GeoSACS: Geometric Shared Autonomy via Canal Surfaces

Shalutha Rajapakshe
Idiap Research Institute & EPFL
Switzerland
shalutha.rajapakshe@idiap.ch

Atharva Dastenavar
Idiap Research Institute & EPFL
Switzerland
adastenavar@idiap.ch

Michael Hagenow
University of Wisconsin-Madison
Madison, USA
mhagenow@wisc.edu

Jean-Marc Odobez
Idiap Research Institute
Martigny, Switzerland
odobez@idiap.ch

Emmanuel Senft
Idiap Research Institute
Martigny, Switzerland
esenft@idiap.ch

Abstract

Shared autonomy (SA), which combines user inputs with autonomous capabilities, presents a significant opportunity for assistive robotics. A key challenge in SA is the dimensionality gap: the mismatch between low-dimensional user inputs from familiar interfaces (e.g., 2D joysticks) and the high-dimensional control required by robot manipulators. To enhance usability and acceptance, this mapping must be as simple and intuitive as possible. We introduce GeoSACS, a geometric framework for SA. GeoSACS uses canal surfaces to encode task structure with as few as two demonstrations. While the robot moves autonomously along the canal, users can then make corrections on the 2D planar circular cross-sections orthogonal to the robot motion. By leveraging geometric structure to partition the 6D control space between the robot and the user, GeoSACS allows the intuitive mapping of 2D user inputs to 6D end-effector control. We describe GeoSACS and evaluate its underlying assumptions in a user study against two baselines. Results from the study demonstrate reduced workload and improved performance, providing insights for the design of future SA systems.

CCS Concepts

• **Human-centered computing** → **Collaborative interaction**; • **Computer systems organization** → **Robotic control**.

Keywords

Shared autonomy, human-robot interaction, assistive robotics

ACM Reference Format:
Shalutha Rajapakshe, Atharva Dastenavar, Michael Hagenow, Jean-Marc Odobez, and Emmanuel Senft. 2026. GeoSACS: Geometric Shared Autonomy via Canal Surfaces. In *Proceedings of the 21st ACM/IEEE International Conference on Human-Robot Interaction (HRI '26), March 16–19, 2026, Edinburgh, Scotland, UK*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3757279.3785579

1 Introduction

Assistive robots provide an opportunity to help users who cannot do tasks themselves [7]. However, building fully autonomous robots for real environments remains a challenge due to the variability in human environments and preferences. Frameworks such as shared autonomy (SA) have emerged as a means to blend human inputs with robot control to achieve success [12, 15, 39]. SA can allow humans to delegate the bulk of a task to the robot, intervening only when adjustments are necessary for managing uncertainties [16]. For instance, in a laundry machine loading task, the motion to pick pieces of clothing from a basket and put them in the machine needs to be repeated multiple times. In such a case, the robot could perform the repetitive elements of the task (e.g., moving between the basket and the washer) while the human only provides the required input to specify grasping positions or recovery behaviors.

Despite the potential of SA, controlling all six Cartesian dimensions (position and orientation) of a robot's end-effector remains challenging due to the mismatch between the robot's configuration

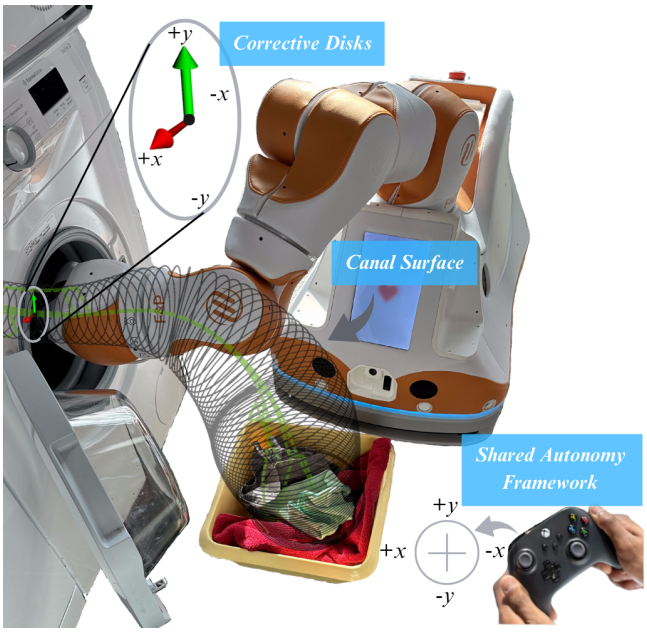


Figure 1: GeoSACS proposes a SA framework to encode robot motions as canal surfaces while allowing users to provide corrections on the cross-sections of the canal to change the way the robot navigates the canal. Using GeoSACS, users can control robots in real task of daily living, such as filling a laundry machine.

This work is licensed under a Creative Commons Attribution 4.0 International License.
HRI '26, Edinburgh, Scotland, UK
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2128-1/2026/03
https://doi.org/10.1145/3757279.3785579

space and the low-dimensional inputs typically provided through familiar interfaces such as joysticks [25]. Latent space methods have been explored to address this challenge [26], but their practical use remains limited: (1) they often require gathering dozens of demonstrations, which is impractical in real-world settings, and (2) the resulting mapping can be hard to comprehend for users [12].

To address these limitations, we introduce in detail GeoSACS, a geometric SA framework that decomposes the 6D control space between the robot and the user, thereby allowing users to control robots with 2D joystick input. With as few as two demonstrations, GeoSACS captures the full range of task variations and enables users to provide real-time corrections to address task-specific uncertainties. Our assumption is that trajectories can be encoded as canal surfaces [20], with the robot autonomously navigating the canal and user providing joystick corrections applied on the canal cross-sections (see Fig. 1), enabling full 6D navigation with a simple joystick. Our contributions are:

- **Detailed description of GeoSACS.** We present the technical details of GeoSACS, an SA framework that maps 2D human correction inputs onto an autonomous behavior.
- **Evaluate GeoSACS design choices.** We conducted a 24-participant study against two baselines to examine (1) control initiation and (2) correction frame orientation within canal-surface cross-sections.
- **Provide design guidelines for SA systems.** Based on the study's evidence of reduced workload and improved usability, we derive insights on how aligning the robot's motion and orientation with task structure can inform the design of more flexible and intuitive alternatives for future SA systems.

2 Related Work

2.1 Learning from Demonstration

Learning from demonstration (LfD) is a well-established technique for teaching robots through human demonstrations, often for autonomous behavior, shared control, or SA [10]. Common LfD approaches include movement primitives (e.g., dynamic movement primitives (DMPs) [21], probabilistic movement primitives [33]), Gaussian mixture regression [8, 9], dynamical systems (e.g., SEDs [23]), and inverse reinforcement learning [31] where a task reward function is inferred from the demonstrations. In addition, behavior cloning [11] has shown promise in handling traditionally challenging tasks, by leveraging hundreds to thousands of demonstrations. However, behavior-cloned policies often remain brittle, with limited generalization to unseen task variations or new environments [3, 44]. To address this limitation, recent work has explored training large behavior models (LBMs) on extensive multitask datasets to learn visuomotor policies [40]. While these approaches improve generalization and robustness, they may still fail to capture user preferences, particularly when such preferences are difficult to articulate or explicitly specify. Another promising LfD approach [1, 2] that meets these requirements uses canal surfaces [20]. It represents a task with a mean curve and a tangent surface that bounds demonstration variability, enabling skills such as pick-and-place and obstacle avoidance. We extend this idea to SA to capture task structure and interpret user inputs.

