moreover, vision-based estimation is limited in scope. While it may be suitable for tasks like surface cleaning, it lacks the generality required for a broader range of surface finishing operations. In practical applications, key surface attributes such as smoothness and gloss are often challenging to evaluate with standard vision systems, necessitating specialized sensor feedback or model-based approaches.

In a different direction, Dong *et al.* [29] present a novel method for time-optimal ergodic search, balancing search thoroughness and trajectory execution time. The approach generates efficient search trajectories by formulating the problem as a minimum time problem with an ergodic inequality constraint, demonstrated in both simulations and drone experiments. However, this approach has the disadvantage that it can not be used for online planning. Another ergodic search method is presented in [43], where an iterative linear quadratic regulator (iLQR) algorithm is used for optimizing a given continuous-time trajectory for a long, fixed planning horizon, sharing similar disadvantages with [29].

The topic of computational complexity in the context of ergodic control has also received attention in recent years. While classical approaches [14], [43] have an exponential complexity with respect to the dimension of the search space, in [36] the authors introduce a computationally efficient ergodic search method applicable to both Euclidean space and Lie groups. A kernel-based ergodic metric is used that scales linearly with search space dimensions. Shetty *et al.* [35] also address the limitations of using ergodic control in high-dimensional spaces by proposing the use of tensor trains, a low-rank tensor decomposition technique, to improve computational and storage efficiency.

In summary, various ergodic control methods have been developed and explored in the context of robotics problems and applications. However, many of the traditional approaches have limitations. Particularly, a large number of them generate erratic trajectories that lack distinct motion directions, as the agents solely prioritize the minimization of the coverage loss. Additionally, these methods assume a constant agent imprint, typically a point mass, across the state space, which is inadequate for surface finishing on non-trivial geometries and can reduce efficiency. To address these limitations, our work extends the SMC method by incorporating a varying tool contact area that adapts to non-trivial surface geometries. We also derive a new formulation to compute control actions while taking into account preferred motion directions and propose a Learning from Demonstration approach to learn preferred finishing directions from human operators. Through the combination of these contributions, we aim for a flexible and adaptive solution to the problem of robotic surface finishing.

III. Background

This section provides an overview of ergodicity, ergodic control, and the Spectral Multiscale Coverage (SMC) method. These concepts lay the groundwork for the subsequent sections of this paper.

A. Ergodicity

According to Birkhoff's ergodic theorem, a dynamical system is considered ergodic if and only if the time averages of functions along trajectories are equal to their spatial averages [14], [43]. The time averages refer to each subset $\mathcal{A}$ of the domain space U and are, in each case, the average time spent in $\mathcal{A}$ by an agent's trajectory $\mathbf{x}(t)$ over a time interval T. It is defined as

$$\tau(\mathcal{A},T) := \frac{1}{T}\int_0^T 1_{\mathcal{A}}(\mathbf{x}(t))\mathrm{d}t, \tag{1}$$

where $1_{\mathcal{A}}(\mathbf{x})$ is the indicator function of the subset $\mathcal{A}$, which equals 1 if $\mathbf{x} \in \mathcal{A}$ and 0 otherwise. The spatial average in the ergodic control context is the probability of the agent's position being in the subset $\mathcal{A}$ of the domain space U, which is defined as

$$p(\mathcal{A}) := \int_{\mathcal{A}} p(\mathbf{x})\mathrm{d}\mathbf{x}, \tag{2}$$

where $p(\mathbf{x})$ is the probability density function of the agent's position in the domain space U. For ergodic dynamics, the time average of the agent's trajectory in any subset $\mathcal{A}$ converges to the spatial average, the probability of the agent's position in that subset, as the time interval T approaches infinity. This can be expressed as

$$\forall \mathcal{A} \subset U : \lim_{T\to\infty} \tau(\mathcal{A},T) = p(\mathcal{A}). \tag{3}$$

Figure 2 illustrates the concept of ergodicity in the context of ergodic control. The squared domain U is depicted, showing the agent's trajectory $\mathbf{x}(t)$ over a time interval T as a blue line. For an agent moving at constant velocity, this concept is reflected in the proportion of its trajectory within a given subset, corresponding to the time spent in that subset relative to the total observation period. The spatial average, represented by the probability density function $p(\mathbf{x})$, is shown in black, indicating the desired coverage distribution. The figure includes three example subsets (I, II, III) of the domain space U, located in regions of different probability density, where the proportion of time spent within each subset – represented by the number of blue lines within it – is close to the probability of the agent's position being in that subset given $p(\mathbf{x})$, indicating a high degree of ergodicity.

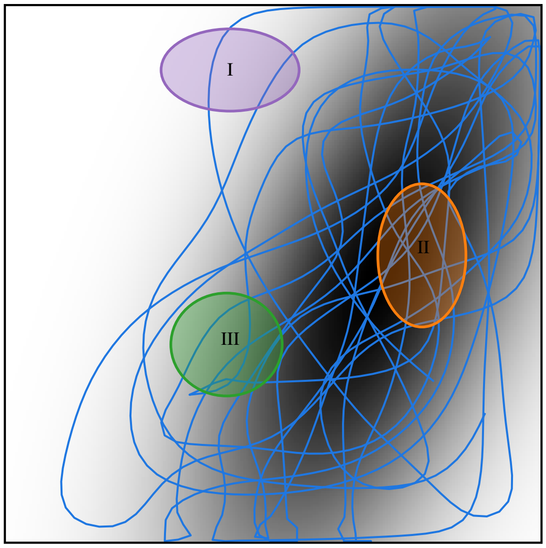


Fig. 2: Illustration of the concept of ergodicity. The agent's trajectory $\mathbf{x}(t)$ through the space is depicted in blue, representing its movement over time. The desired coverage distribution $p(\mathbf{x})$, shown as a grayscale image, indicates where the agent should ideally allocate its time based on a predefined statistical model. Three example subsets (I, II, III) are highlighted to demonstrate that the time spent by the agent in each subset closely aligns with the proportional desired coverage in that subset. This alignment ensures that the agent's behavior reflects the desired statistical properties.

B. Spectral Multiscale Coverage (SMC)

An ergodic controller guides a dynamical system online to exhibit ergodicity by generating control actions that achieve (3). In this work, we leverage the Spectral Multiscale Coverage (SMC) approach [14], [38], [39] to guide the robot in the domain space, where ergodic control aims to achieve a desired coverage, represented by a probability density function $p(\mathbf{x})$. The coverage at position $\mathbf{x}$ and time t of N agents is calculated as

$$c(\mathbf{x},t) = \frac{1}{Nt}\sum_{i=1}^{N}\int_0^t \gamma(\mathbf{x}-\mathbf{x}_i(\tilde{t}))\mathrm{d}\tilde{t}, \tag{4}$$

where, in classical SMC, the agent's imprint $\gamma(\mathbf{x})$ is the Dirac delta function $\delta(\mathbf{x})$, treating the agent as a point mass. This coverage function keeps track of the history of the agents, indicating which areas of the state space have already been covered. A cost function, denoted as $\phi(t)$, quantifies the deviation of the coverage $c(\mathbf{x},t)$ from the target distribution $p(\mathbf{x})$, considering the spectral components of the coverage and the target distribution. It is defined as

$$\phi(t) = \frac{1}{2}\sum_{\mathbf{k}\in\mathcal{K}} \Lambda_{\mathbf{k}}(p_{\mathbf{k}} - c_{\mathbf{k}}(t))^2, \tag{5}$$

where $p_{\mathbf{k}}$ and $c_{\mathbf{k}}(t)$ are the Fourier coefficients of the target $p(\mathbf{x})$ and coverage $c(\mathbf{x},t)$, respectively, for the wave-number array $\mathbf{k} \in \mathcal{K} \subseteq \mathbb{N}^D$ of a rectangular D-dimensional domain U, with $\mathcal{K} = \{0, \ldots, K-1\} \times \ldots \times \{0, \ldots, K-1\}$ and K the number of frequency components. The coefficients are weighted by $\Lambda_{\mathbf{k}}$ to emphasize different frequency components of the distributions. This weighting is commonly chosen as

$$\Lambda_{\mathbf{k}} = \frac{1}{(1 + \|\mathbf{k}\|^2)^{\frac{D+1}{2}}}. \tag{6}$$

where $\|\cdot\|$ is the Euclidean norm. This approach prioritizes low-frequency components over higher ones, leading to agent behavior in which the broader domain is covered initially, with progressively increasing focus on finer details as the process continues. The action $\mathbf{u}_i$ of the SMC motion control is derived from the optimal control formulation

$$\mathbf{u}_i^* = \arg\min_{\mathbf{u}_i} \phi(t + \Delta t) \quad \text{subject to} \quad \|\mathbf{u}_i\| = u_{\max}. \tag{7}$$

The solution to this optimization problem is given by (see details in [14])

$$\mathbf{u}_i^* = -u_{\max} \frac{\mathbf{b}_i(t)}{\|\mathbf{b}_i(t)\|}, \tag{8}$$

where $\mathbf{b}_i(t)$ is defined as

$$\mathbf{b}_i(t) = \sum_{\mathbf{k} \in \mathcal{K}} \Lambda_{\mathbf{k}} (p_{\mathbf{k}} - c_{\mathbf{k}}(t)) \boldsymbol{\nabla} f_{\mathbf{k}}(\mathbf{x}_i(t)), \tag{9}$$

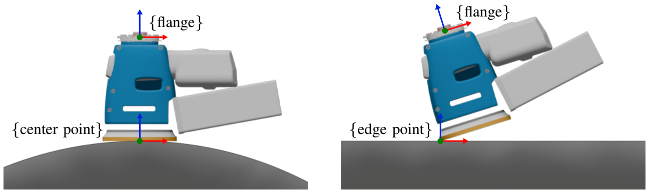
and $f_{\mathbf{k}}$ are the basis functions of the Fourier series decomposition of $\mathbf{x}_i(t)$. Given the control action at each time step (8), ergodic robot motions can be generated either by directly commanding $\mathbf{u}_i^*$ to the system, e.g. in a velocity control scheme, or by integrating it at each time step, using the resulting desired poses as references.

IV. Proposed Approach

We introduce an approach that extends the SMC ergodic control method for surface finishing tasks using a surface parametrization (Section IV-A) by learning desired coverages (Section IV-B), incorporating tool imprints (Section IV-C), and transferring motion preferences from human demonstrations (Section IV-D, Section IV-E). The tool imprint models the effect of the finishing tool on the surface, while the desired coverage and motion preferences guide the robot's motion. With this approach, we aim to improve surface finishing tasks in human-robot collaboration scenarios. Surface finishing applications typically involve a single robotic agent performing the task, as in our experiments. Therefore, we focus on the specific case of a single agent $N = 1$ to simplify the formulations. While the explicit multi-agent setting is omitted, the underlying concept remains applicable to multi-agent scenarios.

A. Preliminaries on Surface Parametrization

To handle arbitrary surface geometries, we follow the approach presented in [5], where we first decompose a given geometry mesh of the workpiece into smaller segments with clearly defined boundaries, see Fig. 6b in [5] for reference. These patches are designed to have well-defined boundaries and no holes, ensuring that each patch can be processed in sequence during surface finishing. As shown in [44], the mesh structure of a segment can be mapped onto a two-dimensional domain space using an as-close-as-possible isometric surface parametrization to simplify the three-dimensional patches. This reduced space is a simplified representation of the robot state space, which provides a concise and tractable framework for controlling the robot's motion on the workpiece surface, allowing for the development and implementation of ergodic control strategies in this reduced-dimensional setting. Particularly, operations such as the Fourier transform can be directly applied in this Euclidean domain. Moreover, the near-isometric nature of the parametrization helps reduce sampling distortions arising from non-uniform vertex densities in the surface mesh, offering advantages over the direct use of point-based three-dimensional meshes [42]. For each simplified segmented surface, which can be any polygon, we compute a minimum-area bounding rectangle that fully encompasses this patch. This bounding rectangle serves as the rectangular domain $U \subset \mathbb{R}^2$ required for applying the Fourier basis functions in the context of SMC ergodic control (see Section III-B). Thus, trajectory generation is initially conducted within the two-dimensional domain and subsequently mapped onto the three-dimensional surface via the inverse mapping. At each surface point, the corresponding normal vector specifies the orientation by constraining the degree of freedom (DoF) to coincide with the surface normal. This condition determines two of the three rotational DoF. At the same time, the remaining DoF, associated with the axial symmetry of the finishing disk, is chosen to mitigate potential collisions of the finishing tool with the environment or robot. Accordingly, a trajectory defined in two dimensions regarding position and velocity is systematically extended into a complete three-dimensional trajectory incorporating both positional and orientational information.



(a) Center point contact

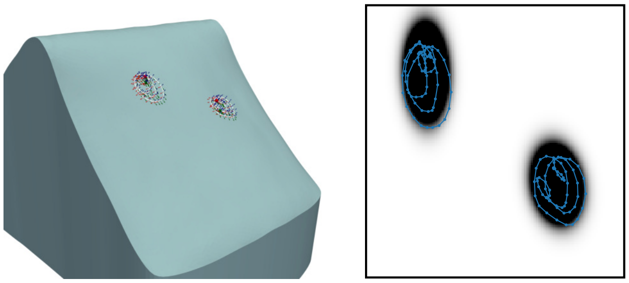
(b) Edge point contact

Fig. 3: Example of tool reference frames for center and edge point contact

The robot's tool center point (TCP) is represented by a single reference frame attached to the tool, which serves as the basis for navigating the robot relative to the workpiece. The placement of this reference frame constitutes a user-defined design choice; however, consistency in its selection is required across both demonstration and execution phases. In our experiments we use the center contact of the finishing disk as tool reference frame except in section V-G, where we use the disk's edge (see Fig. 3).

B. Learning the Desired Distribution from Position Data

We propose to specify surface finishing tasks by learning desired coverage distributions from human demonstrations.



