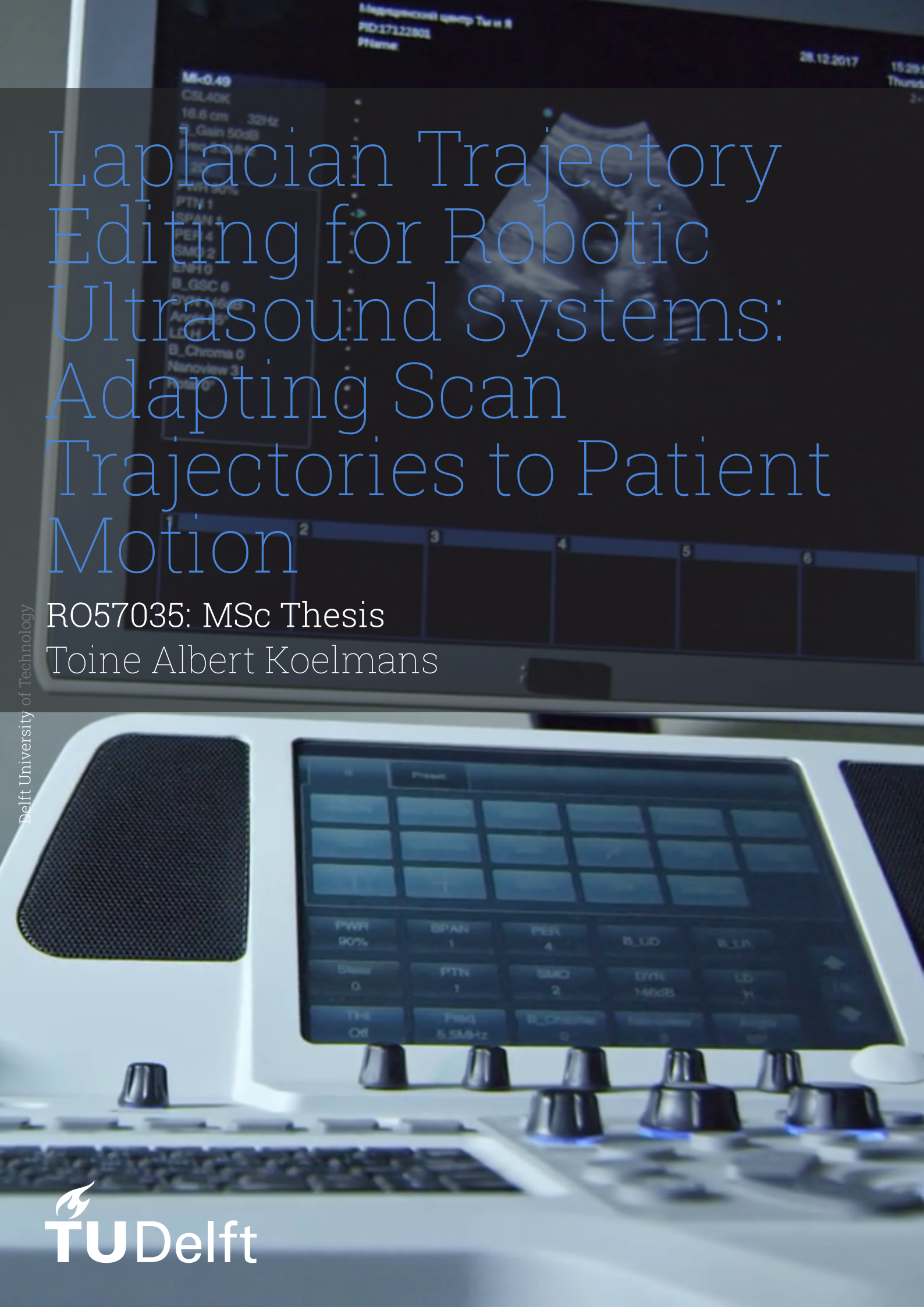
Laplacian Trajectory Editing for Robotic Ultrasound Systems: Adapting Scan Trajectories to Patient Motion

RO57035: MSc Thesis

Toine Albert Koelmans

Delft University of Technology



TU Delft

Laplacian Trajectory Editing for Robotic Ultrasound Systems: Adapting Scan Trajectories to Patient Motion

by

Toine Albert Koelmans

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended on March 13th, 2025.