2.2 Shared Autonomy

In SA, a human and robot work together to complete shared goals [39]. Approaches generally fall into two categories: human-led control [12, 26, 32, 43] and robot-led control [16, 17, 37].

In human-led approaches, the human initiates control while the robot provides assistance or corrections. This paradigm is explored extensively in surgical robotics [28, 43], where surgeons retain initiative and the robot augments execution. In the context of household tasks such as assistive eating and cooking, Losey et al. [26] introduced an encoder–decoder-based method to learn a two degree-of-freedom (DoF) latent space from demonstrations, enabling control through a 2D joystick. However, the encoder–decoder architecture limits the user's ability to interpret the latent space in a meaningful way. Subsequent work [12] sought to address this gap by incorporating additional modalities, such as combining joystick inputs with voice commands, to constrain the latent space.

In contrast, robot-led control allows the robot to initiate task execution, with human providing corrections on top of the autonomous behavior, which is suited for repetitive tasks. Hagenow et al. [16] followed this paradigm in their corrective shared autonomy (CSA) framework, where users provide corrections to robot state variables layered on top of behaviors modeled with DMPs [21]. However, this approach has so far constrained the control space to a 2D manifold, limiting its broader applicability. Similarly, we explore an approach where the robot first learns a nominal task behavior as a canal surface, then adapts it online through a correction framework using low-dimensional 2D joystick input to handle uncertainties.

2.3 Control Mappings

For both teleoperation and SA systems, a key design consideration is the input mapping, or how low-dimensional user inputs map to corrections in the robot's space. Prior work in teleoperation has shown how a poor control mapping can negatively impact user performance [19, 34]. Ideally, this mapping would be intuitive, consistent, and minimize any mental transformations required by the operator [13]. Existing methods in SA have addressed the challenges of input mapping in a limited way (e.g., through heuristics or optimization objectives in learning mappings [26, 35]), however, more general solutions remain an open challenge. Latent spaces, as in [26], offers simplicity, however, such spaces must balance reachability and transparency to be effective. In the context of CSA [6, 16, 36], this is closely tied to the learning method used to establish the robot's nominal behavior and correction dimensions, as the learning process should provide a meaningful foundation for user inputs to effectively modify the nominal behavior and manage task uncertainties. Screw-invariant representations propose frame-independent motion representations that capture coupled translation–rotation structure [24, 42]. While they could be used in SA, they can be complex to expose as a simple, visually grounded low-dimensional correction interface.

2.4 Research Gap

Despite previous work on SA, a fundamental formulation of canal surface-based SA and a systematic examination of its underlying assumptions are still lacking. In particular, questions remain regarding how design choices, such as the allocation of control initiation

between human and robot and the orientation of correction frames, affect system-level performance and user experience. Addressing these gaps is essential for understanding how geometric structures, latent spaces, and control initiation strategies can be leveraged to design future SA systems.

3 Motivating Example

Context. Activities of daily living (ADLs) can be difficult for individuals with disabilities, and robotic assistance can improve independence. However, the complexity and uncertainty of many ADLs limit full autonomy. Keeping the user involved can also be beneficial, especially when task goals are ambiguous, hard to specify, or involve a social aspect [4]. SA can address this by letting the robot perform most of the task while allowing user intervention when needed. For instance, in laundry loading, the robot can autonomously move between the basket and drum, while the user uses a joystick to guide grasping or adjust motion when handling awkward items.

Problem Statement and Proposed Solution. Controlling the 6 Cartesian dimensions of a robot's end-effector using familiar low-DoF input devices, such as gamepads with joysticks, remains a challenge. In this work, we decompose the 6D control space between the user and the robot using canal surfaces, providing a meaningful grounding for mapping 2D user inputs to 6D robot motions.

A caregiver can teach once a robot to load laundry by kinesthetically demonstrating the movement from a laundry basket to the drum twice, showing the boundaries. A canal is then generated, guiding the robot along a nominal trajectory within the canal. As the robot moves back-and-forth within the canal, users can provide real-time joystick corrections that are mapped to a specific correction frame defined on each canal cross-section, refining the trajectory to handle uncertainties such as the varying clothing positions in the basket.

4 Background on Canal Surfaces

Canal Surfaces. The concept of canal surfaces [20] presents a geometric way of representing surfaces and their properties. In the context of robotic control, previous work [1, 2] directly use a discretized version of canal surfaces where the canal is represented as a series of circular cross-sections C_s (referred to as "disks") orthogonal to the tangent vector of a regular curve $\Gamma : \mathbf{x} = c(s) \in \mathbb{R}^3$ (called as the *directrix*) in 3D Cartesian space, with s denoting the discrete state value. The *radii* function $r(s) \in \mathbb{R}$ denotes the radius of the disk, C_s at s^{th} point on the directrix.

Coordinate Frames. Previous work on canal surfaces for LfD [1] has used Bishop frames [5], an alternative to the Frenet–Serret (TNB) frame that minimizes unnecessary rotations and improves trajectory stability, defined by three mutually orthogonal unit vectors: $\mathbf{e}_T$, which aligns with the direction of the *directrix*, and two normal unit vectors, $\mathbf{e}_N$ and $\mathbf{e}_B$. The plane defined by $\mathbf{e}_N$ and $\mathbf{e}_B$, serves as the basis for the disks that define the canal.

Ratio Rule. When navigating the canal, previous work uses the "ratio rule" [2] to accommodate the varying nature of radii along the curve. It maintains a constant ratio of the distance from a point on a disk to its center, applying the same ratio along both the $\mathbf{e}_N$ and $\mathbf{e}_B$ directions across subsequent disks. When needed, an adaptive ratio strategy [1] can be employed using an exponential decay function, for example, to gradually converge to the directrix.

5 GeoSACS

Previous work on canal surfaces has primarily focused on LfD to encode robot motions and support autonomous behaviors. The key insight of this paper is the adaptation of canal surfaces for use in SA. Our motivation for this shift lies in the geometric properties of canal surfaces, which naturally align with task-space representations and could provide high interpretability. We propose that the disks of a canal surface, learned from just two demonstrations, can serve as a grounding for mapping 2D corrective user inputs while the robot autonomously moves along the canal using the ratio rule, influencing robot movements in 3D space. While canal surfaces support circular, elliptical, or B-spline cross-sections [1], we chose circles because (i) they give users uniform correction freedom in all directions, (ii) non-circular shapes need more samples per section to estimate boundaries, conflicting with our goal of minimal demos, and (iii) they isolate the effect of correction-frame alignment without confounding it with changes in the correction boundary.

While the Bishop frames used to generate canal surfaces in [2] could be used as correction frames, they were not designed for SA, leading to frequent directional changes along the curve that lack alignment with a global reference direction (see Fig. 2). We hypothesize that this reduces interpretability and hinders intuitive control. Consequently, instead of using $\mathbf{e}_N$ and $\mathbf{e}_B$ to map user inputs, we define two new correction axes, $\mathbf{x}_s$ and $\mathbf{y}_s$ for each disk (see Section 5.2).

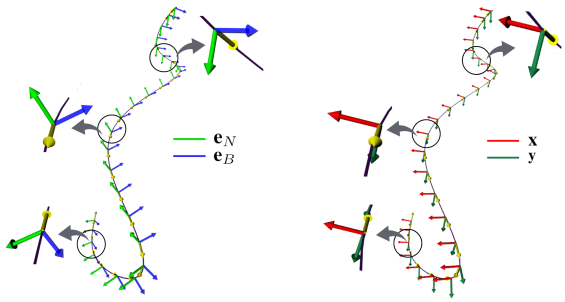


Figure 2: Frame evolution of Bishop frames (left) and our correction frames (right) along a curve, with the tangent vector shown in yellow. Highlighted points show our correction frames are more consistent and exhibit fewer directional changes than Bishop frames.