(a) Demonstration trajectories (b) Learned distribution

Fig. 4: The figure shows, for a surface finishing task, the human demonstration trajectories of the finishing disk center point with its tool coordinate system on the workpiece (a) and the learned GMM in the two-dimensional surface parametrization (b). The grayscale image illustrates the probability density of the GMM distribution, with the demonstration paths represented by blue lines.

Specifically, position data collected from the demonstration process can be utilized to infer a probability distribution that serves as the reference distribution for the ergodic controller. This is achieved using a Gaussian mixture model (GMM), which is learned from the data via the expectation-maximization (EM) algorithm [45]. The maximization problem involves finding the optimal set of parameters $\boldsymbol{\theta}$ that maximizes the log-likelihood function as

$$\log p(\mathcal{X} \mid \boldsymbol{\theta}) = \sum_{m=1}^{M} \log \left(\sum_{c=1}^{C} \pi_c \mathcal{N}(\mathbf{x}^{(m)} | \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \right), \quad (10)$$

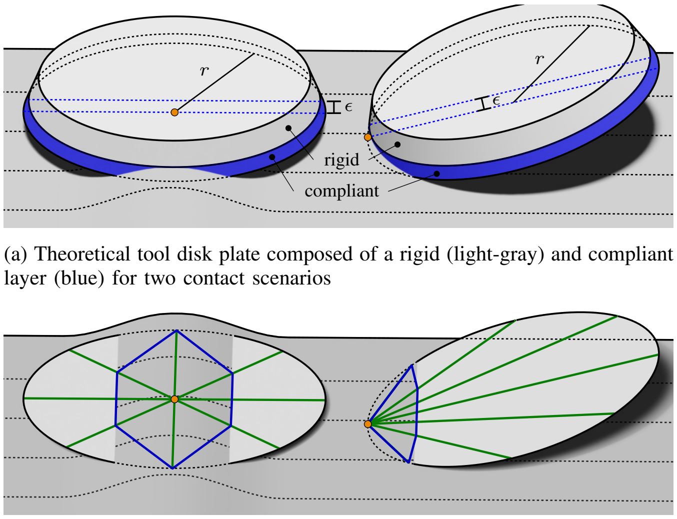
$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \log p(\mathcal{X} \mid \boldsymbol{\theta}), \quad (11)$$

where $\boldsymbol{\theta} = \{\pi_c, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\}_{c=1}^{C}$ represents the parameters of the GMM, π_c are the mixing coefficients, $\mathcal{N}(\mathbf{x}^{(m)} | \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$ is the Gaussian distribution with mean $\boldsymbol{\mu}_c$ and covariance $\boldsymbol{\Sigma}_c$, and M is the number of data points. Here, $\mathcal{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(M)}\}$ denotes the dataset consisting of the two-dimensional position data $\mathbf{x}^{(i)} \in \mathbb{R}^2$ from the human demonstration, where $\mathbf{x}^{(i)}$ represents the contact point coordinates on the domain space U. The GMM provides a probabilistic framework that allows the robot to infer the task requirements and perform it autonomously by following the learned distribution, which is given as

$$p(\mathbf{x}) := p(\mathbf{x} \mid \boldsymbol{\theta}^*). \quad (12)$$

An example of this approach is shown in Fig. 4, where a human demonstration first identifies two locations on the surface. The probabilistic model is then inferred from the corresponding position data in the two-dimensional state space. We argue that this model captures the key aspects of the surface finishing process, including the specific areas to be processed and the required intensity of the finishing. A higher probability density indicates that more time should be allocated to a region, resulting in increased processing in that area. For execution, an ergodic controller is used to guide the robot online along the distribution, optimizing ergodicity.

Note that the reference distribution can also be predefined, as in [14], for instance, to achieve uniform surface finishing. In this scenario, the reference distribution is manually configured to accomplish a particular task. However, this can be either too rigid or impractical for users unfamiliar with probability theory. Our LfD-based approach aims to address these limitations.



(a) Theoretical tool disk plate composed of a rigid (light-gray) and compliant layer (blue) for two contact scenarios

(b) Corresponding tool disks with the imprint on the surface

Fig. 5: Illustration of the rays (green lines) inside the tool disk used to approximate the tool imprint (blue polygon) of a tool reference point (orange) on the workpiece surface (indicated by dashed lines in gray) for two different contact scenarios.

C. Ergodic Control with Imprint Model

In practice, the effect of the tool during surface finishing is influenced by multiple factors, including the applied force, the velocity of the tool contact point trajectory, and the geometry of the freeform workpiece around the tool contact. In particular, the geometry of the freeform workpiece around the tool contact plays a major role. It determines which points are in contact with the surface and how the pressure is distributed on the contact area (see Fig. 1).

To incorporate the contact between the tool and the workpiece within the framework of ergodic control, we model the finishing tool as a flat disk with radius r (see Fig. 5). Taking into account the flexibility of the disk and its ability to conform to the surface, we introduce an elasticity distance ϵ, as proposed for offline planning purposes in [5]. This distance represents the tool's compliance and defines how far the tool and abrasive material can deform under the process forces to conform to the surface. To estimate the contact area between the finishing disk and the workpiece, we first slightly move the disk inward, along its height direction, by a small distance ϵ toward the surface. This shifts the disk slightly into the workpiece. Next, we determine the area where this shifted disk intersects with the workpiece geometry, which serves as an approximation of the contact region. To determine the shape of this contact area, we emit rays starting from the shifted contact point $\mathbf{p}$, which acts as the origin, within the plane of the disk. These rays are sampled at equal angular intervals, such that the angles between consecutive rays are evenly spaced around the origin. We can express this as

$$\mathbf{r}_i = \mathbf{p} + s \begin{bmatrix} \cos \kappa_i \\ \sin \kappa_i \\ 0 \end{bmatrix}, \quad s \geq 0, \qquad (13)$$
$$\kappa_i = \frac{2\pi i}{R}, \quad i \in \{0, 1, \ldots, R-1\},$$

in coordinates where x- and y-directions are aligned with the tool disk plane and R is the number of rays sampled. The rays are limited to the boundaries of the tool disk; for example, when the contact point is at the disk center, the ray parameter s is restricted by the disk radius r, such that $r \geq s$. For each ray, we compute its intersection point with the workpiece surface. If no intersection occurs within the limits of the tool disk, the ray's endpoint at the disk boundary is used instead. The resulting ordered set of points defines the vertices of a polygon that approximates the tool imprint. Consequently, all points within the polygon are considered part of the tool contact region. Figure 5 visualizes our tool imprint modeling approach.

We modify the coverage calculation in the SMC method to ensure that the effect of the tool contact on the surface is taken into account. Instead of using a constant agent's imprint, we model it as a varying function that depends on the tool reference position on the surface and, consequently, the agent's state. Additionally, we use a discrete FFT-based computation for $p_\mathbf{k}$ and $c_\mathbf{k}(t)$ instead of the analytical Fourier transform approach, as, for example, in [35], [39]. The discrete FFT does not compromise accuracy compared to [14], as the original method also relies on a finite number of Fourier basis functions, inherently approximating the coverage signal. The key advantage of this change lies in its ability to handle arbitrary input distributions and adaptable agent imprint functions. Recalling (4), the coverage calculation can then be expressed as

$$c(\mathbf{x}, t) = \frac{1}{t} \int_0^t \gamma_{\mathbf{x}(\tilde{t})}(\mathbf{x} - \mathbf{x}(\tilde{t}))\mathrm{d}\tilde{t}, \qquad (14)$$

where $\gamma_\mathbf{c}(\mathbf{x}) \in \mathbb{R}$ denotes the tool imprint, which is varying according to different tool positions $\mathbf{c}$ in the robot state space (for this case $\mathbf{c} := \mathbf{x}(\tilde{t})$). Let $\mathcal{I}(\mathbf{c})$ represent the subset of the surface in contact with the tool at a reference point $\mathbf{c}$. An illustrative example is shown in Fig. 5, where the region enclosed by the blue polygon approximates $\mathcal{I}(\mathbf{c})$. The reference point $\mathbf{c}$ is depicted in orange in Fig. 5 and corresponds to the origin of the center or edge point frames in Fig. 3. We then define the agent's tool imprint as

$$\gamma_\mathbf{c}(\mathbf{x}) = \alpha \, 1_{\mathcal{I}(c)}(\mathbf{x}). \qquad (15)$$

The parameter α serves as a scaling factor, enabling the formulation to operate in two distinct modes:

1) Constant Pressure Mode ($\alpha = 1$): In this mode, the applied normal force is proportional to the contact area, maintaining a constant pressure across the surface.
2) Constant Force Mode: In this mode, a constant total force is applied, but its distribution varies depending on the contact area. In this context, the imprint function can be mathematically represented as a distribution that sums to one, thus we have

$$\alpha = \frac{1}{\int_{\mathbf{x} \in \mathcal{I}(\mathbf{c})} 1 \mathrm{d}\mathbf{x}}. \qquad (16)$$

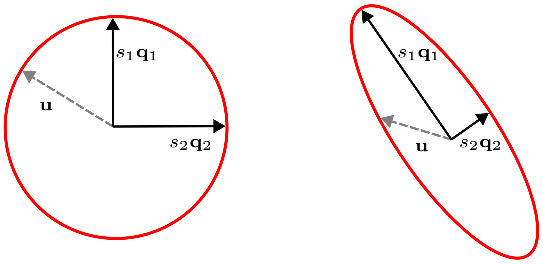


Fig. 6: Illustration of the circular and elliptical constraint for the velocity vector with the scaled eigenvectors ($s_1\mathbf{q}_1$, $s_2\mathbf{q}_2$) of the matrix $\mathbf{\Gamma}$ represented as arrows along the major and minor axes of the ellipse. The dotted arrow $\mathbf{u}$ exemplifies one possible velocity vector that satisfies the constraint, although any vector along the boundary of the ellipse is also valid.

By incorporating the tool imprints into the coverage calculation, we can optimize the surface coverage more accurately, considering the specific characteristics of the surface and the tool.

D. Constrained Ergodic Optimization Problem

The goal of the ergodic control SMC method is to minimize the cost function (5) by computing optimal control actions that maximize its decrease in the next time step. We recall from (7) that the ergodic control action in [14] is derived by solving the optimization problem

$$\begin{aligned} \min_{\mathbf{u}(t)} \quad & \phi(t + \Delta t), \\ \text{s.t.} \quad & \sqrt{\mathbf{u}(t)^\intercal \mathbf{u}(t)} = u_{\max}, \end{aligned} \qquad (17)$$

where the norm of the control action $\mathbf{u}$(t) is constrained to a maximum value $u_{\max}$ regardless of the direction. In further derivations, we drop the dependence on t for simplicity, unless required in specific derivation steps. We extend this approach by modifying the constraint definition using an invertible and positive definite weighting matrix $\mathbf{\Gamma}$, which allows the different velocity directions to be constrained in varying ranges as

$$\sqrt{\mathbf{u}^\intercal \mathbf{\Gamma}^{-1} \mathbf{u}} = 1. \qquad (18)$$

We further assume $\det(\mathbf{\Gamma}) = 1$ without loss of generality since scaling $\mathbf{\Gamma}$ is equivalent to scaling the right term in (18). Moreover, we assume $\mathbf{\Gamma}$ to be symmetric, similarly to a covariance matrix, thus in the two-dimensional case (18) defines an elliptical constraint. This ellipse can be represented using the eigenvalue decomposition as

$$\mathbf{\Gamma} = \mathbf{Q} \begin{bmatrix} s_1^2 & 0 \\ 0 & s_2^2 \end{bmatrix} \mathbf{Q}^{-1}, \qquad (19)$$

where $\mathbf{Q} = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 \end{bmatrix}$ is the rotational part that defines the orientation of the ellipse, and s_1^2 and s_2^2 are the eigenvalues that scale the ellipse along the principal axes by the factors s_1 and s_2 (see Fig. 6). This decomposition allows us to understand the influence of the matrix $\mathbf{\Gamma}$ on the control input $\mathbf{u}$, providing a precise geometric interpretation of the constraints imposed on the system. The two eigenvectors, $\mathbf{q}_1$ and $\mathbf{q}_2$, of the matrix represent the directions with weakest and strongest motion constraints. By visualizing the constraint as an ellipse (Fig. 6), we can see that each point on its surface represents a possible velocity vector $\mathbf{u}$ that satisfies the constraint. The constraint is a circle if the matrix is the identity matrix, which is one special case of the general formulation.