Supervisors: Dr. J.M. Prendergast, N. Mol, MSc
Department: Cognitive Robotics (CoR)
Faculty: Faculty of Mechanical Engineering (ME)

TU Delft

Preface

This thesis marks the end of six and a half years at TU Delft. The past ten months have been an intense yet rewarding period. It was my first academic journey and also the first project where I had no clear idea of where it would lead. This often brought uncertainty about the next steps, but along the way, I learned that not knowing the full path is simply part of the process.

I would like to sincerely thank Dr. J.M. Prendergast and N. Mol, MSc., for guiding me through this unfamiliar process. Micah, your encouragement—especially the frequent reassurance that “you are in good shape”—helped me navigate moments of doubt. As the project took shape, our goals became clearer, and I am very pleased with the outcome. Your availability, constructive feedback, and clear and pleasant communication style made a significant difference throughout this project. Nicky, thank you for your help over the past three months, not only with the technical tasks but also for being a bit more pragmatic when things just needed to get done. This really helped with the final sprint and in concretizing this report.

Additionally, I would like to thank Giovanni. You were not part of my supervisor team, but just through our conversations, a large part of the conceptual work in this paper came to life.

Finally, I want to thank Thomas and my father for being the only ones outside the Thesis Committee to read parts of this paper. I truly appreciate your time and insights.

Toine Albert Koelmans
Delft, March 2025

Contents

1

Paper Submission

Laplacian Trajectory Editing for Robotic Ultrasound Systems: Adapting Scan Trajectories to Patient Motion

Toine Koelmans[1]

***Abstract*— Robotic Ultrasound Systems (RUSS) provide a promising solution to reduce operator dependency, alleviate physical strain, and meet the growing demand for ultrasound procedures. However, their clinical applicability remains limited by their inability to adapt to dynamic patient movements and tissue deformations during scans. This work introduces a novel RUSS framework that leverages Laplacian Trajectory Editing (LTE) for real-time adaptation of scan trajectories in response to both rigid and non-rigid patient movements. The system integrates an RGB-D camera to capture surface point clouds, which are processed to estimate displacements between consecutive frames. These displacements define anchor points for LTE-based trajectory adaptations, ensuring smooth motion while preserving local trajectory properties. The framework is validated through experiments spanning rigid phantom movements, generalization across differently shaped phantoms, and non-rigid human arm motion. Adaptation accuracy is quantified by comparing adapted trajectories to a ground-truth reference, with root mean squared errors averaging 0.026 ± 0.012 m in non-rigid scenarios. The system adapts scan trajectories in real-time, with LTE adaptation averaging 373 ms per trial. Furthermore, this RUSS achieved low tracking errors across all conditions while maintaining a high success rate in diverse movement scenarios. These results demonstrate the feasibility of LTE for real-time trajectory adaptation in ultrasound scanning, offering a pathway to more autonomous and clinically viable RUSS implementations.**

I. INTRODUCTION

MEDICAL imaging has become integral to modern healthcare, with ultrasound being the second most widely used modality [1]. Ultrasound is non-invasive, affordable, portable, and radiation-free. It is a powerful tool for analyzing internal organs and tissues, enabling early diagnosis and treatment selection. Still, its effectiveness depends heavily on the sonographer's skill making image quality and repeatability highly operator-dependent [2], [3]. This dependence becomes particularly evident in musculoskeletal ultrasound (MSK-US), where imaging often requires patients to move their joints during scanning, forcing sonographers to continuously adjust the probe to maintain diagnostic accuracy [4], [5]. Moreover, in recent years, the demand for ultrasound scans in general and those used to evaluate musculoskeletal disorders has outpaced the growth in the number of trained sonographers [6], [7]. In addition to these variations and shortages, performing an ultrasound procedure exerts a physical and cognitive burden on the sonographer, leading to pain and musculoskeletal disorders [8], [9]. Robotic Ultrasound Systems (RUSS) have gained attention over the last decades as a promising solution to address inter-operator variations, intra-operator discomfort and sonographer shortages [10]. Additionally, a RUSS's ability to reduce physical operator-patient contact became of even greater interest during the Covid-19 pandemic [11].