5.1 Integrating Corrections

With GeoSACS, users can provide corrections at any time using a joystick. When the user provides a correction, the robot pauses at the current disk and responds to joystick input along the two defined correction axes. For example, if a correction occurs when the end-effector is at p_s, a point on the disk C_s, the robot's motion is directed by joystick input but constrained to C_s (see Fig. 3), meaning it does not advance along the canal. Once the joystick is released, the robot resumes motion along the canal from the updated position using the ratio rule. To simplify control, we do

not currently support direct user corrections on orientation, the robot orientation is directly inferred from the demonstrations.

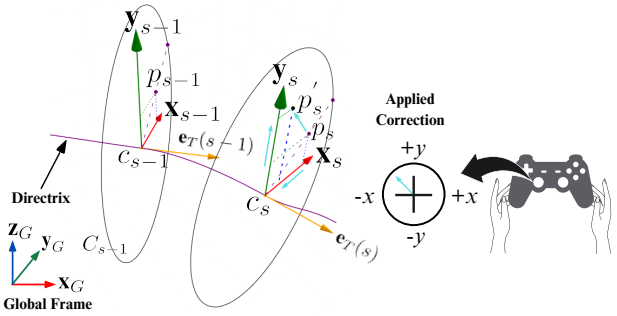


Figure 3: Applying a correction on C_s (blue arrow on the joystick). The x and y axes on the joystick are mapped to corrections on C_s along $\mathbf{x}_s$ and $\mathbf{y}_s$ respectively. The blue arrows marked on C_s indicate how joystick input affects p_s to reach $p_s^{'}$. After the corrections, a new trajectory is computed from $p_s^{'}$ using the ratio rule.

5.2 Correction Frames

To increase interpretability, we align "correction x-axis" $\mathbf{x}_s$ as closely as possible with the fixed global x-axis $\mathbf{x}_G$, while ensuring continuity. We define $\mathbf{x}_G$ as the unit vector pointing toward the front of the robot from its base in the global frame. The "correction y-axis" $\mathbf{y}_s$, is made orthogonal to $\mathbf{x}_s$ and secures local alignment by maintaining continuity with the y-axes of previous disks. Although $\mathbf{y}_s$ is not explicitly aligned with a fixed global direction, it typically corresponds to left/right or up/down directions. These axes enable two-dimensional corrections over the full 2D space of the disk, that are both intuitive and stable.

Defining $\mathbf{x}_s$. We aim to balance a smooth transition from $\mathbf{x}_{s-1}$, while maintaining global interpretabiliy. To do so, for $s \geq 2$, we first blend $\mathbf{x}_G^{C_s}$ and $\mathbf{x}_{s-1}^{C_s}$, the projections on C_s of $\mathbf{x}_G$ and $\mathbf{x}_{s-1}$, using the following spherical linear interpolation (Slerp) equation:

$$\mathbf{x}_s = \begin{cases} \frac{\sin((1-t)\cdot\theta)\cdot\mathbf{x}_G^{C_s} + \sin(t\cdot\theta)\cdot\mathbf{x}_{s-1}^{C_s}}{\sin(\theta)}, & \text{if } \sin(\theta) > \epsilon \\ \mathbf{x}_{s-1} & \text{otherwise} \end{cases} \quad (1)$$

where $\theta = \cos^{-1}\left(\frac{\mathbf{x}_{s-1}^{C_s}}{\|\mathbf{x}_{s-1}^{C_s}\|} \cdot \frac{\mathbf{x}_G^{C_s}}{\|\mathbf{x}_G^{C_s}\|}\right)$ denotes the angular distance between the projected vectors, $t = \frac{\theta_X}{\pi/2}$ denotes and interpolation constant taking into account $\theta_X = \cos^{-1}\left(\frac{\mathbf{x}_G^{C_s}}{\|\mathbf{x}_G^{C_s}\|} \cdot \mathbf{x}_G\right)$ which characterizes the alignment of C_s with $\mathbf{x}_G$, and ϵ is a constant set to e^{-10}. However, other heuristics could be used to set the interpolation constant. The initial correction axis, $\mathbf{x}_1$, is computed by directly projecting $\mathbf{x}_G$ onto the plane of the first disk C_1.

Defining $\mathbf{y}_s$. To maintain orthogonality between $\mathbf{x}_s$ and $\mathbf{y}_s$, there are only two options for $\mathbf{y}_s$, and our goal is to maintain local alignment and smooth continuity. First, we generate a preliminary $\mathbf{y}_s^{'}$ by computing the cross product of $\mathbf{e}_T(s)$ and $\mathbf{x}_s$. This result tends to be sufficient most of the time, but in some situations, it needs to be flipped. For example, when moving up after having picked an object, $\mathbf{e}_T(s)$ would flip, while $\mathbf{x}_s$ maintains the same direction, resulting in a discontinuity between $\mathbf{y}_{s-1}$ and $\mathbf{y}_s^{'}$, leading to an unintuitive experience for users.

Therefore, we use a window-based strategy, averaging $\mathbf{y}$ over the previous 10 disks and projecting the mean vector onto C_s to derive $\mathbf{y}_\mu^{C_s}$. We then measure the angle θ_Y between $\mathbf{y}_\mu^{C_s}$ and $\mathbf{y}_s^{'}$ on Cs. If θ_Y exceeds $\frac{\pi}{2}$ radians, indicating a discontinuity, we invert $\mathbf{y}_s^{'}$ to match the historical $\mathbf{y}$ trend; otherwise, $\mathbf{y}_s^{'}$ remains unchanged. While this approach deviates from the right-hand rule, it enables the definition of an orthonormal correction frame on C_s, consisting of $\mathbf{e}_T(s)$, $\mathbf{x}_s$, and $\mathbf{y}_s$ with smoother transitions (see Fig. 2).

5.3 Integrating Orientation into Canal Surfaces

Although canal surfaces have been used to represent robotic trajectories [1, 2], they capture only position and omit orientation. While orientation is often critical, enforcing a fixed end-effector orientation can limit reachability or degrade position tracking [38]. Therefore, we compute the mean orientation from demonstrations for each disk as the target and use the standard deviation (SD) to define allowable variation. Details and behavioral effects are provided in §6.2. In GeoSACS, corrections currently do not affect orientation.

5.4 Backtracking for Repetitive Tasks

Inspired by [16], we allow trajectory reproduction in reverse to support repetitive tasks (e.g., returning to pick the next clothing item after placing one). Additionally, to improve interaction, trajectories heading toward the canal center use a convergent-ratio strategy with a decay function, whereas trajectories heading toward the canal ends (i.e., *pick* and *place* locations) use a fixed-ratio strategy. Following [1], we reset corrections between pick-and-place actions while allowing adjustments during each action.

6 Implementation

We implemented GeoSACS on the Lio robot [29] (F&P Robotics), an assistive mobile platform with a 6-DoF arm designed for eldercare and home environments, using a gamepad for corrections. Since canals are generated in Cartesian (not joint) space, the method naturally extends to robots with more than 6 DoF, as trajectories and corrections are decoupled from joint configurations.

6.1 Preprocessing Demonstrations

Recording. We record the 3D Cartesian position and the orientation of the robot's end effector. Demonstrations start from the pickup location, and end at the placing location.