The general optimization problem considering (18) can then be written as

$$\begin{aligned} \min_{\mathbf{u}} \quad & \Phi(t+\Delta t) \\ \text{s.t.} \quad & \sqrt{\mathbf{u}^\intercal \boldsymbol{\Gamma}^{-1}\mathbf{u}} = 1. \end{aligned} \tag{20}$$

where we use $\Phi(t) := t^2\phi(t)$ for notational convenience following (16) from the original approach [14], which simplifies the computation of derivatives with respect to time. Notably, this substitution does not alter the optimization problem. Assuming continuity of the cost concerning time, as in [14], the Taylor expansion is used to approximate the cost function for subsequent points in time as

$$\Phi(t+\Delta t) \approx \Phi(t) + \dot{\Phi}(t)\Delta t + \frac{1}{2}\ddot{\Phi}(t)\Delta t^2, \tag{21}$$

where its derivatives with respect to time are given by

$$\dot{\Phi}(t) = \sum_{\mathbf{k}\in\mathcal{K}} \Lambda_{\mathbf{k}} S_{\mathbf{k}}(t) W_{\mathbf{k}}(t), \tag{22}$$

$$\ddot{\Phi}(t) = \sum_{\mathbf{k}\in\mathcal{K}} \Lambda_{\mathbf{k}} \left(W_{\mathbf{k}}(t)\right)^2 + \sum_{\mathbf{k}\in\mathcal{K}} \Lambda_{\mathbf{k}} S_{\mathbf{k}}(t) \dot{W}_{\mathbf{k}}(t) \tag{23}$$

$$= \sum_{\mathbf{k}\in\mathcal{K}} \Lambda_{\mathbf{k}} \left(W_{\mathbf{k}}(t)\right)^2 + \mathbf{b}(t)^\intercal \mathbf{u}(t), \tag{24}$$

$$S_{\mathbf{k}}(t) := (p_{\mathbf{k}} - c_{\mathbf{k}}(t))t, \tag{25}$$

$$W_{\mathbf{k}}(t) := \dot{S}_{\mathbf{k}}(t). \tag{26}$$

Note that we use lowercase letters for vectors such as $\mathbf{b}$ here for reasons of consistency of our notation, whereas capital letters are used in the original work [14]. Given these terms, we can simplify the optimization problem, where the optimal control input $\mathbf{u}^*$ is then obtained by solving

$$\begin{aligned} \mathbf{u}^* = \arg\min_{\mathbf{u}} \quad & \mathbf{b}(t)^\intercal \mathbf{u} \\ \text{subject to} \quad & \sqrt{\mathbf{u}^\intercal \boldsymbol{\Gamma}^{-1}\mathbf{u}} = 1. \end{aligned} \tag{27}$$

We can solve this constrained optimization problem as shown in Appendix A using Lagrange multipliers, where the optimal control input $\mathbf{u}^*$ is given by

$$\mathbf{u}^* = -\frac{\boldsymbol{\Gamma}\mathbf{b}(t)}{\sqrt{\mathbf{b}(t)^\intercal \boldsymbol{\Gamma}\mathbf{b}(t)}}. \tag{28}$$

The matrix $\boldsymbol{\Gamma}$ introduces an additional degree of freedom to influence the control output of the system. Note that if $\det(\boldsymbol{\Gamma}) \neq 1$, the solution (28) would be the same up to a scaling factor (see Appendix B for details). By specifying this matrix with preferred directions, we can set priorities for the robot's motion while simultaneously achieving a desired coverage distribution with the ergodic controller.

As $\mathbf{u}$ is constrained to the elliptical shape, the constraint enables the robot to exhibit varying magnitudes of velocity in different directions, with higher velocities in regions where the ellipse constraint is stretched and lower velocities in areas where the ellipse constraint is compressed. However, to achieve a desired velocity magnitude $u_{\max}$ in the real system, the velocity vector that is commanded to the robot $\hat{\mathbf{u}}$ can be computed as a re-scaled version of $\mathbf{u}^*$ as

$$\hat{\mathbf{u}} = u_{\max}\frac{\mathbf{u}^*}{\|\mathbf{u}^*\|}. \tag{29}$$

Importantly, this re-scaling process does not alter the prioritization of directions exhibited by $\mathbf{u}^*$. The optimization (20) continues to favor movements in the preferred directions, which means that these directions are more likely to be chosen, even if they are not applied with increased velocity. Equation (29) merely ensures that the final velocity vector attains the desired magnitude while maintaining the directional preferences. As such, this process enables the system to maintain its preferred motion directions while ensuring a constant speed.

The provided examples for the derivation focus on the two-dimensional case, which suffices for surface finishing purposes. However, the derivation of motion preference using the matrix $\boldsymbol{\Gamma}$ is also applicable in n-dimensional spaces.

E. Motion preference transfer from human demonstrations

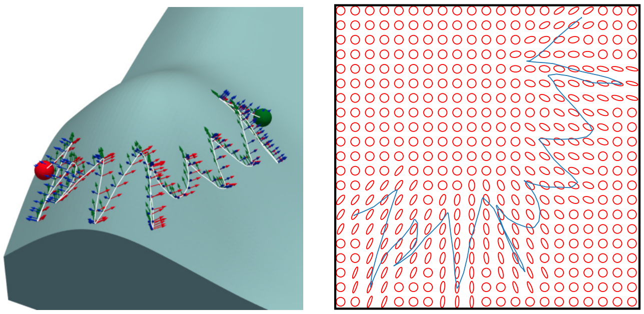
Building on the constrained ergodic optimization problem, we propose to extract motion preferences from human demonstrations by learning a state-dependent $\boldsymbol{\Gamma}$ and transferring them to the robot. Let us denote a training dataset with human demonstrations as

$$\mathcal{D} = \{(\mathbf{x}^{(1)}, \mathbf{u}^{(1)}), (\mathbf{x}^{(2)}, \mathbf{u}^{(2)}), \ldots, (\mathbf{x}^{(M)}, \mathbf{u}^{(M)})\} \tag{30}$$

where each element contains a time step with the robot state $\mathbf{x}^{(i)}$ on the surface parametrization space and its corresponding velocity direction $\mathbf{u}^{(i)} = \frac{\dot{\mathbf{x}}^{(i)}}{\|\dot{\mathbf{x}}^{(i)}\|}$. We model the velocity vectors as random variables with mean $\boldsymbol{\mu}_u \in \mathbb{R}^2$ and covariance $\boldsymbol{\Sigma}_u \in \mathbb{R}^{2\times 2}$. We assume that surface finishing motions on a limited space are approximately closed paths, and therefore the average velocity in a set of demonstrations is close to zero, i.e. $\boldsymbol{\mu}_u \approx 0$, see Appendix C for details. Also, due to the periodic nature of surface finishing motions, passing through the same point multiple times with opposing velocities is common, making the average velocity at each point approach zero over time. We therefore turn to the covariance matrix of the observed velocities as the descriptor of the motion pattern distribution. One of the advantages of considering the covariance is that its computation is invariant to the sign of the velocity vector, capturing the trend of the motion instead. Indeed, under $\boldsymbol{\mu}_u \approx 0$, the maximum likelihood estimation for the covariance of $\mathbf{u}$ in $\mathcal{D}$ is given by

$$\boldsymbol{\Sigma}_u = \mathbb{E}(\mathbf{u}\mathbf{u}^\intercal) = \frac{1}{M}\sum_{i=1}^{M} \mathbf{u}^{(i)}\mathbf{u}^{(i)\intercal}. \tag{31}$$

We thus propose to learn a state-dependent $\boldsymbol{\Gamma}$ by identifying a covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{u}}$ from the dataset for any given state $\mathbf{x}$. In this scenario, we seek to define a function $(\mathbf{x}, \mathcal{D}) \mapsto \boldsymbol{\Sigma}_u$. Consequently, we derive a function that is fully defined over the entire state space, thereby enabling us to accurately predict



(a) Demonstration trajectory

(b) Learned $\mathbf{\Gamma}$-map with trajectory

Fig. 7: The figure shows the workpiece with human demonstration of the surface finishing task (a), alongside the learned $\mathbf{\Gamma}$-map with the trajectory mapped onto the surface parametrization (b).

outputs for any given input. For this, we follow a non-parametric approach and use kernel regression [45], [46] to predict covariance matrices locally for any state $\mathbf{x}$ using a distance-based influence for each datapoint in the neighborhood of $\mathbf{x}$. We use a radial basis function (RBF) kernel, formulated as

$$k(\mathbf{x}, \mathbf{x}^{(i)}) := \beta \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}^{(i)}\|^2}{2l^2}\right) \tag{32}$$

where l is the length scale of the kernel and β is a scaling value that makes sure that the trace of the weighted sum of matrices is preserved (see Appendix D for more details), i.e., that the resulting matrices do not scale arbitrarily. The learned covariance matrix for an unseen state $\mathbf{x}$ can then be determined as

$$\mathbf{\Sigma}_u(\mathbf{x}) = \sum_{i=1}^{M} k(\mathbf{x}, \mathbf{x}^{(i)}) \mathbf{u}^{(i)} \mathbf{u}^{(i)\intercal}. \tag{33}$$

Despite $\mathbf{u}^{(i)} \mathbf{u}^{(i)\intercal}$ being rank deficient, we assume l to be large enough that $\mathbf{\Sigma}_u(\mathbf{x})$ never is, due to the influence of neighboring datapoints. We introduce a weighting scheme to ensure that positions away from data gradually converge to an isotropic covariance matrix, namely a scaled version of the identity. In such regions, no demonstration data is available; hence, no motion direction should be preferred. The constraint matrix is then given by

$$\mathbf{\Gamma}(\mathbf{x}) = \lambda \mathbf{\Sigma}_u(\mathbf{x}) + (1-\lambda)\frac{1}{2}\mathbf{I}, \tag{34}$$

where $\mathbf{I}$ is the identify matrix and λ a weighting parameter that indicates how far the robot state $\mathbf{x}$ is away from the demonstration data and is defined as

$$\lambda = \max_{i \in \{1,\dots,M\}} k(\mathbf{x}, \mathbf{x}^{(i)}). \tag{35}$$

The weighted sum of matrices (34) preserves the trace of the matrix as shown in Appendix D. Ensuring a constant trace is important to guarantee that the matrices are unique and do not scale arbitrarily. We refer to the function $\mathbf{x} \mapsto \mathbf{\Gamma}$, which assigns a specific $\mathbf{\Gamma}$ to each unique state in the system, as $\mathbf{\Gamma}$-map. An example of a learned $\mathbf{\Gamma}$-map is shown in Fig. 7.

Note that $\mathbf{\Gamma}(\mathbf{x})$ can also be defined in a manual, handcrafted manner, where constraints are explicitly chosen based on domain knowledge or expert insight instead of being learned. For instance, a region with a constant constraint can be constructed, and furthermore, the mixture term in (34) can be used to get a gradual fading transition. In Section V we extensively evaluate both learned and manually defined $\mathbf{\Gamma}$-maps.

Algorithm 1 Ergodic Control with Imprint Model and Preferred Motion Directions

```
Input: Demonstration datasets $\mathcal{X}$ and $\mathcal{D}$, initial state $\mathbf{x}_0$,
    stopping criteria $\eta$, time step $\Delta t$
Define desired distribution by Learning from Demonstration (12) using $\mathcal{X}$ or manually
Learn motion covariance $\mathbf{\Sigma}_u(\mathbf{x})$ from the dataset $\mathcal{D}$ using kernel regression (33)
Compute $\mathbf{\Gamma}$-map via (34) or define it manually
Initialize state $\mathbf{x} \leftarrow \mathbf{x}_0$ and time $t \leftarrow 0$
while $\phi(t) > \eta$ do
    Compute $c(\mathbf{x}, t)$ as in (14) with imprint model
    Compute gradient $\mathbf{b}(t)$ by (9) using FFT coefficients
    Compute $\mathbf{\Gamma}(\mathbf{x})$ for current $\mathbf{x}$
    Compute control input $\hat{\mathbf{u}}$ using (29)
    Compute ergodic cost $\phi(t)$ (5)
    Update state (in simulation: $\mathbf{x} \leftarrow \mathbf{x} + \hat{\mathbf{u}} \cdot \Delta t$)
    Update time $t \leftarrow t + \Delta t$
end while
```

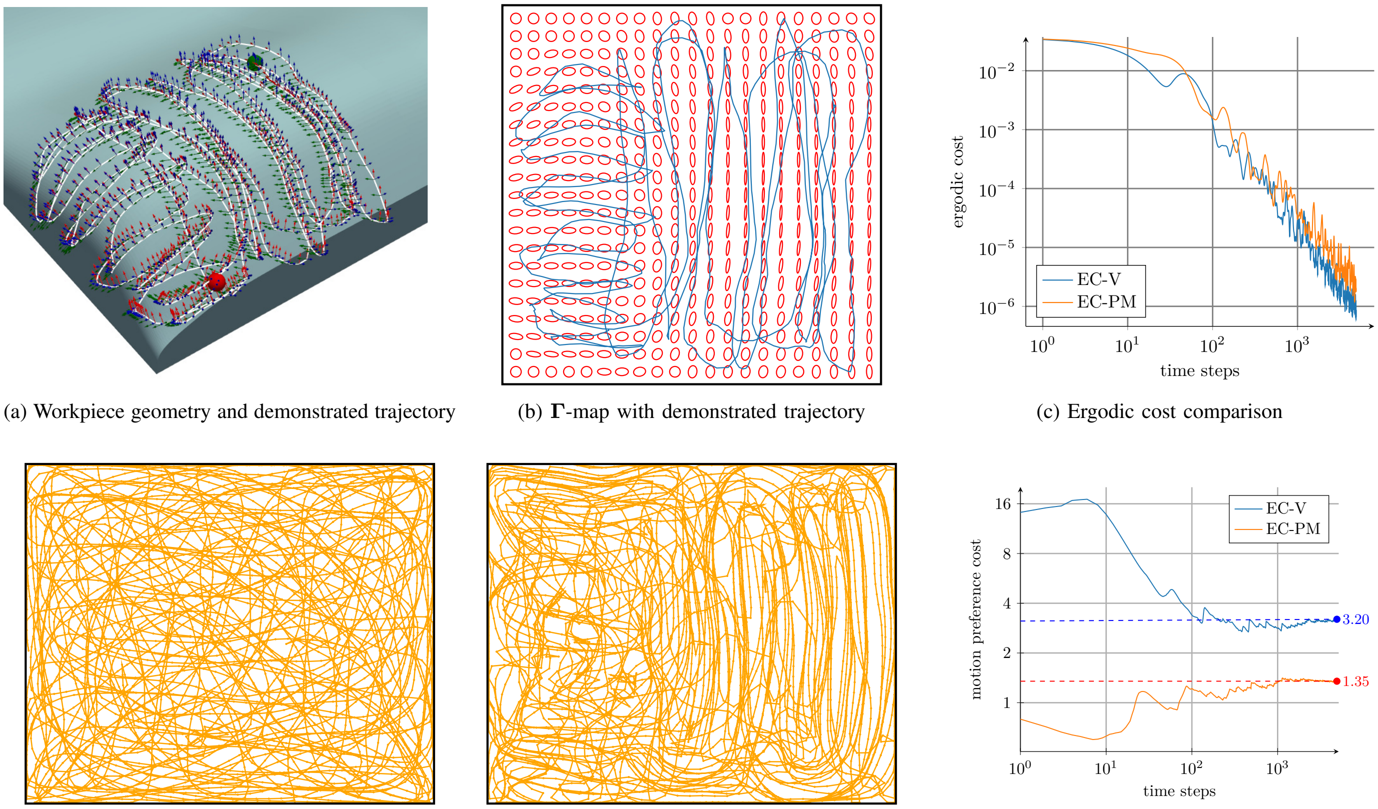
Our overall approach brings together the contributions in Sections IV-B to IV-E into a framework that leverages human demonstrations and tool imprint models to facilitate the transfer of surface finishing skills from humans to robots. The steps of the proposed approach are outlined in Algorithm 1.