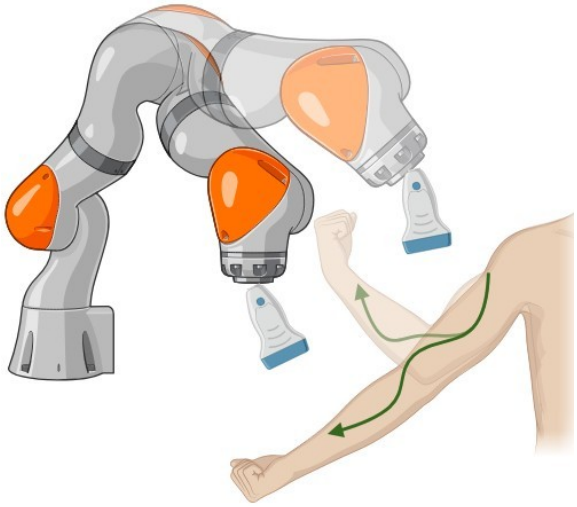


Fig. 1: Example of a Robotic Ultrasound System (RUSS) performing musculoskeletal ultrasound (MSK-US), where continuous patient movement is required for proper imaging. The patient bends their arm while the robot performs the scan, adapting its trajectory in real-time to maintain probe alignment.

To mitigate operator dependency, physical strain, and the rising demand for ultrasound, RUSS can be teleoperated, shared-controlled, or fully autonomous. While teleoperated and shared-control systems still rely on an operator, autonomous RUSS can function independently, reducing workload and ensuring consistent imaging. In the autonomous case, the system must determine a scan path, control probe orientation, and regulate contact force. In addition, it must address patient movement, tissue deformation, and other dynamic changes to the scan environment (Fig. 1). While patient movement affects all ultrasound applications, it is especially critical in MSK-US, where scanning inherently involves joint articulation. As patients flex and extend their limbs, underlying musculoskeletal abnormalities become visible. Achieving consistent imaging in these procedures requires real-time trajectory adaptation to maintain probe alignment. Regardless of the application, keeping the ultrasound

*This work was not supported by any organization

[1]The author is with the Cognitive Robotics Departments, Delft University of Technology, 2628, CD Delft, Netherlands (e-mail:

probe perpendicular to the surface ensures optimal ultrasound wave propagation, resulting in higher-quality images [12]. As a result, most RUSS aim to follow this principle, albeit through different means [13]–[15]. Maintaining appropriate contact force, typically between 1.2 N and 20 N depending on the application, is critical for effective imaging [10], [16], [17]. Several studies have proposed force control methods to stay within these acceptable force ranges for ultrasound tasks [18]–[20].

Moving beyond pure force or orientation control methods, previous work on autonomous RUSS can be broadly categorized into three main directions: Reinforcement Learning (RL), Learning from Demonstrations (LfD), and geometry-based or feedback-driven methods. RL-based approaches [21]–[30] optimize scan policies by interacting with the environment using predefined reward functions. LfD techniques [31]–[38] infer optimal policies from expert demonstrations through behavioral cloning or inverse reinforcement learning. Although these learning-based methods enable adaptation to new tasks, they often require large datasets and may lack interpretability. In contrast, geometry-based and feedback-driven methods [39]–[49] follow a rule-based approach using marker tracking, depth cameras, geometric modeling, and visual servoing to define and refine scan paths. These rule-based designs facilitate the implementation of safety measures and offer more predictable behavior. Previous methods have successfully used point cloud data to reconstruct patient anatomy, define scan trajectories preoperatively, and follow them for specific anatomical regions. Examples include straight-line scanning over a lumbar phantom [39], zig-zag coverage patterns for breast imaging [42], [44], and YOLO-Pose-based key point extraction to define Regions of Interest (RoIs) [46], [49].

Although effective in generating initial scan trajectories, these methods fail to adapt dynamically to **patient movement** or **tissue deformations** (Fig. 1). Several works have addressed these challenges, exploring different ways to account for deformation in RUSS. One approach is to correct the obtained ultrasound images for deformations [50], [51]. Another approach is to estimate tissue deformation and feedforward this offset to correct the planned scan path [45]. To address patient motion and specifically articulated arm movement, Jiang et al. previously used a non-rigid registration method to align arm surface point clouds with an atlas MRI scan [41]. This approach mapped predefined trajectories based on annotated vascular structures to different patients and joint configurations. Although well-suited for preoperative trajectory transfer, it lacked online adaptability. Jiang et al. did demonstrate a RUSS capable of online adaptation to rigid arm phantom motion using registration between RGB-D and preoperative CT point clouds [47]. Although this system compensated for movement and handled partial point clouds, it struggled with non-rigid deformations. More recently, Sun et al. proposed an Automated Robotic Ultrasound Scanning (ARUS) system that integrates force control, deep-learning-based segmentation, and 3D reconstruction for musculoskeletal ultrasound [52]. However, while ARUS enhances 3D imaging accuracy, it does not provide real-time trajectory adaptation for non-rigid motion or tissue deformations. Real-time adaptation to non-rigid patient movements remains a significant challenge for the clinical adoption of RUSS, especially as a means to perform MSK-US [4], [5].