Preprocessing Position Data. Demonstrations may vary in length, so we use a two-step method to align them and ensure smooth canal generation. First, we apply dynamic time warping (DTW) [14] for temporal alignment. Then, we reduce noise from overshoots (e.g., unintended movements during demonstrations) using a cubic spline with a step filter, with a step size h set to 10% of the trajectory length. We sample points at this step size, fit a cubic spline, and resample to produce a smoothed trajectory with $N_f = h \times L_{\max}$, where $L_{\max}$ is the length of the longest demonstration.

Preprocessing Orientation Data. After applying the DTW algorithm for positional data, we extract quaternion-based orientation

values corresponding to the same filtered trajectory points. Using the same step size h, we sample orientation values and interpolate them using Catmull-Rom quaternion interpolation[1], followed by resampling. The final dataset T_f consists of n demonstrations (in our case, $n = 2$), each with N_f points, represented as $\phi^{7\times N_f\times n}$, where ϕ encodes the combined position and orientation data for each trajectory point across demonstrations.

6.2 Canal Surface Generation

After obtaining the pre-processed data, the next step is to generate the canal surface.

Directrix. To find the directrix Γ, where the centers of all disks lie, we calculate the mean value μ_p across the positional coordinates of T_f, where $\mu_p \in \mathbb{R}^{3\times N_f}$.

Radii. Following previous work [2], we then calculate the radius at a specific directrix point by measuring the distances between this point and the corresponding points on the filtered trajectories in T_f. We designate the largest of these distances as the radius for the focal point, defining the permissible action space at that location. Afterwards, we find the correction frames as detailed in §5.2.

Orientation Measures. We obtain the mean quaternion $\bar{\mathbf{q}}_s$ and associated SD σ_{q_s} for C_s by leveraging the processed demonstration data in T_f. We use an Eigen value-based method to determine the mean [41], and a measure of absolute distance to ascertain the SD[2] across the demonstrations. This SD is then used in our inverse kinematics (IK), with further details provided in §6.3

Cross Section Refinements. Because demonstrations are kinesthetic, canal geometry over horizontal surfaces (e.g., tables or floors) can produce tangent vectors that are not perfectly orthogonal, causing disks to extend below the surface and prompting the robot to reach unintended regions. To address this, we use the lowest point in the demonstrations to detect such cases, reorient the affected disks horizontally, and adjust their correction frames accordingly.

6.3 Motion Generation

Our optimization-based IK engine, inspired from RelaxedIK [38], seeks a joint configuration $\boldsymbol{\Theta}$ that balances accurate end-effector position ($\mathbf{p}_s$) and orientation ($\bar{\mathbf{q}}_s$) by minimizing the cost function $J(s) = w_\mathrm{p}\cdot J_\mathrm{p}(\mathbf{p}_s,\boldsymbol{\Theta}) + w_\mathrm{o}(s)\cdot J_\mathrm{o}(\bar{\mathbf{q}}_s,\boldsymbol{\Theta})$, where J_p and J_o are the cost functions for positional and orientational errors, and w_p (=100) and $w_\mathrm{o}(s)$ are the weighting factors for each cost component.

For $w_\mathrm{o}(s)$, we assign a higher weight for a lower SD in the demonstrations, indicating a narrow range of orientation variability, and a lower weight for a higher SD, suggesting less stringent constraints on orientation, using $w_\mathrm{o}(s) = e^{-\alpha(\sigma_{q_s}-\beta)}$, where α and β are empirically determined to be of values 9 and 0.3 respectively, ensuring that $w_\mathrm{o}(s)$ falls within the specific range $[0, 15]$ to keep the balance between position and orientation while $w_\mathrm{o}(s)$ changes.

7 User study

We conducted a within-subject study with two tasks, three conditions, and 24 participants to evaluate two key assumptions of GeoSACS: (1) that robot-led control reduces user workload compared to human-led control, and (2) that the variability in orientation of Bishop frames reduces control intuitivity.

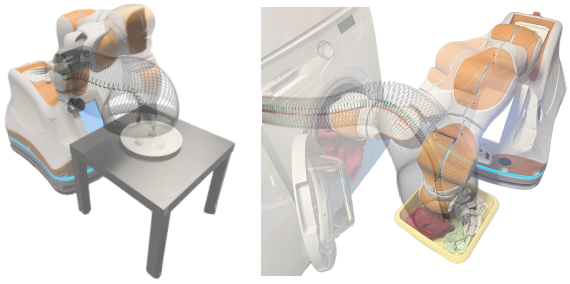


Figure 4: Setup for marshmallow task (left) and laundry loading task (right) including the generated canals.

7.1 Tasks

Task 1 - Assistive Eating. Our first task is an assistive feeding task inspired by the HARMONIC dataset [30] (see Fig. 4). In this task, the user guides the robot to select and spear a marshmallow of their choice out of three marshmallows placed on a plate.

Task 2 - Laundry Loading. The second task demonstrates our method in a routine household activity, specifically loading a washing machine (see Fig. 4) which involves 6D interactions. Users had to place as many of five clothing items as possible from a basket into the machine within a set time.

7.2 Conditions

As part of assessing the assumptions underlying GeoSACS, we included two baselines: (1) a latent actions baseline [26] to evaluate the influence of control initiation, and (2) canal surfaces with Bishop frames (referred to as "Bishop canals") to evaluate how frame alignment affects control. Conceptually, these three conditions fall into two categories: latent actions map 2D user inputs directly to 6D robot motions, whereas canal-based methods (Bishop and GeoSACS) split control of the 6D control space into an autonomous component (navigation along the canal) and a 2D user correction subspace. However, they differ in the correction frame: Bishop canals uses the Bishop frame, whereas GeoSACS aligns the correction axes with the global axes as proposed. More specifically, we have:

Latent Actions. In the marshmallow task, we provided separate demonstrations to illustrate two movement axes over the plate, allowing forward, backward, left, and right motions. The initial motion to reach the plate was fixed, following [26]. For laundry loading, we used a similar strategy over the basket area, with additional demonstrations for transitions to the laundry drum. We collected multiple short demonstrations (~4 min each) over ~8 hours, then retained three high-quality sets (~53,000 state–action pairs) that produced stable behavior and covered the full workspace. We trained a conditional autoencoder (CAE) to map 12D state–action inputs to a 2D latent space (12→30→30→2) and reconstruct actions conditioned on state (8→30→30→6).

Bishop Canals. We provided two demonstrations for each of the tasks, showcasing the boundaries of the task's operating space to

[1]Splines library was employed for Catmul-Rom quaternion interpolation: https://pypi.org/project/splines/

[2]Pyquaternion library was utilized to find the SD of orientation within a disk: https://kieranwynn.github.io/pyquaternion/

generate the canals. The two demonstrations used are not identical but lie within the same demonstration workspace as in latent actions. During execution, the robot navigates the canal autonomously, and users can provide corrections using the Bishop frame as the correction frame.

GeoSACS. The same demonstrations used for Bishop canals were also used with GeoSACS. During execution, the robot navigates the canal autonomously, and users can provide corrections using our own frame as the correction frame.

7.3 Participants

We recruited 24 abled participants (sixteen male, eight female) aged 19-57 (M = 31.75, SD = 8.47) primarily from our research institution, but covering both technical (eighteen) and administrative (six) staff. The study protocol was reviewed and approved by our institutional ethics committee. Prior experience with robots and gamepads was measured using a 5-point Likert scale (1-5), where lower values indicate less experience. The mean gamepad experience was 3.25 ± 0.92, while the mean robot control experience was 1.62 ± 1.03.