V. Experiments

In this section, we evaluate the proposed approach on both simulated and real surface finishing tasks and compare it to the original SMC method [14], which we refer to as *vanilla* ergodic control (EC-V). We show the effectiveness of incorporating preferred motion directions (Section V-A, Section V-B) and tool imprints (Section V-C, Section V-D). Furthermore, we demonstrate the benefits of using the proposed approach in a real-world scenario with a robotic manipulator (Section V-E, Section V-F, Section V-G).

A. Comparison of Vanilla Ergodic Control and Ergodic Control with Preferred Motion Patterns

In the first experiment, we compare vanilla ergodic control (EC-V) with the proposed approach incorporating preferred motion directions (EC-PM) learned from human demonstrations. In this setting, we focus on the motion preference aspect only, and do not yet consider the imprint model. Thus, the agent's imprint is a Dirac delta function. We use a uniform coverage as the desired distribution for the ergodic controller. For all experiments, we have a rectangular domain and use 15 frequency components for the SMC method (see (5)), such that $\mathcal{K} = \{0, \dots, 14\} \times \{0, \dots, 14\}$.



(a) Workpiece geometry and demonstrated trajectory

(b) $\mathbf{\Gamma}$-map with demonstrated trajectory

(c) Ergodic cost comparison

(d) Actual coverage of approach EC-V

(e) Actual coverage of approach EC-PM

(f) Motion preference cost comparison

Fig. 8: Comparison of Vanilla Ergodic Control (EC-V) and Preferred Motion Incorporated Ergodic Control (EC-PM) for a workpiece (a) with a uniform desired distribution. The EC-PM uses the $\mathbf{\Gamma}$-map (b) learned from demonstration. The figures (d) and (e) show the resulting coverage maps of the experiments and the diagram (c) depicts the ergodic costs over time and (f) the motion preference costs, which measure the correspondence of the trajectories with the specified motion preference.

The trajectory depicted in Fig. 8a was demonstrated by a human operator using the 7-DoF DLR SARA robot through kinesthetic teaching. The operator ensures that desired motion directions are demonstrated on the workpiece. The recorded data of the contact points are transformed to the two-dimensional state space. The $\mathbf{\Gamma}$-map illustrated in Fig. 8b was learned by kernel regression from this demonstration data, as explained in Section IV-E. Figure 8 further shows the agent's trajectory for EC-V (Fig. 8d), our approach using the learned $\mathbf{\Gamma}$-map (Fig. 8e), and the ergodic cost (Fig. 8c) for the two approaches. The path of the agent is plotted in orange for step $5 \cdot 10^3$. The ergodic cost is plotted over time for both approaches, with vanilla ergodic control (EC-V) shown in blue and ergodic control with preferred motion directions (EC-PM) in orange.

To quantitatively assess how well a given trajectory $\mathbf{x}(t)$ corresponds to the desired motion preference $\mathbf{\Gamma}(\mathbf{x})$, we propose using the squared Mahalanobis distance of velocities over time, defined as

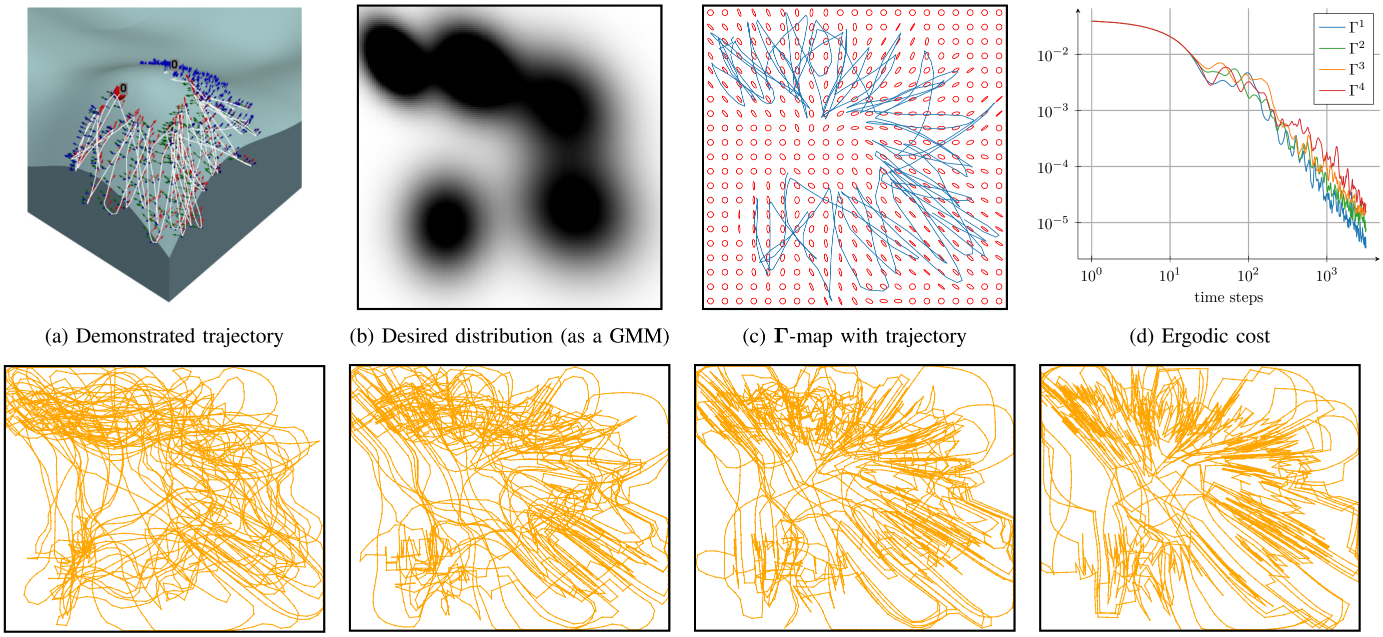
$$d(\mathbf{\Gamma}(\mathbf{x}), \mathbf{x}(t)) = \int_{t=0}^{t_{\text{final}}} \mathbf{u}(t)^{\intercal} \mathbf{\Gamma}(\mathbf{x}(t))^{-1} \mathbf{u}(t)\, dt. \tag{36}$$

where $\mathbf{u}(t)$ denotes the velocity at time t. This metric effectively measures the degree to which the trajectory's velocities comply with the motion preference encoded by $\mathbf{\Gamma}(\mathbf{x})$. We employed this metric to compare the performance of the two approaches (EC-V and EC-PM). The results, illustrated in Fig. 8f, demonstrate quantitatively that EC-PM achieves a lower average Mahalanobis distance than EC-V, indicating a better correspondence to the specified motion preferences.

The results in Fig. 8 show that the proposed approach effectively covers the desired distribution while prioritizing the demonstrated motion directions. Compared to the proposed approach, EC-V achieves a slightly lower ergodic cost for the same number of time steps. This is expected because the EC-PM approach minimizes the ergodic cost by including additional directional constraints on movement, whereas EC-V minimizes the cost function without such constraints. Moreover, the cost function does not measure the fulfillment of these constraints.

B. Learning Distribution and Preferred Motion from Data

In a subsequent experiment, we demonstrate the capability of EC-PM to imitate a human-demonstrated surface finishing task by learning both the desired coverage distribution and the preferred motion directions directly from human demonstrations. The experiment is conducted using a workpiece with a convex, curved surface (Fig. 9a). Kinesthetic teaching is used to demonstrate the target probability distribution of the task together with the desired motion patterns. The reference distribution is encoded by a GMM (Fig. 9b), that models the distribution of the end-effector position mapped onto the surface parametrization of the workpiece, as described in Section IV-B. We choose $C = 5$ to achieve a practical balance between overfitting and underfitting, but its optimal value may



(a) Demonstrated trajectory

(b) Desired distribution (as a GMM)

(c) $\mathbf{\Gamma}$-map with trajectory

(d) Ergodic cost

(e) Coverage map with $\mathbf{\Gamma}$

(f) Coverage map with $\mathbf{\Gamma}^2$

(g) Coverage map with $\mathbf{\Gamma}^3$

(h) Coverage map with $\mathbf{\Gamma}^4$

Fig. 9: Learning distribution and preferred motion from demonstration data. The figure shows the human demonstration of the surface finishing task (a), the learned desired distribution using a GMM (b), the learned $\mathbf{\Gamma}$-map with the trajectory of the robot (c), the ergodic cost over time (d), and the coverage maps for different strengths of the $\mathbf{\Gamma}$ covariance matrix (e-h) for the EC-PM method.

vary depending on the dataset and application. Additionally, we learn the $\mathbf{\Gamma}$-map by kernel regression to imitate the motion directions shown in the demonstration (Fig. 9c).

We evaluate the ergodic control result across four different scenarios, comparing the execution using $\mathbf{\Gamma}$ (we drop the argument $\mathbf{x}$ for simplicity) extracted from data (Fig. 9e) to scaled versions of itself where the constraints are accentuated by taking it to different powers $\mathbf{\Gamma}^n$. Particularly we evaluated $\mathbf{\Gamma}^2$ (Fig. 9f), $\mathbf{\Gamma}^3$ (Fig. 9g), and $\mathbf{\Gamma}^4$ (Fig. 9h). In this case, the exponentiation corresponds to the repeated matrix multiplication of $\mathbf{\Gamma}$, which increases the severity of the motion constraint as the eigenvalues become more extreme. In the two-dimensional case, (19) becomes $\mathbf{\Gamma} = \mathbf{Q} \begin{bmatrix} s_1^2 & 0 \\ 0 & \frac{1}{s_1^2} \end{bmatrix} \mathbf{Q}^{-1}$, hence taking powers of $\mathbf{\Gamma}$ with $n > 1$ increases the largest eigenvalue while decreasing the smallest one, assigning a higher weight to the main motion direction.

The corresponding ergodic cost functions over time are shown in Fig. 9d. The results indicate that increasing n leads to stricter adherence to the motion constraints, while the convergence of the ergodic cost slows down. This is expected since a stronger adherence to the motion constraints limits the solution space, hindering optimality in terms of covering the reference distribution. Note that, nonetheless, the ergodic cost decreases consistently over time in all cases. Finally, also note that the possibility to accentuate the effect of the main motion directions provides users with a principled way to regulate the system's adherence to the demonstrated patterns.

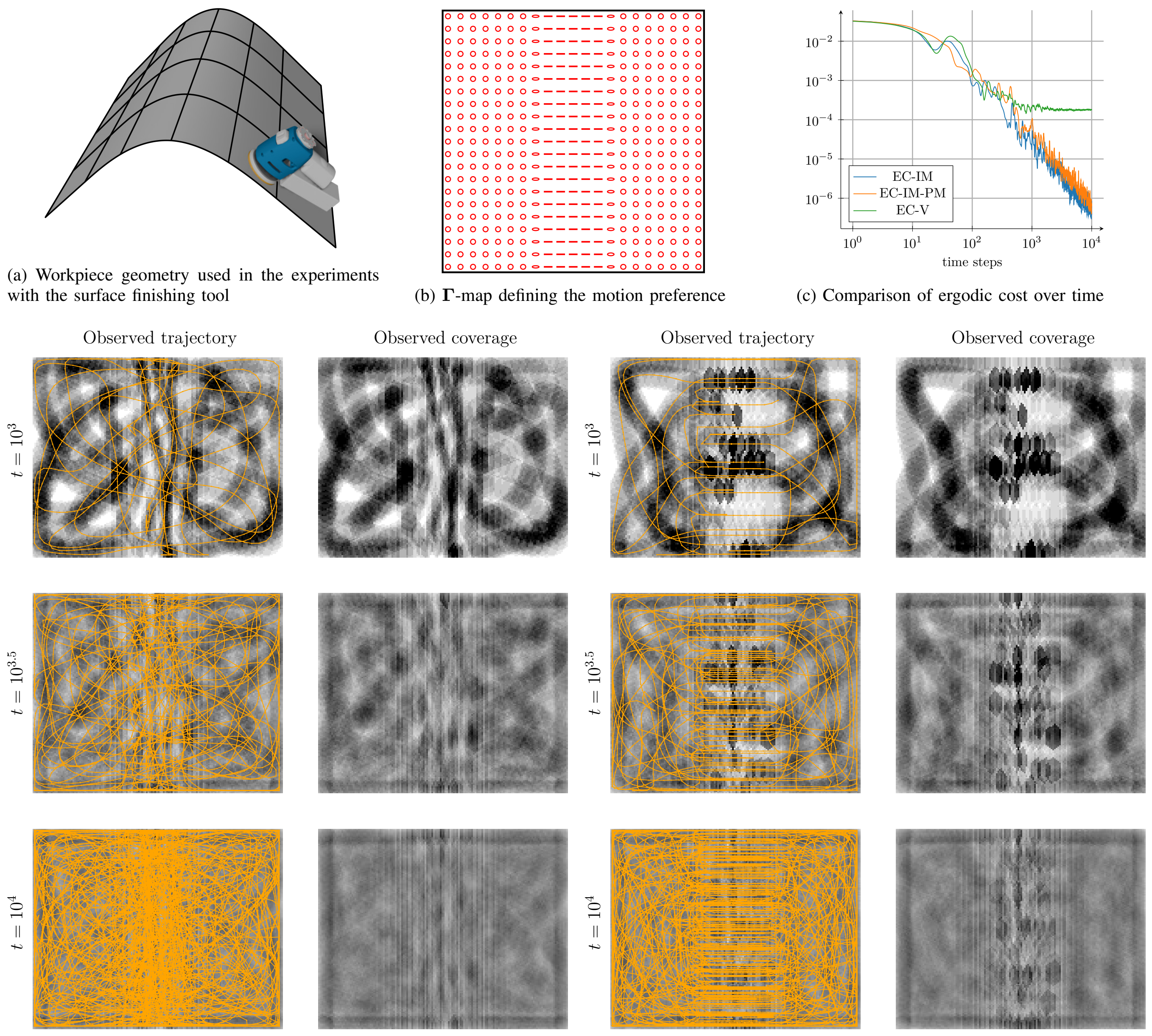
C. Simulation with Imprint Model

We further extend the previous experiment by incorporating an imprint model to simulate the interaction between the finishing tool and the surface. The simulation assumes a constant normal pressure applied across varying imprints (compare (15)).