A common approach to trajectory adaptation in other fields of robotics is the previously mentioned LfD. It enables robots to encode expert demonstrations by either directly reproducing observed movements or inferring underlying task objectives. This allows robots to generalize demonstrated trajectories to new conditions while preserving essential motion characteristics. Several LfD methods have been proposed in the literature. Dynamic Movement Primitives (DMPs) encode movements as dynamical systems with attractor states, allowing adaptation to varying start and goal positions [53]. Extensions like Probabilistic Movement Primitives (ProMPs) and Kernelized Movement Primitives (KMPs) improve flexibility by capturing variability and using kernel-based regression [54], [55]. Gaussian Mixture Models (GMMs) generate smooth trajectories via Gaussian Mixture Regression (GMR) [56], while Inverse Reinforcement Learning (IRL) estimates reward functions from expert demonstrations to infer task objectives [57]. A potential drawback of many of these representations is that they require multiple demonstrations. Although DMPs can work with a single demonstration, they are primarily suited for adapting trajectories to rigid movements. In other fields of robotics, trajectory adaptation to non-rigid deformations has been successfully achieved using (a form of) Laplacian Trajectory Editing (LTE) [58], [59]. LTE is a framework that utilizes the Laplace-Beltrami operator, originally developed in the computer graphics community to deform triangular surface meshes [60]. LTE provides a smooth and flexible framework for modifying a single demonstration while preserving its local properties [61]. Its ability to require only a single demonstration and handle non-rigid deformations with low computational cost makes LTE a promising method for adapting ultrasound scan trajectories in real time to accommodate changes in the scanning environment.

This work aims to develop a RUSS (Fig. 2) capable of real-time adaptation to patient movement during MSK-US procedures while preserving the smoothness and characteristics of the original scan path. In addition, the proposed method allows for generalization of preoperatively planned trajectories and adaptation to patient movement in various ultrasound applications. More broadly, the framework could also be used in other areas where continuous adaptation to changing, deformable, or unpredictable surfaces is essential. To achieve this, the system leverages LTE, using an RGB-D camera to capture real-time surface point clouds of the scanned region. Point cloud registration estimates displacement between frames, defining anchor points for LTE-based trajectory adaptation. This ensures the probe maintains consistent surface contact while preserving smoothness and local trajectory properties. The initial trajectory can be demonstrated by an expert or generated from camera-based ROI extraction or user input [46], [49], [52]. The versatility