7.4 Setup

Similar to other SA work, we used an Xbox gamepad as the control interface due to its flexibility, mobility, and familiarity. The left joystick directed the robot within latent dimensions for latent actions, and along correction axes for the canal-based methods. One button initiated movement; in the marshmallow task, another button lowered the robot to spear the marshmallow. In the laundry task, a button controlled the gripper across all methods. For canal-based methods only, an extra button changed direction within the canal.

7.5 Procedure

We used a within-subjects design with full counterbalancing of the three conditions to reduce ordering effects. The study began with an introduction, consent, and a demographic survey. For each condition, participants completed the marshmallow task with five minutes of training followed by five trials (following the HARMONIC dataset procedure). Next, they performed the laundry task with two minutes of training and five minutes to load as many clothing items as possible. After each condition, participants completed a questionnaire. This process was repeated for all three conditions. The study concluded with a semi-structured interview, debriefing, and compensation of CHF 31.25, lasting approximately 1.5 hours.

7.6 Metrics

Objective. For the marshmallow task, we measured completion time (time to reach the desired marshmallow) and input time (time spent providing inputs), reporting the mean and SD over five trials. We also computed success rate as the number of trials in which the user reached the intended marshmallow. For the laundry task, we imposed a five-minute time limit to avoid prolonged sessions, which was often insufficient to complete the task. Therefore, for this task, we measured only input time, and not the completion time. We also used a point-based scoring system (maximum 10 points): 2 points if the cloth was fully placed inside the laundry drum, 1 point if partially hanging from the opening, and 0 points if it remained in the basket or ended up in an unreachable area for the robot.

Subjective. After completing both tasks for each method, participants filled out a questionnaire including 7-point Likert items from the USE questionnaire [27] (ease of use, ease of learning, satisfaction) and NASA-TLX [18] for workload. At the end, we conducted a semi-structured interview to gather feedback on usability, efficiency, perceived safety, workload, and opinions on the methods.

7.7 Hypotheses

Our hypotheses are as follows.

H1: Canal surface based SA is more efficient compared to latent actions based SA. To evaluate this hypothesis we used all the objective and subjective metrics mentioned in §7.6.

H2: Participants will prefer GeoSACS over latent actions or Bishop canals. It should be noted that we anticipated similar objective and subjective scores for GeoSACS and Bishop canals given their similarities aside from frame alignment. Therefore, to evaluate this hypothesis, we asked participants during the semi-structured interview to rank the three methods based on their experience.

Table 1: Completion time (CT), input time (IT), marshmallow success rate (MS), and laundry points (LP) for the three evaluated methods, p-values were calculated using repeated measures ANOVA.

Task	Metric	Latent Actions	Bishop	GeoSACS	p-Val
		Mean (SD)	**Mean** (SD)	**Mean** (SD)	
Marshmallow	CT	**36.68** (16.23)	**13.87** (2.09)	**13.03** (1.65)	< 0.001
	IT	**20.60** (12.95)	**4.56** (1.51)	**4.04** (0.74)	< 0.001
	MS	**4.29** (0.75)	**5.00** (0.00)	**5.00** (0.00)	< 0.001
Laundry	IT	**218.84** (40.60)	**56.45** (17.02)	**54.97** (19.53)	< 0.001
	LP	**4.41** (1.99)	**6.16** (1.90)	**6.70** (1.16)	< 0.001

7.8 Results

Performance. Table 1 summarizes the metrics: completion time, input time, marshmallow success rate, and laundry points. Due to a collision-related technical issue during laundry loading with latent actions, we excluded one participant's data when computing the mean and SD for laundry input time. Since repeated measures ANOVA indicated a significant difference among the methods for all metrics, we conducted post-hoc tests to localize these effects. Normality was first assessed using the Shapiro–Wilk test for each metric. Consequently, we applied Wilcoxon signed-rank test with Holm-Bonferroni correction for the marshmallow task's completion time, input time, success rate, and the laundry task's point evaluation, and paired t-test with Holm-Bonferroni correction for the laundry task's input time. Both canal-based methods significantly outperformed latent actions across all five metrics ($p < 0.001$), except for laundry points, where the differences were still significant (Latent vs. Bishop: $W = 9.0$, $p < 0.01$; Latent vs. GeoSACS: $W = 22.5$, $p < 0.01$). Although GeoSACS achieved marginally better results, no statistically significant difference was found between Bishop canals and GeoSACS, which is expected as they share conceptual similarity. During the laundry loading task with latent actions, 12 participants experienced between one and four collisions with the laundry machine, which required minor intervention by the experimenter to allow the task to proceed.

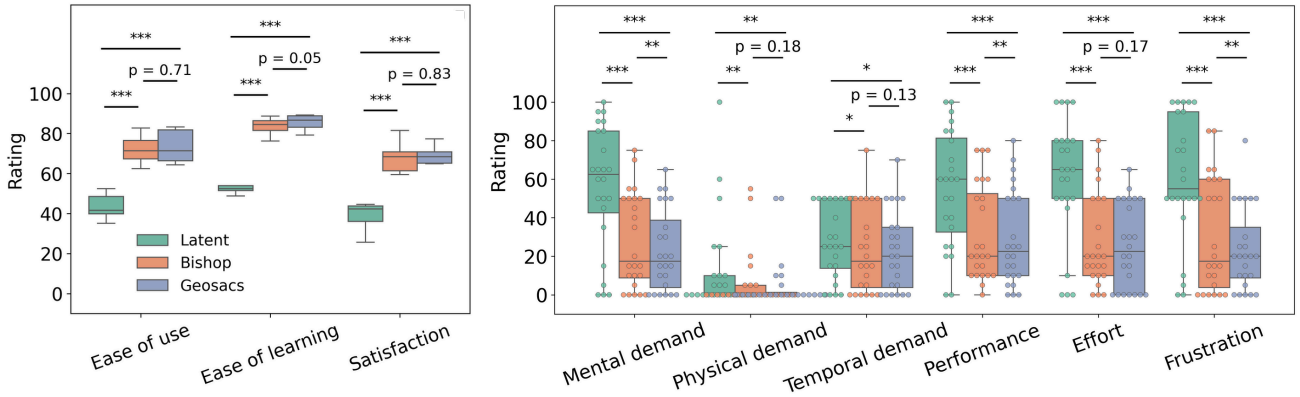


Figure 5: Left: Participant ratings for usability (higher is better): ease of use, ease of learning, and satisfaction, assessed with paired t-test. Right: Workload across the six NASA-TLX dimensions (lower is better), assessed with the Wilcoxon signed-rank test. The p-values were Holm–Bonferroni corrected [(*) denotes $p < 0.05$, (**) denotes $p < 0.01$, and (***) denotes $p < 0.001$].

Usability and Workload. Figure 5 presents the questionnaire results. Across both the USE and NASA-TLX dimensions, canal-based methods showed statistically significant improvements and outperformed latent actions. Combined with the objective results, these findings support hypothesis **H1**: our canal surface-based methods are more efficient than a latent action-based approach.

In NASA-TLX (lower is better), GeoSACS yielded statistically significant lower scores than Bishop canals on mental demand, performance, and frustration (Wilcoxon signed-rank, Holm–Bonferroni corrected), indicating reduced workload. Although group medians sometimes appear lower for Bishop canals, the median masks the extreme high-workload Bishop trials that occur when frequent corrections are needed; these extremes are captured by the means and the paired analysis, which both favor GeoSACS.