For this experiment, we use the geometry shown in Fig. 10a, which consists of flat regions on the sides and a varying curvature in the middle section. The geometry affects the imprint of the tool in a way that, in flat regions, the entire finishing disk is in contact with the surface, creating a circular imprint. In contrast, increasing curvature reduces the contact area of the finishing disk. The reference distribution is manually set to be uniform in the rectangular domain U. In Fig. 10d we show the result of the ergodic control approach considering the imprint model (EC-IM) for different numbers of time steps: 10^3 (first row), $10^{3.5}$ (second row), and 10^4 (third row). It can be seen that the density of the trajectory points (orange line) increases in the regions where the tool has a smaller contact area with the geometry, compensating for the smaller imprint by visiting these areas more frequently than those with a larger imprint.

A direct consequence of the need to spend more time in the middle section due to the smaller contact area is that, if preferred motion directions are not enforced – as in the case of EC-IM – a significant number of motions are generated orthogonally to the curvature. This behavior is arguably suboptimal for surface finishing, as discussed in Section I. With this in mind, we subsequently evaluate the approach extended with preferred motion directions (EC-IM-PM).

As discussed in IV, our formulation allows both for learning the motion constraints from demonstrations or directly



(a) Workpiece geometry used in the experiments with the surface finishing tool

(b) $\mathbf{\Gamma}$-map defining the motion preference

(c) Comparison of ergodic cost over time

(d) EC-IM

(e) EC-IM-PM

Fig. 10: Simulation experiment to compare EC-IM and EC-IM-PM for a single curved workpiece surface (a). The experiment demonstrates motion preferences that follow the surface curvature, as shown in (b). It compares their performance using the SMC ergodic cost function (c) with EC-V as a baseline. Subfigures (d) and (e) show the coverage maps augmented with the tool reference trajectory to compare EC-IM results on the left (d) and EC-IM-PM on the right (e) for three different numbers of time steps.

handcrafting them, through the manual definition of $\mathbf{\Gamma}$. In this experiment, we again use a uniform desired distribution, and add a handcrafted constraint such that, in the center of the workpiece, the motion follows the direction of the surface curvature, as a craftspeople would do in such a region, counteracting the tendency observed with EC-IM. The corresponding $\mathbf{\Gamma}$-map is shown in Fig. 10b. In the center parts, the preferred direction of $\mathbf{\Gamma}$ aligns with the horizontal coordinate axes, with one eigenvalue significantly larger than the other; hence, the constraint is strongly enforced in these regions. Towards the left and right outer regions, the constraint gradually fades into an unconstrained area, exhibiting the same behavior observed from the mixture term (34) when learning the $\mathbf{\Gamma}$-map from demonstration.

In Fig. 10e, we show the results in the same manner as for EC-IM at three different time steps. EC-IM-PM achieves the desired coverage at $t = 10^4$, while respecting the desired motion patterns in the middle section, as encoded in $\mathbf{\Gamma}$. The coverage in the center part, where motion constraints are enforced, converges slightly more slowly than EC-IM to the desired distribution, as motion constraints influence the fulfillment of the ergodic objective. Furthermore, as seen in Fig. 10c, EC-IM-PM achieves a slightly higher ergodic cost than EC-IM while, nonetheless, still consistently decreasing over time.

To assess the importance of using the imprint model in

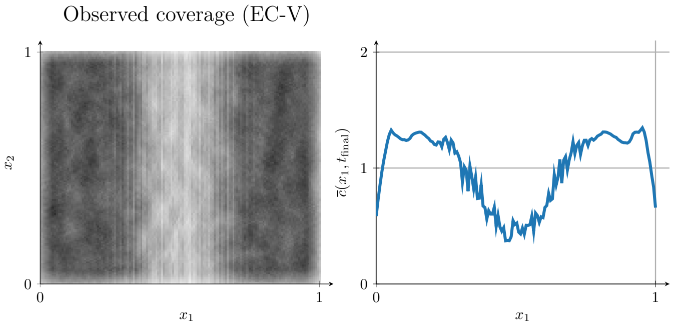
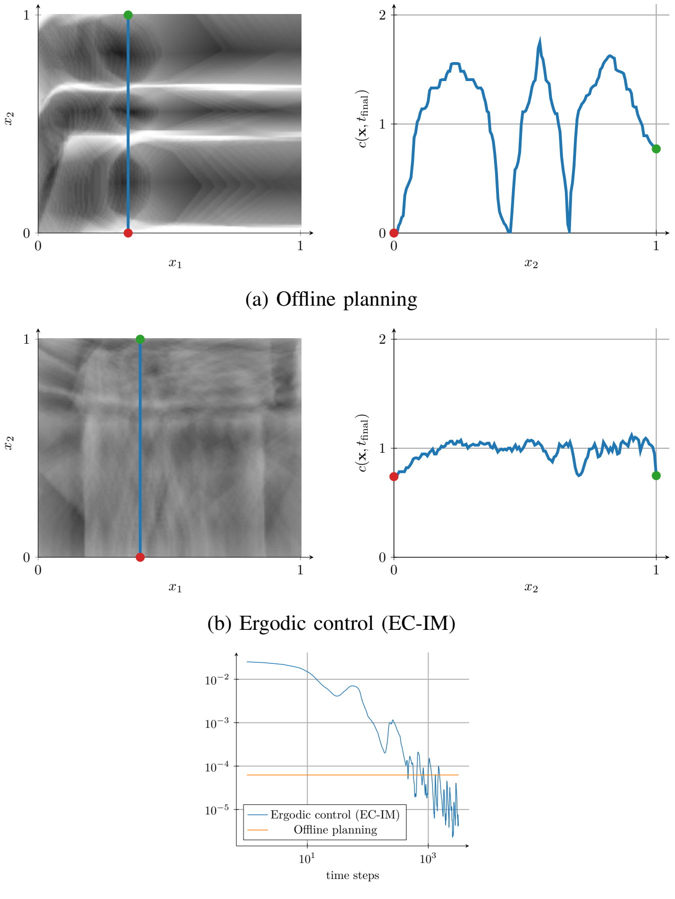


Fig. 11: Result for an EC-V experiment to guide the system and evaluated with the coverage map considering the imprint model as in EC-IM. The result does not converge to the target distribution but to a constant ergodic cost value, as regions with smaller imprints are not compensated by more dense trajectories that spend more time in those regions. The subplot on the left shows the coverage map at the end of the experiment t_{final}, while the subplot on the right displays the mean coverage $\bar{c}(x_1, t) := \int_0^1 c(\begin{bmatrix} x_1 & x_2 \end{bmatrix}^\mathsf{T}, t)\, \mathrm{d}x_2$ along the x_2 axis for each x_1 value. This highlights the regions that are under- and overprocessed relative to the uniform target distribution.



(a) Offline planning

(b) Ergodic control (EC-IM)

(c) Ergodic cost

Fig. 12: Comparison between offline surface finishing approach with adaptive spacing to achieve a full coverage (a) with ergodic control EC-IM with imprint model (b). For both subplots of the methods, we show the coverage map of the resulting trajectories at the end of the experiment t_{final} on the left and its cross-section marked as a blue line on the right to break down the profile in more detail. The right-hand plot thus shows the coverage intensity $c(\mathbf{x}, t)$ along the blue lines in the x_2-direction in the two-dimensional state space $\mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^\mathsf{T}$. The position $x_2 = 0$, marked in red, corresponds to the lower end and $x_2 = 1$, marked in green, to the upper end of the coverage map. The subplot (c) shows both approaches' ergodic costs over time. Since the offline approach has a constant runtime and is not performed online, it is shown here as a constant value. In contrast, the ergodic control approach improves with increasing runtime and outperforms the offline coverage after some time.

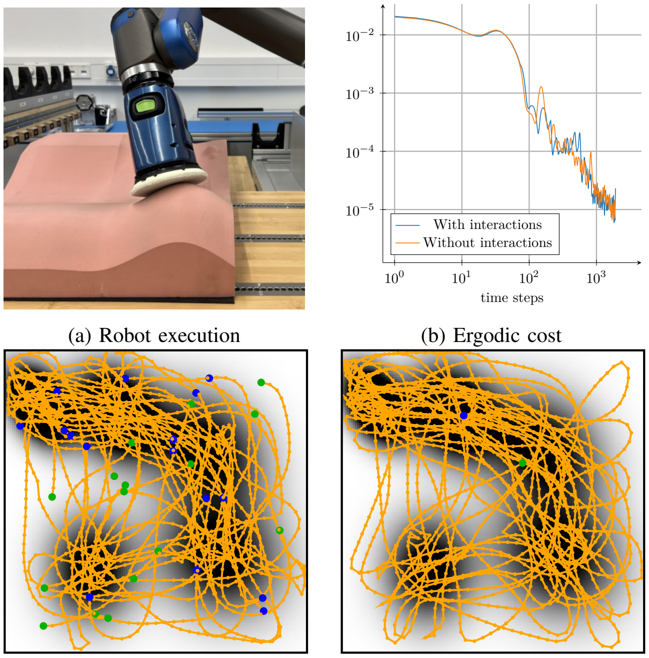
such a task where the contact area varies, we simulate a *vanilla* ergodic controller where the agent is treated as a point mass (EC-V) and evaluate the ergodic cost of the resulting trajectory using the imprint model. In this way, we can better assess the importance of accounting for the contact area when computing the coverage history. Figure 10c shows the obtained result (green line). We observe that the ergodic cost stops decreasing around $t = 10^3$, highlighting the mismatch between the actual agent footprint and the simplified, point mass one. Furthermore, we show the corresponding coverage in Fig. 11, side by side with the desired one. The results indicate that the approach without the imprint model does not converge to the desired distribution. Specifically, areas with a smaller imprint are visited less frequently than those with a larger imprint. This provides a visual interpretation of EC-V not minimizing the cost function effectively and asymptotically converging to a constant cost error as shown in Fig. 10c. For completeness, in Section E we provide a comparison between ergodic cost and a probabilistic coverage error (PCE) metric, highlighting that the former is a valid metric for evaluating coverage quality. The rate of convergence for EC-IM-PM maintains an $\mathcal{O}(t^{-1})$ complexity, consistent with the EC-IM approach, however, with a slightly lower rate.

D. Comparison to Offline Planning

In our next experiment, we compare EC-IM against offline planning approaches [5], [11] for achieving uniform coverage on a convex, non-trivial geometry. The geometry in question is shown Fig. 8a from the previous experiment. It corresponds to segment 1 used in the offline planning approach [5], where the width of the tool imprint is utilized to adjust the spacing between trajectory lines in a grid pattern to avoid gaps and overlaps.

The result of the coverage map produced by the offline planning approach is depicted in Fig. 12a. We observe that the parallel coverage lines yield varying intensity of coverage, with points near the center of the tool imprint receiving more intense coverage than those at the edge. This disparity arises from the longer imprint length in the direction of the trajectory for points near the center compared to those at the edge. As the length of the imprint in the direction of the trajectory increases, the overlapping in the motion direction becomes more pronounced. In contrast, EC-IM directly minimizes the deviation of the coverage distribution from the desired uniform distribution, thereby producing a more uniform coverage as shown in Fig. 12b.

The ergodic cost over time is presented in Fig. 12c compared to the cost obtained from the offline planning approach. Notably, the ergodic control with the imprint model starts with a higher cost but ultimately outperforms the offline planning approach after some time. This is because the ergodic control consistently minimizes the cost function. In contrast, the grid pattern constrains the offline planning approach and avoids overlapping trajectory lines to compensate for the varying



(a) Robot execution

(b) Ergodic cost

(c) With interactions

(d) Without interactions

Fig. 13: In comparing two scenarios of a robot surface finishing tasks (a), we analyze the ergodic cost over time (b). The first scenario (c) involves 15 interruptions to the ergodic control path using a state machine, marked by green dots as starting points of the ergodic trajectory and blue dots as end points. The second scenario (d) features uninterrupted execution of a single ergodic control path.

intensity of coverage.

However, as demonstrated in previous studies [47], the overlap of parallel coverage lines, which arises from offline planning approaches, also has its limitations. Specifically, the overlapping of two continuous coverage distributions of the offline planning method can only approximately sum up to a uniform distribution, rather than achieving it precisely. Furthermore, these approaches are typically designed for uniform distributions and lack the flexibility to accommodate other desired distributions. Another key advantage of ergodic control with the imprint model is the ability to create more natural surface textures.

Unlike traditional offline planning methods that result in parallel lines on the surface, this approach creates unstructured paths that maintain the texture of a naturally occurring workpiece. Workers often prefer to process surfaces manually rather than relying on machines because it allows for more natural results. Automation with ergodic control can achieve similar benefits.