Beyond usability and workload measures, participants relied on GeoSACS's backtracking feature to push clothes fully into the drum, showing its practical value for real-world assistive tasks.

Interviews. In the semi-structured interview, participants ranked their preferred method. GeoSACS was ranked first by 13 participants, 5 preferred Bishop canals, 3 preferred latent actions, and 3 ranked both GeoSACS and Bishop canals first, noting similar effectiveness. We ran a Friedman test, which showed a significant difference among methods ($p < 0.001$). Followed with post-hoc Wilcoxon signed-rank tests with Holm-Bonferroni correction showed GeoSACS ($W = 21.5$, $p < 0.001$) and Bishop ($W = 59.5$, $p < 0.05$) were significantly preferred over latent actions, while no significant difference was observed between GeoSACS and Bishop canals, despite the trend towards significance. These results partially support hypothesis **H2**.

Remarks such as *"first two methods [GeoSACS and Bishop canals] were very similar and very intuitive to use"*, *"Honestly, that [GeoSACS] was the most intuitive"*, further emphasized that the canal-based methods were more intuitive to use. Also for both canal-based methods, participants often explored and refined motions rather than minimizing input time; several even described the interaction as "feeling like a game". The main comment participant made regarding latent actions centered around the difficulty of understanding the control mapping, with comments like *"I couldn't find a logic"*, *"it seems random"* and *"I wasn't able to understand what were the logic behind it"*. However, some participants noted that with additional practice, the method could become more usable, saying *"with more time it can get better"*, *"with practice it would be better"* and ranked it as their first preference highlighting its flexibility over canal-based methods (e.g., the ability to move outside the region constrained by the canal boundary).

Despite the similarities between Bishop canals and GeoSACS, several participants highlighted GeoSACS as being smoother and easier to control, stating *"I felt it was smoother"*, *"third one [GeoSACS] seemed quite smooth"*, and *"I find it quite fast to use"*. This showcases the value of our global and local alignment approach for the correction axes. Interestingly, a few participants even identified subtle differences in control behavior between Bishop canals and GeoSACS, specifically making remarks about Bishop canals such as: *"it was a little bit harder [...] felt like the plane was rotated 45 degrees. So when I moved the joystick forward, it would move diagonally [...] So there's like an extra cognitive load to adapt to that."*

8 Discussion

In this section, we first summarize the key findings from our user study and then outline the design guidelines derived from the results and participant feedback to inform future research in SA.

8.1 Results Discussion

Participants found GeoSACS and Bishop canals easier and more pleasant to use than latent actions. Both canal-based methods outperformed latent actions on all subjective measures, and GeoSACS showed significant improvements over Bishop canals in mental demand, perceived performance, and frustration. Other subjective dimensions showed similar trends. Overall, GeoSACS tended to achieve superior outcomes, suggesting aligning correction axes with a global reference frame enables smoother and faster control.

Beyond these findings, one outcome of this work is to show that SA is possible without complex vision systems or many demonstrations. A second outcome is to show that interpretable high-dimensional robot control can be achieved with low-dimensional, commonly available interfaces by partitioning the control space between robot autonomy and user input. In GeoSACS, this interpretability comes from task structure and the robot's natural direction of motion.

8.2 Design Recommendations for SA

From our user study, we identified several design recommendations that can be further explored and adopted within the SA domain to better support the assistive robotics community.

(1) Robot-led approaches such as GeoSACS are better suited for repetitive tasks with strong structural constraints, where timely corrections are not critical. For repetitive motions with predictable structure, such as laundry loading or item stacking, robot-led approaches like GeoSACS may be more appropriate, as users can offload timing control to the robot and intervene only when needed. While GeoSACS achieved better performance in our study, some participants noted they might prefer latent actions with more training due to their flexibility.

This highlights the value of human-led methods, which are useful when timing and flexibility matter, such as robotic surgery, where experts must retain initiative while the robot assists. Latent actions have also been evaluated against SA methods with varying levels of robot assistance [22] and were found to outperform them in the marshmallow task [26]. Our findings suggest that canal surface-based approaches are a complementary alternative to latent spaces. Future work could explore combining the two: latent spaces for flexible nominal behaviors and geometry for interpretable correction channels. Moreover, control can be viewed as a continuum, where decreasing robot speed increases human initiative, while increasing it shifts initiative to the robot.

(2) Structural decoupling of the robot action space enables more comprehensible input mappings for users. Latent actions compress the entire 6D action space into a latent representation, allowing low-dimensional inputs to influence all degrees of freedom simultaneously. While expressive, this can introduce unnecessary couplings that feel unintuitive for simple, linear corrections, as reflected in user feedback describing random movements. In contrast, GeoSACS structures control by having the robot regulate the 6D space while users provide corrections on 2D planes orthogonal to the motion, making the mapping more predictable and intuitive.

(3) User-driven orientation control can be enabled through task-relevant direction vectors. Orientation becomes more intuitive when the end-effector's pointing direction relative to objects in the scene is used as a natural reference. This insight emerged from our observations in the laundry task, where the robot's motion was orthogonal to the 2D correction planes and the tangent vectors of these planes naturally pointed toward the goal poses (e.g., the basket or laundry drum). Allowing users to adjust this direction vector provides a straightforward way to specify grasping orientations when reaching target objects, and future systems could leverage vision-based perception to define such vectors directly, reducing the need for kinesthetic demonstrations.

9 Limitations and Future Work

Despite the successful demonstration of feasibility, our approach and the study have some limitations.

9.1 Method Limitations

Cross-Section Refinements and Correction Basis Design. Although our refinements generalize to preventing collisions with horizontal surfaces, they can introduce discontinuities in disk orientation and do not apply to vertical surfaces. We aim to develop an approach that works for both by refining angular transitions across adjacent disks. Currently, we use an orthonormal correction basis for simplicity and interpretability, and to isolate the effect of canal surfaces on SA. Future work could explore adaptive or non-orthogonal bases, especially with non-uniform cross-sections [1] to match task-specific variability. Another direction is to learn the correction basis (e.g., via PCA per section), though this would require more demonstrations and careful handling of discontinuities.

Orientation, Speed, and Impedance Control. Although positional control was sufficient, participants wanted direct orientation control. Future work will therefore add user orientation control (e.g., via a second joystick, voice, or other modalities). Participants also suggested controls to adjust speed and pause autonomous motion via a dedicated button. Another future direction is force interaction via impedance control.

Demonstrations. While our preprocessing produced smooth, usable canal surfaces despite minor demonstration noise, SA systems should not assume high-quality demonstrations. Future work should evaluate how demonstration quality affects controllability. We also plan to use vision to automatically generate canal surfaces, removing the need for demonstrations.

9.2 Study Limitations

Target Population. Users with disabilities were not included in this phase, as our goal was to understand the impact of GeoSACS design decisions. However, in [37], we present an extension of this method that is grounded in users' perspectives on input mapping and focuses on accessibility with wheelchair users; future work will include participatory design sessions to better address their needs.

Conditions. Our study only compared GeoSACS against two baselines and only evaluated it on two task types. To avoid prolonged user studies that could frustrate participants, we limited the scope in this way. Nevertheless, future work could incorporate additional baseline methods for comparison with GeoSACS.

10 Conclusion

We presented GeoSACS, a geometric SA framework that uses canal surfaces to encode variability in repetitive tasks. In a user study, participants used GeoSACS to control a robot in household tasks, such as laundry loading, with only two demonstrations, highlighting its usability. Results showed a clear preference for GeoSACS over the latent actions baseline, supporting the assumption that allocating greater robot initiative can reduce workload. Moreover, the study indicated that aligning correction frames with global axes, rather than Bishop frames, leads to a better control experience, as reflected in subjective measures, even though overall task performance remained comparable.

References

[1] S. Reza Ahmadzadeh and Sonia Chernova. 2018. Trajectory-Based Skill Learning Using Generalized Cylinders. *Frontiers in Robotics and AI* 5 (2018).

[2] S. Reza Ahmadzadeh, Roshni Kaushik, and Sonia Chernova. 2016. Trajectory learning from demonstration with canal surfaces: A parameter-free approach. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*.

[3] Brenna Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. 2009. A survey of robot learning from demonstration. *Robotics and Autonomous Systems* 57 (05 2009).

[4] Tapomayukh Bhattacharjee, Ethan K. Gordon, Rosario Scalise, Maria E. Cabrera, Anat Caspi, Maya Cakmak, and Siddhartha S. Srinivasa. 2020. Is More Autonomy Always Better? Exploring Preferences of Users with Mobility Impairments in Robot-assisted Feeding. In *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction* (Cambridge, United Kingdom) *(HRI '20)*. Association for Computing Machinery, New York, NY, USA, 181–190. doi:10.1145/3319502.3374818

[5] Richard Bishop. 1975. There is More than One Way to Frame a Curve. *The American Mathematical Monthly* 82 (03 1975), Mathematical Association of America–. doi:10.2307/2319846

[6] Alexander Broad, Jacob Arkin, Nathan Ratliff, Thomas Howard, and Brenna Argall. 2017. Real-time natural language corrections for assistive robotic manipulators. *Int. J. Rob. Res.* 36, 5–7 (June 2017), 684–698. doi:10.1177/0278364917706418

[7] Steven W Brose, Douglas J Weber, Ben A Salatin, Garret G Grindle, Hongwu Wang, Juan J Vazquez, and Rory A Cooper. 2010. The role of assistive robotics in the lives of persons with disability. *American Journal of Physical Medicine & Rehabilitation* 89, 6 (2010).

[8] Sylvain Calinon. 2016. A Tutorial on Task-Parameterized Movement Learning and Retrieval. *Intelligent Service Robotics* 9 (01 2016). doi:10.1007/s11370-015-0187-9

[9] Sylvain Calinon, Florent Guenter, and Aude Billard. 2007. On Learning, Representing, and Generalizing a Task in a Humanoid Robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 37, 2 (2007). doi:10.1109/TSMCB.2006.886952

[10] Harish chaandar Ravichandar, Athanasios S. Polydoros, Sonia Chernova, and Aude Billard. 2020. Recent Advances in Robot Learning from Demonstration. *Annu. Rev. Control. Robotics Auton. Syst.* 3 (2020).

[11] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. 2025. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research* 44, 10-11 (2025), 1684–1704. arXiv:https://doi.org/10.1177/02783649241273668 doi:10.1177/02783649241273668

[12] Yuchen Cui, Siddharth Karamcheti, Raj Palleti, Nidhya Shivakumar, Percy Liang, and Dorsa Sadigh. 2023. No, to the Right: Online Language Corrections for Robotic Manipulation via Shared Autonomy. In *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction* (Stockholm, Sweden) *(HRI '23)*. Association for Computing Machinery, New York, NY, USA, 9 pages. doi:10.1145/3568162.3578623

[13] B. P. DeJong, J. E. Colgate, and M. A. Peshkin. 2011. *Mental Transformations in Human-Robot Interaction*. Springer Netherlands, Dordrecht.

[14] Toni Giorgino. 2009. Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package. *Journal of Statistical Software* 31, 7 (2009). doi:10.18637/jss.v031.i07

[15] Michael Hagenow, Emmanuel Senft, Nitzan Orr, Robert Radwin, Michael Gleicher, Bilge Mutlu, Dylan P. Losey, and Michael Zinn. 2023. Coordinated Multi-Robot Shared Autonomy Based on Scheduling and Demonstrations. *IEEE Robotics and Automation Letters* 8, 12 (2023), 8335–8342. doi:10.1109/LRA.2023.3327625

[16] Michael Hagenow, Emmanuel Senft, Robert Radwin, Michael Gleicher, Bilge Mutlu, and Michael Zinn. 2021. Corrective Shared Autonomy for Addressing Task Variability. *IEEE Robotics and Automation Letters* 6, 2.

[17] Michael Hagenow, Emmanuel Senft, Robert Radwin, Michael Gleicher, Bilge Mutlu, and Michael Zinn. 2021. Informing Real-Time Corrections in Corrective Shared Autonomy Through Expert Demonstrations. *IEEE Robotics and Automation Letters* 6, 4 (2021), 6442–6449. doi:10.1109/LRA.2021.3094480

[18] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Human Mental Workload*, Peter A. Hancock and Najmedin Meshkati (Eds.). Advances in Psychology, Vol. 52. North-Holland.

[19] Laura M. Hiatt and Reid Simmons. 2006. Coordinate Frames in Robotic Teleoperation. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. doi:10.1109/IROS.2006.282130

[20] D. Hilbert and S. Cohn-Vossen. 1953. Geometry and the Imagination. *Physics Today* 6, 5 (05 1953), 19–19.

[21] A.J. Ijspeert, Jun Nakanishi, Heiko Hoffmann, Peter Pastor, and Stefan Schaal. 2012. Dynamical Movement Primitives: Learning Attractor Models for Motor Behaviors. *Neural computation* 25 (11 2012).

[22] Shervin Javdani, Henny Admoni, Stefania Pellegrinelli, Siddhartha S. Srinivasa, and J. Andrew Bagnell. 2018. Shared autonomy via hindsight optimization for teleoperation and teaming. *Int. J. Rob. Res.* 37, 7 (June 2018), 717–742. doi:10.1177/0278364918776060

[23] S. Mohammad Khansari-Zadeh and Aude Billard. 2011. Learning Stable Nonlinear Dynamical Systems With Gaussian Mixture Models. *IEEE Transactions on Robotics* 27, 5 (2011). doi:10.1109/TRO.2011.2159412

[24] Riddhiman Laha, Anjali Rao, Luis F. C. Figueredo, Qing Chang, Sami Haddadin, and Nilanjan Chakraborty. 2021. Point-to-Point Path Planning Based on User Guidance and Screw Linear Interpolation *(International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. Volume 8B: 45th Mechanisms and Robotics Conference (MR))*. V08BT08A010.

[25] Zihao Li, Fugui Xie, Yanlei Ye, Peng Li, and Xinjun Liu. 2022. A Novel 6-DOF Force-Sensed Human-Robot Interface for an Intuitive Teleoperation. *Chinese Journal of Mechanical Engineering* 35 (12 2022). doi:10.1186/s10033-022-00813-1

[26] Dylan P. Losey, Krishnan Srinivasan, Ajay Mandlekar, Animesh Garg, and Dorsa Sadigh. 2020. Controlling Assistive Robots with Learned Latent Actions. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. doi:10.1109/ICRA40945.2020.9197197

[27] Arnold Lund. 2001. Measuring Usability with the USE Questionnaire. *Usability and User Experience Newsletter of the STC Usability SIG* 8 (01 2001).