E. Robot Experiment: Interactive Surface Finishing

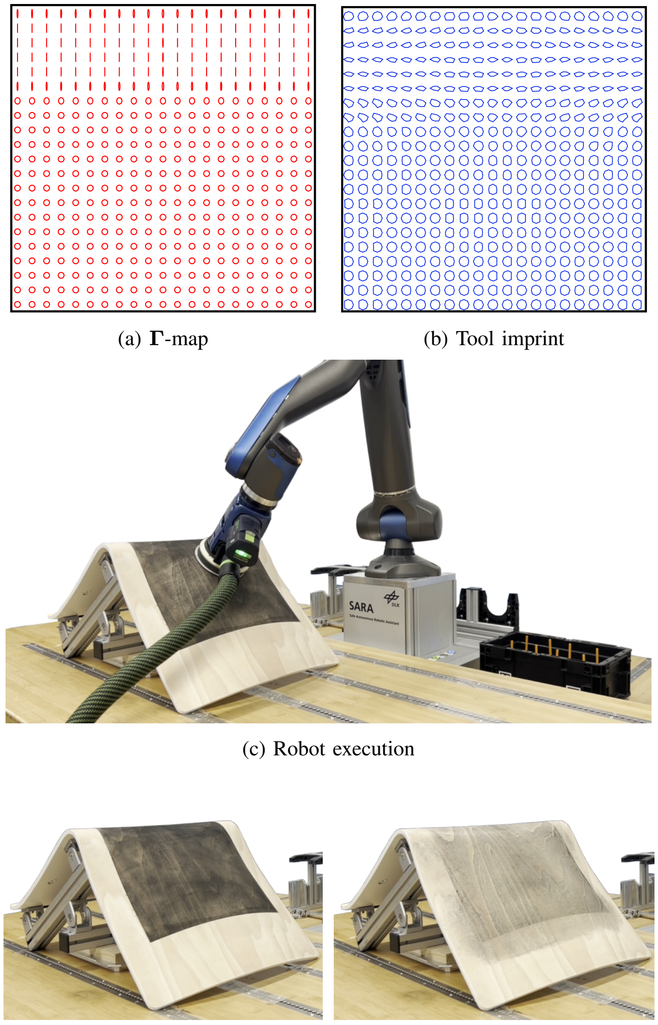
In this subsection, we present an experiment demonstrating the robot's interactive behavior during execution. The experiment leverages the online formalism of ergodic control, enabling state changes without compromising overall coverage. We utilize the DLR SARA robot, a 7-DoF torque-controlled manipulator, to validate the proposed approach in a real-world scenario. The joint torque sensor, in combination with additional force-torque sensors, allows for measuring the interaction forces during surface finishing tasks, as detailed in [48]. We employ a Festool random orbital sander with a 125 mm disk diameter for the finishing tool. The robot employs impedance control [49] with a force overlay $\mathbf{f}_d$ to generate the desired normal force during task execution, as described by $\boldsymbol{\tau} = \mathbf{J}^{\mathsf{T}}(\mathbf{q})[\mathbf{G}_P(\mathbf{p}_d - \mathbf{p}) + \mathbf{G}_D(\dot{\mathbf{p}}_d - \dot{\mathbf{p}}) + \mathbf{f}_d]$ where $\boldsymbol{\tau}$ is the torque vector commanded to the robot, $\mathbf{G}_P$, $\mathbf{G}_D$ are the stiffness and the damping gain matrices respectively, $\mathbf{J}(\mathbf{q})$ denotes the Jacobian of the end-effector for configuration $\mathbf{q}$, and $\mathbf{p}_d$, $\mathbf{p}$, $\dot{\mathbf{p}}_d$, $\dot{\mathbf{p}}$ are the end-effector desired and current positions and velocities. The desired signals are generated by the ergodic controller. For more details on robot control, please refer to [48], [50], [51]. To enable seamless interaction and task execution, we make the end-effector stiffness $\mathbf{G}_P$ decrease with the distance to the surface, varying from a high stiffness near the surface to a low stiffness far from the surface, facilitating the manual repositioning of the robot. When the robot's tool is lifted by human intervention and is no longer in contact with the surface, the ergodic control execution is paused. It is resumes once the tool reestablishes contact with the surface, at which point the new position is updated, and the execution of ergodic control continues. We again use the workpiece surface as illustrated in Fig. 8a to test the online planning approach. Moreover, we use the same desired distribution learned from the human operator without considering motion constraints.

The comparison of two execution scenarios is shown in Fig. 13. In scenario one (see Fig. 13c), the robot executes the task while enabling user interaction for position adjustments. User interruptions occur 15 times, where the user moves the robot away from the surface, repositioning it at another location on the workpiece. Trajectory starts are marked with green dots, and endpoints with blue dots. Scenario two (see Fig. 13d) involves uninterrupted execution of an ergodic control path, taking about 10 minutes, compared to the first scenario, which is prolonged by user interactions, such that the ergodic control in both cases is executed with the same number of time steps.

The robot during the experiment is shown in Fig. 13a, with a fleece material attached to the surface finishing tool for polishing. Both scenarios demonstrate effective task following and surface coverage (see Fig. 13). As shown in Fig. 13b, the ergodic cost decreases similarly in both scenarios, indicating that the robot can adapt to user interactions without compromising convergence.

F. Robot Experiment: Application Paint Removal

Moreover, we evaluate the performance of our proposed methods on a paint removal application to achieve uniform coverage on a non-trivial geometry with varying curvature, focusing on motion preference aspects (see Fig. 14). The experimental setup involves a wooden chair, which serves as the workpiece and consists of an almost flat region with small curvature and a bending region with a large main curvature. The central point of the disk is used as the tool center point (TCP). The tool imprint varies depending on the position on the workpiece, being larger in flat regions and smaller in curved areas, as illustrated in Fig. 14b. For the



(a) $\mathbf{\Gamma}$-map

(b) Tool imprint

(c) Robot execution

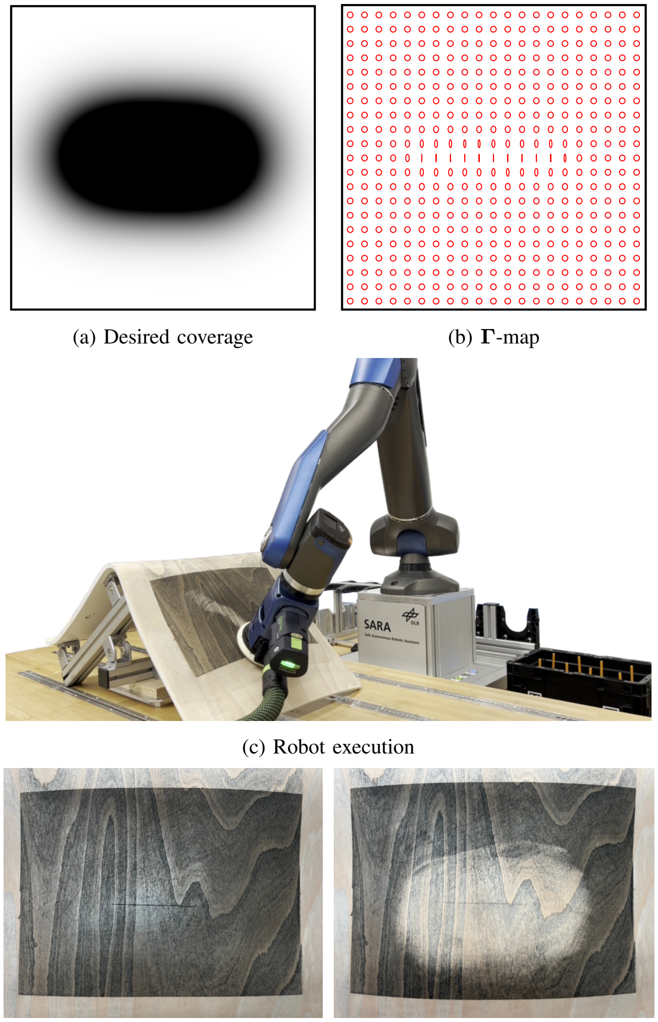
(d) Surface before experiment

(e) Final outcome

Fig. 14: The 7-DoF robot executes a paint removal task with uniform coverage, prioritizing motion along the main curvature (a) while incorporating tool imprint modeling (b). The experiment setup is shown in (c), utilizing the finishing disk center point as TCP. The surface was pre-colored before the experiment (d), and after processing, the color was evenly removed across the entire surface (e).

practical experiment, we apply a constant normal force across varying imprints (compare (15)). This provides a challenging yet practical testbed for surface finishing applications.

The Festool random orbital sander is equipped with sandpaper with grit 180. An interface soft pad is attached between the sander and the sandpaper to enhance adaptability to curved surfaces (see Fig. 14c). Additionally, a Festool suction system is used to minimize dust and particle generation during the process. To evaluate the coverage uniformity, we use Festool marking powder, which is distributed uniformly in a rectangular shape on the surface of the chair workpiece before the experiment begins. The goal for the robotic system is to remove this marking powder, ensuring uniform coverage across the selected region. In addition to ergodic coverage, the EC-IM-PM incorporates motion constraints to guide the robotic system through regions of high curvature (see Fig. 14a). The result demonstrates that the approach achieves uniform coverage across all regions of the wooden chair (see Fig. 14d and Fig. 14e). Furthermore, EC-IM-PM moves in a natural, human-like way in the curved bending region by adhering to the prescribed motion constraints and following the curvature of the surface. The robot experiment lasted 32 minutes, a duration comparable to the time a human operator needs for this paint removal task. The reader is referred to the accompanying video for details.



(a) Desired coverage

(b) $\mathbf{\Gamma}$-map

(c) Robot execution

(d) Surface before experiment

(e) Final outcome

Fig. 15: The experiment involves scratch removal using a convoluted Gaussian kernel as the desired distribution (a). Motion constraints perpendicular to the scratch direction are employed for effective removal (b). The robotic workcell setup and the robot execution using the finishing disk's edge point as TCP, are illustrated in (c). The surface is uniformly covered with marking powder (d) to facilitate precise profiling. The robot successfully removes the desired convoluted Gaussian profile from the surface paint (e).

G. Robot Experiment: Application Scratch Removal

Finally, we investigate the ability of our system to remove localized defects, e.g., scratches, from a surface. Once more, the wooden chair is used as the workpiece, with a single scratch introduced on its surface before processing (see Fig. 15). The goal is for the robotic system to selectively process the region around the scratch while leaving the rest of the surface untouched. We define the desired coverage as a Gaussian function convoluted along the scratch points to achieve this. It ensures that the system focuses only on the immediate vicinity of the scratch (see Fig. 15a). Additionally, we impose a motion constraint that requires the robotic system to move perpendicular to the scratch line during processing (see Fig. 15b).

The SARA robot is equipped with the Festool random orbital sander with grit 180 and uses the vacuum system again to reduce the dust produced (see Fig. 15c). We do not use the interface soft pad to enhance precision and employ an edge-point contact configuration instead of the standard center-point contact. This results in a smaller imprint of the finishing disk and increased pressure per unit area, which is beneficial for localized defect removal while maintaining constant force. The imprint model is not used in this scenario because the edge-point contact produces a nearly constant, punctual imprint in the processing region. We again use Festool marking powder to assess the coverage uniformity within the processing region for evaluation.

The results demonstrate that the system successfully removes the desired shape corresponding to the scratch and adheres to the prescribed motion constraints (see Fig. 15d and Fig. 15e). This experiment highlights the effectiveness of EC-PM in achieving precise and localized defect removal while achieving the Gaussian-shaped coverage function. The duration of this robot experiment was 17 minutes. The reader is referred to the accompanying video for details.

VI. Discussion

The results in Section V show that our approach successfully permits users to formulate surface finishing problems via ergodic control, while accounting for many of the intricacies of this type of task. Experiments V-A, V-B show that our solution allows for the learning and fulfillment of preferred motion directions during surface finishing. The results consistently indicate that, despite the motion constraints, the coverage cost decreases over time. They also suggest that enforcing preferred motion directions affects the rate of ergodic cost convergence. Particularly, this effect is more noticeable as constraints are enforced more strongly. We argue that this trade-off between coverage and naturalness might also be implicitly present in craftsmen-driven surface finishing, as it is known that the motion directions play an important role [13]. Our optimization-based framework thus provides a principled way to quantify this trade-off mathematically, making the task better suited for execution by autonomous systems, as we show in experiments V-F, V-G. The results in experiments V-C, V-D validate our approach to include the imprint model of the finishing disk as part of the agent footprint in ergodic control. Particularly, they show that the robot spends more time in regions where the contact area is smaller and can better avoid both under- and over-processed regions, which are common in offline planning solutions. Finally, experiment V-E shows that our formulation allows the operator to physically interact with the robot without compromising the overall coverage quality, making it well-suited to collaborative surface finishing scenarios.

It should be noted that, despite our focus on surface finishing, several of our contributions are application-agnostic and are likely to have an impact on other ergodic control frameworks. Learning and enforcing preferred motion directions during coverage may benefit the broader scope of robotic exploration. For instance, aerial or underwater vehicles can favor paths aligned with favorable currents or wind to reduce energy expenditure and extend mission duration, agricultural robots can bias traversal perpendicular to gentle slopes to improve energy efficiency and coverage consistency, and swarm robots can develop collective motion preferences that reduce overlap while maximizing area coverage. Even in planetary or remote exploration, such statistical preferences can guide agents toward energetically favorable or information-rich motions, improving endurance and data collection without restricting flexibility.

Although due to our surface parametrization framework, we focus on the two-dimensional case, our derivations are generic to spaces of arbitrary dimension.