[28] M. M. Marinho, R. Oikawa, K. Hayashi, S. Takazawa, K. Harada, and M. Mitsuishi. 2023. Design and Validation of Looping Assistance Methods in Robotic-Assisted Neonatal Surgical Suturing in a Chest Model. *The International Journal of Medical Robotics and Computer Assisted Surgery (IJMRCAS)* 19, 2 (April 2023), e2476. doi:10.1002/rcs.2476

[29] Justinas Mišeikis, Pietro Caroni, Patricia Duchamp, Alina Gasser, Rastislav Marko, Nelija Mišeikienė, Frederik Zwilling, Charles de Castelbajac, Lucas Eicher, Michael Früh, and Hansruedi Früh. 2020. Lio-A Personal Robot Assistant for Human-Robot Interaction and Care Applications.

[30] Benjamin A. Newman, Reuben M. Aronson, Siddhartha S. Srinivasa, Kris Kitani, and Henny Admoni. 2018. HARMONIC: A multimodal dataset of assistive human–robot collaboration. *The International Journal of Robotics Research* 41 (2018).

[31] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J. Andrew Bagnell, Pieter Abbeel, and Jan Peters. 2018. An Algorithmic Perspective on Imitation Learning. *Foundations and Trends® in Robotics* 7, 1-2 (2018). doi:10.1561/2300000053

[32] Akhil Padmanabha, Janavi Gupta, Chen Chen, Jehan Yang, Vy Nguyen, Douglas J Weber, Carmel Majidi, and Zackory Erickson. 2024. Independence in the Home: A Wearable Interface for a Person with Quadriplegia to Teleoperate a Mobile Manipulator. In *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction* (<conf-loc>, <city>Boulder</city>, <state>CO</state>, <country>USA</country>, </conf-loc>) *(HRI '24)*. Association for Computing Machinery, New York, NY, USA, 542–551. doi:10.1145/3610977.3634964

[33] Alexandros Paraschos, Christian Daniel, Jan R Peters, and Gerhard Neumann. 2013. Probabilistic Movement Primitives. In *Advances in Neural Information Processing Systems*, C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger (Eds.), Vol. 26. Curran Associates, Inc.

[34] Pragathi Praveena, Luis Molina, Yeping Wang, Emmanuel Senft, Bilge Mutlu, and Michael Gleicher. 2022. Understanding Control Frames in Multi-Camera Robot Telemanipulation. In *Proceedings of the 2022 ACM/IEEE International Conference on Human-Robot Interaction* (Sapporo, Hokkaido, Japan) *(HRI '22)*. IEEE Press, 9 pages.

[35] Michael Przystupa, Kerrick Johnstonbaugh, Zichen Zhang, Laura Petrich, Masood Dehghan, Faezeh Haghverd, and Martin Jagersand. 2023. Learning State Conditioned Linear Mappings for Low-Dimensional Control of Robotic Manipulators. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 857–863. doi:10.1109/ICRA48891.2023.10160585

[36] Gabriel Quere, Freek Stulp, David Filliat, and João Silvério. 2024. A probabilistic approach for learning and adapting shared control skills with the human in the loop. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*. 15728–15734. doi:10.1109/ICRA57147.2024.10610956

[37] Shalutha Rajapakshe, Jean-Marc Odobez, and Emmanuel Senft. 2025. Giving Sense to Inputs: Toward an Accessible Control Framework for Shared Autonomy. In *Proceedings of the 2025 ACM/IEEE International Conference on Human-Robot Interaction* (Melbourne, Australia) *(HRI '25)*. IEEE Press, 213–221.

[38] Daniel Rakita, Bilge Mutlu, and Michael Gleicher. 2018. RelaxedIK: Real-time Synthesis of Accurate and Feasible Robot Arm Motion. *Robotics: Science and Systems XIV* (2018).

[39] Mario Selvaggio, Marco Cognetti, Stefanos Nikolaidis, Serena Ivaldi, and Bruno Siciliano. 2021. Autonomy in Physical Human-Robot Interaction: A Brief Survey. *IEEE Robotics and Automation Letters* 6, 4 (2021). doi:10.1109/LRA.2021.3100603

[40] TRI LBM Team, Jose Barreiros, Andrew Beaulieu, Aditya Bhat, Rick Cory, Eric Cousineau, Hongkai Dai, Ching-Hsin Fang, Kunimatsu Hashimoto, Muhammad Zubair Irshad, Masha Itkina, Naveen Kuppuswamy, Kuan-Hui Lee, Katherine Liu, Dale McConachie, Ian McMahon, Haruki Nishimura, Calder Phillips-Grafflin, Charles Richter, Paarth Shah, Krishnan Srinivasan, Blake Wulfe, Chen Xu, Mengchao Zhang, Alex Alspach, Maya Angeles, Kushal Arora, Vitor Campagnolo Guizilini, Alejandro Castro, Dian Chen, Ting-Sheng Chu, Sam Creasey, Sean Curtis, Richard Denitto, Emma Dixon, Eric Dusel, Matthew Ferreira, Aimee Goncalves, Grant Gould, Damrong Guoy, Swati Gupta, Xuchen Han, Kyle Hatch,

Brendan Hathaway, Allison Henry, Hillel Hochsztein, Phoebe Horgan, Shun Iwase, Donovon Jackson, Siddharth Karamcheti, Sedrick Keh, Joseph Masterjohn, Jean Mercat, Patrick Miller, Paul Mitiguy, Tony Nguyen, Jeremy Nimmer, Yuki Noguchi, Reko Ong, Aykut Onol, Owen Pfannenstiehl, Richard Poyner, Leticia Priebe Mendes Rocha, Gordon Richardson, Christopher Rodriguez, Derick Seale, Michael Sherman, Mariah Smith-Jones, David Tago, Pavel Tokmakov, Matthew Tran, Basile Van Hoorick, Igor Vasiljevic, Sergey Zakharov, Mark Zolotas, Rares Ambrus, Kerri Fetzer-Borelli, Benjamin Burchfiel, Hadas Kress-Gazit, Siyuan Feng, Stacie Ford, and Russ Tedrake. 2025. A Careful Examination of Large Behavior Models for Multitask Dexterous Manipulation. arXiv:2507.05331 [cs.RO] https://arxiv.org/abs/2507.05331

[41] Stephen Thompson, Thomas Dowrick, Mian Ahmad, Goufang Xiao, Bongjin Koo, Ester Bonmati, Kim Kahl, and Matthew J. Clarkson. 2020. SciKit-Surgery: compact libraries for surgical navigation. *International Journal of Computer Assisted Radiology and Surgery* 15, 7 (01 Jul 2020). doi:10.1007/s11548-020-02180-5

[42] Maxim Vochten, Ali Mousavi Mohammadi, Arno Verduyn, Tinne De Laet, Erwin Aertbeliën, and Joris De Schutter. 2023. Invariant Descriptors of Motion and Force Trajectories for Interpreting Object Manipulation Tasks in Contact. *Trans. Rob.* 39, 6 (Dec. 2023), 4892–4912. doi:10.1109/TRO.2023.3309230

[43] Dan-Dan Zhang, Zicong Wu, Junhong Chen, Ruiqi Zhu, Adnan Munawar, Bo Xiao, Yuan Guan, Hang Su, Wuzhou Hong, Yao Guo, Gregory Fischer, Benny Lo, and Guang-Zhong Yang. 2022. Human-Robot Shared Control for Surgical Robot Based on Context-Aware Sim-to-Real Adaptation. 7694–7700. doi:10.1109/ICRA46639.2022.9812379

[44] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. 2023. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In *Proceedings of Robotics: Science and Systems*. Daegu, Republic of Korea. doi:10.15607/RSS.2023.XIX.016

Received 2025-09-30; accepted 2025-12-01