One limitation of the presented approach is its reliance on prior geometric knowledge about the workpiece, which limits its direct applicability in scenarios involving unknown geometries that require upstream scanning of three-dimensional meshes. Another limitation lies in the footprint model used by our agent. While it captures essential information about the contact area, its accuracy is limited when accounting for real-world effects during tasks. Deviations from the workpiece geometry or position can impact force distribution and material removal accuracy at different points of the disk, making it difficult to model. This underscores the need for integrating sensors to enhance model fidelity and reduce modeling errors. The current motion constraint is limited to a single dominant direction, which could be overcome by enabling the representation of multiple directional motion preferences in future work. Further research is needed to explore continuous changes in contact points. Lastly, while our online ergodic control approach for generating ergodic trajectories is robust, its time efficiency could be improved regarding convergence speed. Enhancing this aspect would make the approach more practical for real-world applications with stringent cycle time requirements.

VII. Conclusion

We presented a novel ergodic control framework for learning robot surface finishing skills with human preferences. Our approach builds on the SMC method and incorporates tool imprints, target distributions, and motion preferences learned from human demonstrations. The proposed framework enables robots to learn complex surface finishing tasks with high-level task descriptions or human demonstrations, which facilitates easier programming of robotic systems by craftspeople. The experimental results show the effectiveness of our approach for achieving uniform coverage on convex geometries, outperforming *vanilla* ergodic control and offline planning methods. Additionally, they show that it is possible to learn both desired coverage behaviors and preferred motion directions from human demonstrations. The results were further validated successfully using the DLR SARA robot in a real-world scenario.

ACKNOWLEDGMENTS

This work was supported in part by the German Federal Ministry of Education and Research (BMBF) through the ”The Future of Value Creation — Research on Production, Services and Work” Program under Grant 02K20D032 and in part by the DLR internal Project ”ASPIRO”.

APPENDIX

A. Minimization of Ergodic Cost under Motion Constraints

Consider the constrained optimization problem of minimizing the objective function

$$\begin{aligned} \min_{\mathbf{u}} \quad & \mathbf{b}(t)^\intercal \mathbf{u}, \\ \text{subject to} \quad & \mathbf{u}^\intercal \mathbf{\Gamma}^{-1} \mathbf{u} = d^2. \end{aligned} \tag{37}$$

We can solve this constrained optimization problem using the method of Lagrange multipliers and define the objective function and the constraint as

$$\mathbf{f}(\mathbf{u}) = \mathbf{b}(t)^\intercal \mathbf{u}, \tag{38}$$

$$\mathbf{g}(\mathbf{u}) = \mathbf{u}^\intercal \mathbf{\Gamma}^{-1} \mathbf{u} - d^2. \tag{39}$$

The Lagrangian function is then given by

$$\mathcal{L}(\mathbf{u}, \lambda) = \mathbf{f}(\mathbf{u}) - \lambda \mathbf{g}(\mathbf{u}). \tag{40}$$

To find the stationary points of the Lagrangian, we set the partial derivatives of the Lagrangian with respect to the control input $\mathbf{u}$ and the Lagrange multiplier λ to zero. This yields the following system of equations

$$\frac{\partial \mathcal{L}}{\partial \mathbf{u}} = \frac{\partial \mathbf{f}(\mathbf{u})}{\partial \mathbf{u}} - \lambda \frac{\partial \mathbf{g}(\mathbf{u})}{\partial \mathbf{u}} = \mathbf{b}(t) - 2\lambda \mathbf{\Gamma}^{-1} \mathbf{u} = 0, \tag{41}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = \mathbf{g}(\mathbf{u}) = \mathbf{u}^\intercal \mathbf{\Gamma}^{-1} \mathbf{u} - d^2 = 0, \tag{42}$$

considering the symmetric property of $\mathbf{\Gamma}$. Solving (41) for $\mathbf{u}$, we obtain

$$\mathbf{u} = \frac{1}{2\lambda} \mathbf{\Gamma} \mathbf{b}(t). \tag{43}$$

Substituting this expression for $\mathbf{u}$ into (42) yields

$$\lambda = \pm \frac{\sqrt{\mathbf{b}(t)^\intercal \mathbf{\Gamma} \mathbf{b}(t)}}{2d}. \tag{44}$$

Finally, substituting λ back into the expression for $\mathbf{u}$, we obtain

$$\mathbf{u} = \pm d \frac{\mathbf{\Gamma} \mathbf{b}(t)}{\sqrt{\mathbf{b}(t)^\intercal \mathbf{\Gamma} \mathbf{b}(t)}}. \tag{45}$$

The solution with the negative sign minimizes the objective function. Therefore, the optimal control input $\mathbf{u}^*$ is given by

$$\mathbf{u}^* = -d \frac{\mathbf{\Gamma} \mathbf{b}(t)}{\sqrt{\mathbf{b}(t)^\intercal \mathbf{\Gamma} \mathbf{b}(t)}}. \tag{46}$$

B. Motion Direction of (28) is Invariant to Scaling

Given the constraint

$$\sqrt{\mathbf{u}^\intercal \mathbf{A} \mathbf{u}} = d \tag{47}$$

where $\mathbf{A}$ is an invertible, positive definite matrix in D dimensions and $\mathbf{A} := \mathbf{\Gamma}^{-1}$, we aim to justify restricting the determinant of $\mathbf{\Gamma}$.

If $\det(\mathbf{A}) \neq 1$, we can normalize $\mathbf{A}$ by scaling it with a factor c such that

$$\det(c\mathbf{A}) = c^D \cdot \det(\mathbf{A}) = 1, \tag{48}$$

$$c = \frac{1}{\sqrt[D]{\det(\mathbf{A})}}. \tag{49}$$

This scaling leads to a matrix $\mathbf{A}' := c\mathbf{A}$ with $\det(\mathbf{A}') = 1$. Correspondingly, the parameter d can be adjusted to

$$\sqrt{\mathbf{u}^\intercal \frac{\mathbf{A}'}{c} \mathbf{u}} = d, \tag{50}$$

$$\sqrt{\mathbf{u}^\intercal \mathbf{A}' \mathbf{u}} = \sqrt{c} \cdot d =: d'. \tag{51}$$

such that the constraint remains satisfied.

This demonstrates that choosing a matrix $\mathbf{A}$ with a determinant different from one merely scales the parameter d and does not alter the constraint's essential nature. Therefore, restricting $\det(\mathbf{\Gamma}) = 1$ is without loss of generality, as any scaling can be compensated by adjusting d. Thus, it is redundant to consider different determinants for $\mathbf{A}$ (and consequently for its inverse $\mathbf{\Gamma}$) since they can all be normalized to have a determinant of one.

C. Mean Velocity in Surface Finishing Demonstrations is Close to Zero

To justify the assumption that the mean velocity $\boldsymbol{\mu}_u$ is zero, we start by considering the average overall velocities, which can be calculated as

$$\boldsymbol{\mu}_u = \frac{1}{M} \sum_{i=0}^{M} \mathbf{u}^{(i)}, \tag{52}$$

where $\mathbf{u}^{(i)}$ represents the velocity at the i-th data point, and M is the total number of data points. Integrating the velocities over time gives the overall displacement of the system

$$\mathbf{u}(t) = \dot{\mathbf{x}}(t), \tag{53}$$

$$\int_0^T \mathbf{u}(t)\, \mathrm{d}t = \int_0^T \dot{\mathbf{x}}(t)\, \mathrm{d}t = \mathbf{x}(T) - \mathbf{x}(0), \tag{54}$$

where $\mathbf{x}(t)$ is the position at time t, and T is the total time duration. Assuming the data points are sampled uniformly over time, we can discretize the integral as

$$\sum_{i=0}^{N} \mathbf{u}^{(i)} \Delta t = \mathbf{x}(T) - \mathbf{x}(0), \tag{55}$$

where Δt is the time interval between consecutive data points, and $T = N \cdot \Delta t$. An ergodic controller aims to cover the entire state space rather than reaching a specific point. Therefore, it is reasonable to assume that the initial and final positions are similar, $\mathbf{x}(0) \approx \mathbf{x}(T)$, or at least that the displacement is relatively small compared to the overall trajectory. Under this assumption, we have

$$\mathbf{x}(T) - \mathbf{x}(0) \approx 0. \tag{56}$$

Substituting this into the discretized integral, we get

$$\sum_{i=0}^{N} \mathbf{u}^{(i)} \Delta t \approx 0. \tag{57}$$

Dividing both sides by $N\Delta t$, we obtain

$$\boldsymbol{\mu}_u = \frac{1}{M} \sum_{i=0}^{M} \mathbf{u}^{(i)} \approx 0. \tag{58}$$

Thus, we can conclude that the mean velocity $\boldsymbol{\mu}_u$ is approximately zero, which justifies the assumption used in our model.

D. Weighted Sum of Matrices is Preserving the Trace

Given a weighted sum of matrices as for $\boldsymbol{\Sigma}_u(\mathbf{x})$ and $\boldsymbol{\Gamma}(\mathbf{x})$, defined in (33) and (34), we demonstrate that the trace of the weighting scheme is preserved. This preservation holds under the assumption that

$$\forall (\mathbf{x}^{(i)}, \mathbf{u}^{(i)}) \in \mathcal{D} : \mathbf{u}^{(i)} \in \mathbb{R}^D,\ \mathrm{tr}(\mathbf{u}^{(i)} \mathbf{u}^{(i)\intercal}) = 1. \tag{59}$$

and use a scaled identity matrix $D^{-1}\mathbf{I}$ with trace one. The trace of the weighted sum can be calculated as

$$\begin{aligned} \mathrm{tr}\left(\boldsymbol{\Sigma}_u(\mathbf{x})\right) =& \sum_{i=1}^{M} k(\mathbf{x}, \mathbf{x}^{(i)}) \underbrace{\mathrm{tr}\left(\mathbf{u}^{(i)} \mathbf{u}^{(i)\intercal}\right)}_{1} = \\ =& \sum_{i=1}^{M} \beta \cdot \exp(-\frac{\|\mathbf{x} - \mathbf{x}^{(i)}\|^2}{2l^2}), \end{aligned} \tag{60}$$

since the trace is linear. To ensure that (33) is trace-preserving, we can determine the scaling factor β as

$$\mathrm{tr}\left(\boldsymbol{\Sigma}_u(\mathbf{x})\right) = 1, \tag{61}$$

$$\beta = \frac{1}{\sum_{i=1}^{M} \exp(-\frac{\|\mathbf{x} - \mathbf{x}^{(i)}\|^2}{2l^2})}. \tag{62}$$

Furthermore, we can derive that the trace of the weighting term as in (33) and (34) is preserved as

$$\begin{aligned} \mathrm{tr}(\boldsymbol{\Gamma}(\mathbf{x})) =& \lambda \mathrm{tr}\left(\boldsymbol{\Sigma}_u(\mathbf{x})\right) + (1-\lambda)\,\mathrm{tr}\left(D^{-1}\mathbf{I}\right) & (63) \\ =& \lambda \cdot 1 + (1-\lambda) \cdot 1 = 1. & (64) \end{aligned}$$

E. Comparison between Ergodic Cost and Probabilistic Coverage Metric

We argue that the SMC ergodic cost can be directly interpreted within surface finishing as a coverage measure when the tool imprint is considered the agent's footprint. This way, it provides a task-relevant performance metric that evaluates whether contact is appropriately distributed over the target region according to a prescribed distribution. To validate our argument, we introduce a probabilistic coverage error (PCE) that measures the discrepancy between the achieved and the desired coverage maps in state space, rather than in the spectral domain, as is done for the SMC cost. Formally, we define it as

$$\mathrm{PCE}(p, c) = \int \left(p(\mathbf{x}) - c(\mathbf{x}, t)\right)^2 \mathrm{d}\mathbf{x}, \tag{65}$$

where $p(\mathbf{x})$ denotes the target distribution and $c(\mathbf{x}, t)$ the coverage distribution.

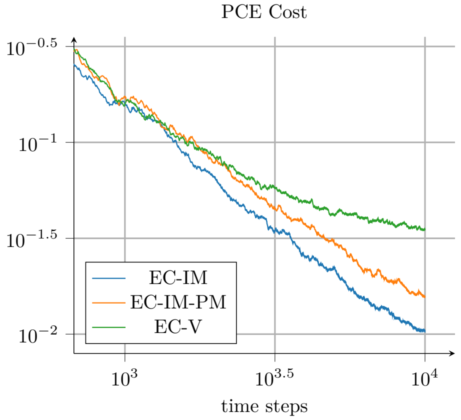


Fig. 16: Comparison of probabilistic coverage error (PCE) for EC-IM, EC-IM-PM, and EC-V (Experiment section V-C)

A comparison of the PCE results for the experiment in section V-C is presented in Fig. 16, highlighting the time steps where the curves of the different approaches start to diverge, for visualization purposes. This comparison demonstrates a strong alignment to the SMC ergodic cost (see Fig. 10c), although the two measures differ in scale. The resulting coverage maps for EC-IM, EC-IM-PM, and EC-V correspond to Fig. 10d, Fig. 10e, and Fig. 11 (left), respectively. Both metrics, PCE and SMC ergodic cost, consistently reflect the quality of coverage and lead to the same qualitative conclusions. Specifically, they exhibit matching trends across algorithms and agree on the relative ranking of different strategies. The primary distinction lies in scaling and more pronounced oscillations observed in the SMC ergodic cost. This behavior can be attributed to the spectral weighting inherent in the ergodic formulation, where deviations in low-frequency components are penalized more strongly than in the PCE measure. These results indicate that the ergodic cost provides a comprehensive and task-relevant characterization of spatial coverage.

References

[1] Y. Li, H. Chen, and N. Xi, "Automatic programming for robotic grinding using real time 3d measurement," in *IEEE Int. Conf. Cyber Technol. Autom., Control, Intell. Syst.*, 2017, pp. 803–808.

[2] Y. Dong, T. Ren, K. Hu, D. Wu, and K. Chen, "Contact force detection and control for robotic polishing based on joint torque sensors," *Int. Journal Adv. Manuf. Technol.*, vol. 107, pp. 2745–2756, 2020.

[3] B. Alt, F. Stöckl, S. Müller, C. Braun, J. Raible, S. Alhasan, O. Rettig, L. Ringle, D. Katic, R. Jäkel, M. Beetz, M. Strand, and M. F. Huber, "Robogrind: Intuitive and interactive surface treatment with industrial robots," in *IEEE Int. Conf. Robot. Autom.*, 2024, pp. 2140–2146.

[4] J. Nguyen, M. Bailey, I. Carlucho, and C. Barbalata, "Robotic manipulators performing smart sanding operation: A vibration approach," in *IEEE Int. Conf. Robot. Autom.*, 2022, pp. 2958–2964.

[5] S. Schneyer, A. Sachtler, T. Eiband, and K. Nottensteiner, "Segmentation and coverage planning of freeform geometries for robotic surface finishing," *IEEE Robot. Autom. Letters*, vol. 8, no. 8, p. 5267–5274, 2023.

[6] P. Xu, C. F. Cheung, C. Wang, and C. Zhao, "Novel hybrid robot and its processes for precision polishing of freeform surfaces," *Precision Engineering*, vol. 64, pp. 53–62, 2020.

[7] A. Garcia, V. Girbes-Juan, J. E. Solanes, L. Gracia, C. Perez-Vidal, and J. Tornero, "Human-robot cooperation for surface repair combining automatic and manual modes," *IEEE Access*, vol. 8, pp. 154 024–154 035, 2020.

[8] J. Shaw and Y.-J. Fang, "Development of grinding and polishing technology for stainless steel with a robot manipulator," *Journal Autom. Mobile Robot. Intell. Syst.*, vol. 15, 2021.

[9] Y. Huo, P. Li, D. Chen, Y.-H. Liu, and X. Li, "Model-free adaptive impedance control for autonomous robotic sanding," in *Trans. Autom. Sci. Eng.* IEEE, 2021.

[10] Y. Zhao, J. Zhao, L. Zhang, L. Qi, and Q. Tang, "Path planning for automatic robotic blade grinding," in *Int. Conf. Mechatron. Autom.* IEEE, 2009, pp. 1556–1560.

[11] Y. Wen, D. J. Jaeger, and P. R. Pagilla, "Uniform coverage tool path generation for robotic surface finishing of curved surfaces," *IEEE Robot. Autom. Letters*, vol. 7, no. 2, pp. 4931–4938, 2022.

[12] H.-Y. Tam, O. Chi-hang Lui, and A. C.K. Mok, "Robotic polishing of free-form surfaces using scanning paths," *Journal Mater. Process. Technol.*, vol. 95, no. 1, pp. 191–200, 1999.

[13] E.-A. Salcă and S. Hiziroglu, "Analysis of surface roughness of black alder as function of various processing parameters." *Pro ligno*, vol. 8, no. 2, 2012.

[14] G. Mathew and I. Mezić, "Metrics for ergodicity and design of ergodic dynamics for multi-agent systems," *Physica D: Nonlinear Phenomena*, vol. 240, no. 4-5, p. 432–442, 2011.

[15] X. Liu, Y. Li, and Q. Li, "A region-based 3+ 2-axis machining toolpath generation method for freeform surface," *Int. Journal Adv. Manuf. Technol.*, vol. 97, pp. 1149–1163, 2018.

[16] P. N. Atkar, D. C. Conner, A. Greenfield, H. Choset, and A. A. Rizzi, "Hierarchical segmentation of piecewise pseudoextruded surfaces for uniform coverage," *IEEE Trans. Autom. Sci. Eng.*, vol. 6, no. 1, pp. 107–120, 2009.

[17] H. Choset, "Coverage for robotics – a survey of recent results," in *Annals Math. Artif. Intell.*, vol. 31, no. 1, 2001, pp. 113–126.

[18] J. Liu, X. Huang, S. Fang, H. Chen, and N. Xi, "Industrial robot path planning for polishing applications," in *IEEE Int. Conf. Robot. Biomimetics*, 2016, pp. 1764–1769.

[19] W. Sheng, H. Chen, N. Xi, J. Tan, and Y. Chen, "Optimal tool path planning for compound surfaces in spray forming processes," in *IEEE Int. Conf. Robot. Autom.*, vol. 1, 2004, pp. 45–50 Vol.1.

[20] D. Bochtis and T. Oksanen, "Combined coverage and path planning for field operations," in *Precision agriculture '09*. Leiden, The Netherlands: Wageningen Academic, 2009, pp. 521 – 527.

[21] M. Xiao, Y. Ding, and G. Yang, "A model-based trajectory planning method for robotic polishing of complex surfaces," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 4, pp. 2890–2903, 2022.

[22] I. Mohsin, K. He, Z. Li, and R. Du, "Path planning under force control in robotic polishing of the complex curved surfaces," *Applied Sciences*, vol. 9, no. 24, 2019.

[23] S. Schneyer, A. Albu-Schaeffer, F. Stulp, and J. Silvério, "Learning-based ergodic control for interactive surface finishing with collaborative robots," in *CoRL 2024 Workshop CoRoboLearn: Advancing Learning for Human-Centered Collaborative Robots*, 2024.

[24] M. Rososhansky, F. Xi, and Y. Li, "Coverage based tool path planning for automated polishing using contact stress theory," in *IEEE Int. Conf. Autom. Sci. Eng.*, 2010, pp. 592–597.

[25] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, "Recent advances in robot learning from demonstration," *Annu. Rev. Control Robot. Auton. Syst.*, vol. 3, no. 1, pp. 297–330, 2020.

[26] A. Seewald, C. J. Lerch, M. Chancán, A. M. Dollar, and I. Abraham, "Energy-aware ergodic search: Continuous exploration for multi-agent systems with battery constraints," in *IEEE Int. Conf. Robot. Autom.*, 2024, pp. 7048–7054.

[27] K. Fitzsimons and T. D. Murphey, "Ergodic shared control: Closing the loop on phri based on information encoded in motion," *ACM Trans. Hum.-Robot Interact.*, vol. 11, no. 4, sep 2022.

[28] L. M. Miller, Y. Silverman, M. A. MacIver, and T. D. Murphey, "Ergodic exploration of distributed information," *IEEE Trans. Robot.*, vol. 32, no. 1, p. 36–52, 2016.

[29] D. Dong, H. Berger, and I. Abraham, "Time optimal ergodic search," *Conf. Robot. Sci. Syst.*, 2023.

[30] G. Sartoretti, A. Rao, and H. Choset, "Spectral-based distributed ergodic coverage for heterogeneous multi-agent search," in *Distributed Autonomous Robotic Systems*, F. Matsuno, S. Azuma, and M. Yamamoto, Eds. Springer International Publishing, 2022, p. 227–241.

[31] L. Lanča, K. Jakac, and S. Ivić, "Model predictive altitude and velocity control in ergodic potential field directed multi-uav search," 2024.

[32] A. Kalinowska, A. Prabhakar, K. Fitzsimons, and T. Murphey, "Ergodic imitation: Learning from what to do and what not to do," in *IEEE Int. Conf. Robot. Autom.* IEEE, 2021, p. 3648–3654.

[33] J. Ketchum, A. Prabhakar, and T. D. Murphey, "Active exploration for real-time haptic training," *IEEE Int. Conf. Robot. Autom.*, 2024.

[34] I. Abraham, A. Prabhakar, M. J. Z. Hartmann, and T. D. Murphey, "Ergodic exploration using binary sensing for nonparametric shape estimation," *IEEE Robot. Autom. Letters*, vol. 2, no. 2, p. 827–834, 2017.

[35] S. Shetty, J. Silvério, and S. Calinon, "Ergodic exploration using tensor train: Applications in insertion tasks," *IEEE Trans. Robot.*, vol. 38, no. 2, p. 906–921, 2022.

[36] M. M. Sun, A. Gaggar, P. Trautman, and T. Murphey, "Fast ergodic search with kernel functions," *IEEE Trans. Robot.*, pp. 1–20, 2025.

[37] T. Löw, J. Maceiras, and S. Calinon, "drozbot: Using ergodic control to draw portraits," *IEEE Robot. Autom. Letters*, vol. 7, no. 4, p. 11728–11734, 2022.

[38] G. Mathew and I. Mezic, "Spectral multiscale coverage: A uniform coverage algorithm for mobile sensor networks," in *IEEE Conference on Decision and Control*, 2009, pp. 7872–7877.

[39] S. Calinon, *Mixture Models for the Analysis, Edition, and Synthesis of Continuous Time Series*. Springer International Publishing, 2020, pp. 39–57.

[40] S. Ivić, B. Crnković, and I. Mezić, "Ergodicity-based cooperative multiagent area coverage via a potential field," *IEEE Trans. Cybern.*, vol. 47, no. 8, p. 1983–1993, 2017.

[41] C. Bilaloglu, T. Löw, and S. Calinon, "Whole-body ergodic exploration with a manipulator using diffusion," *IEEE Robot. Autom. Letters*, vol. 8, no. 12, p. 8581–8587, 2023.

[42] ——, "Tactile ergodic coverage on curved surfaces," *IEEE Trans. Robot.*, pp. 1–15, 2025.

[43] L. M. Miller and T. D. Murphey, "Trajectory optimization for continuous ergodic exploration," in *2013 American Control Conference*, 2013, p. 4196–4201.

[44] L. Liu, L. Zhang, Y. Xu, C. Gotsman, and S. J. Gortler, "A local/global approach to mesh parameterization," in *Computer graphics forum*, vol. 27, no. 5. Wiley Online Library, 2008, pp. 1495–1504.

[45] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[46] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[47] D. Feng, Y. Sun, and H. Du, "Investigations on the automatic precision polishing of curved surfaces using a five-axis machining centre," *Int. Journal Adv. Manuf. Technol.*, vol. 72, no. 9-12, pp. 1625–1637, 2014.

[48] M. Iskandar, O. Eiberger, A. Albu-Schäffer, A. De Luca, and A. Dietrich, "Collision detection, identification, and localization on the DLR SARA robot with sensing redundancy," in *IEEE Int. Conf. Robot. Autom.*, 2021, pp. 3111–3117.

[49] N. Hogan, "Impedance control: An approach to manipulation: Part I-Theory; Part II-Implementation; Part III-Applications," in *Journal of Dynamic Systems, Measurement and Control*, vol. 107, no. 1, 1985, pp. 1–24.

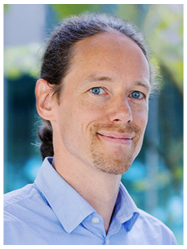
[50] M. Iskandar, C. Ott, O. Eiberger, M. Keppler, A. Albu-Schäffer, and A. Dietrich, "Joint-level control of the dlr lightweight robot sara," in *IEEE Int. Conf. Intell. Robots Syst.* IEEE, 2020, pp. 8903–8910.

[51] M. Iskandar, C. Ott, A. Albu-Schäffer, B. Siciliano, and A. Dietrich, "Hybrid force-impedance control for fast end-effector motions," *IEEE Robot. Autom. Letters*, vol. 8, no. 7, pp. 3931–3938, 2023.

Stefan Schneyer received the M.Sc. degree in robotics, cognition, intelligence from the Technical University of Munich (TUM), Munich, Germany, in 2022. He is currently working toward the Ph.D. degree with the Institute of Robotics and Mechatronics, German Aerospace Center, Weßling, Germany, and the School of Computation, Information and Technology, Technical University of Munich. His research interests include tactile robotics, machine learning, and the development of algorithms for autonomous exploration and coverage path planning.

Korbinian Nottensteiner has been working at the Institute of Robotics and Mechatronics at the German Aerospace Center (DLR) since 2012, where he coordinates activities and projects in the application field of future manufacturing. He previously studied Mechatronics and Information Technology at the Technical University of Munich (TUM). He has headed multiple projects in robot-based manufacturing and has successfully completed his doctorate in the field of autonomous and tactile robotic systems in assembly at TUM in 2023. His research goals are to develop new solutions for applications of robotic systems in production tasks.

Alin Albu-Schäffer received the Ph.D. degree in automatic control from the Technical University of Munich (TUM), Munich, Germany, in 2002. Since 2012, he has been the Head of the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Weßling, Germany, which he joined in 1995. He is also a Professor with TUM and holds the Chair for ”Sensor-Based Robotic Systems and Intelligent Assistance Systems” with the Computer Science Department. He was strongly involved in the development of the DLR lightweight robot and its commercialization through technology transfer to KUKA. His research interests include robot design, modeling and control, nonlinear control, flexible joint and variable compliance robots for manipulation and locomotion, physical human-robot interaction, and bioinspired robot design. He was a recipient of several awards, including the IEEE King-Sun Fu Best Paper Award of IEEE TRANSACTIONS ON ROBOTICS in 2012 and 2014, the EU-Robotics Technology Transfer Award in 2011, several Best Paper Awards at the major IEEE Robotics Conferences as well as the DLR Science Award. In 2019, he was also a recipient of an ERC Advanced Grant for the Project M-Runner, about energy efficient locomotion based on nonlinear mechanical resonance principles.



Freek Stulp received the doctorate degree in computer science from the Technical University of Munich, Munich, Germany, in 2007. He is currently the Head of the Department of Cognitive Robotics, Institute of Robotics and Mechatronics, German Aerospace Center. Previously, he was an Assistant Professor with the École Nationale Supérieure de Techniques Avancées.

João Silvério is the Group Leader of the Interactive Skill Learning Group at the German Aerospace Center (DLR) since April 2022. He received his Ph.D. in Robotics (2017) from the University of Genoa (Genoa, Italy) and the Italian Institute of Technology, where he was also a postdoctoral researcher until May 2019. Between June 2019 and March 2022 he was a postdoctoral researcher at the Idiap Research Institute (Martigny, Switzerland). He is interested in machine learning for robotics, particularly imitation and reinforcement learning. Webpage: http://jpsilverio.github.io.