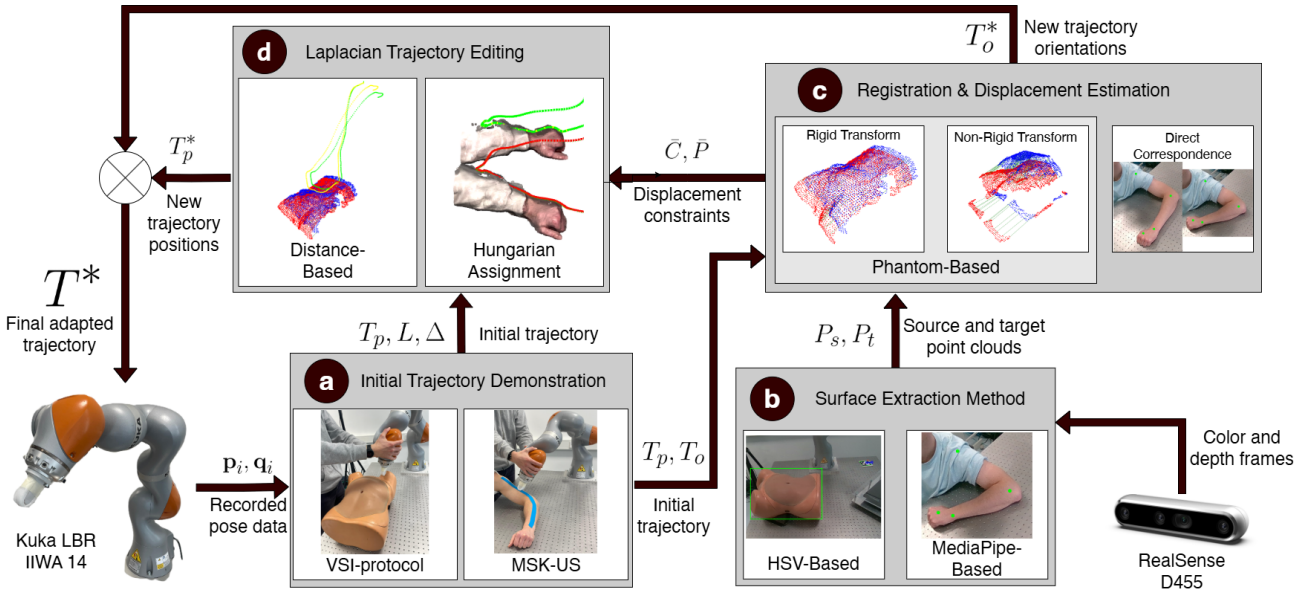


Fig. 2: Overview of the proposed Robotic Ultrasound System (RUSS). An initial kinesthetic demonstration of the scan trajectory is obtained (a), the scanned region's surface is extracted using an RGB-D camera (b), surface displacements are estimated via point cloud registration (c), and the trajectory is adapted to these displacements using Laplacian Trajectory Editing (LTE) (d). This framework enables real-time trajectory adaptation to accommodate rigid and non-rigid surface changes.

of the LTE technique to dynamically adapt preoperatively planned scan trajectories makes this technique widely applicable. With this work, we aim to achieve the following contributions:

1) The development of a point cloud registration pipeline for rigid and non-rigid transformations, including generalization between differently shaped lower abdomen phantoms and non-rigid human arm movements.
2) A method for dynamically defining new waypoints from registration results, enabling LTE-based trajectory transfer for smooth preoperative generalization and real-time adaptation to patient movement.

The proposed system (Fig. 2) is validated through three proof-of-concept experiments of increasing complexity, defined by the growing challenge of estimating surface displacement and adapting the trajectory. First, we test trajectory adaptation to rigid transformations on a lower abdomen phantom. Next, we evaluate generalization by transferring a trajectory between differently shaped abdominal phantoms. These two experiments demonstrate the system's broader applicability to ultrasound scanning. Finally, we apply the system to an MSK-US use case by demonstrating real-time adaptation to non-rigid arm movement, simulating the dynamic conditions encountered in clinical procedures.

II. METHODS

The proposed system (Fig. 2) consists of four main components: an initial demonstration of the ultrasound scan trajectory by a human user (Fig. 2a), the extraction of the surface of the scanned region (Fig. 2b), estimation of the displacement of the corresponding points on the surface due to movement or deformation (Fig. 2c), and the application of Laplacian Trajectory Editing (LTE) to adapt the trajectory to these changes (Fig. 2d). This section details each of these components and their implementation. II-A describes how initial scan trajectories are obtained through kinesthetic demonstrations following predefined ultrasound scanning protocols. Next, II-B explains how the surface of the scanned region is extracted and how all data is transformed into the robot's base frame. II-C then presents the surface registration algorithms used to estimate displacements between consecutive frames. Finally, II-D details the steps taken to transport the trajectory to the new surface using Laplacian Trajectory Editing.

A. Initial Trajectory Demonstration

To ensure clinical relevance, initial trajectories were based on existing ultrasound protocols rather than arbitrary scan paths. These trajectories were obtained through kinesthetic demonstrations, where an operator physically guided the ultrasound probe along a predefined scanning path. For lower abdomen phantom experiments, we used the Obstetric Volume Sweep Imaging (VSI) protocol [62], while an MSK-US-based scanning trajectory was used for non-rigid arm movement. These protocols reflected the nature of each experiment: VSI ensured a structured, repeatable scan for **phantom-based experiments**, whereas the MSK-US-based trajectory reflected the dynamic adjustments required for **non-rigid arm motion tracking**.

The VSI Protocol is a standardized ultrasound acquisition method designed for operators with minimal training, utilizing predefined sweeps over the abdomen to ensure consistent and repeatable image acquisition. In this study, we used the leftmost VSI sweep, which begins near the pelvic area and moves upward toward the upper abdomen. VSI has been applied in various clinical settings, including the automatic

detection of gestational age, fetal presentation, and twin pregnancies [62].

For the **non-rigid arm motion tracking scenarios**, the initial demonstration followed an MSK-US-based scanning trajectory. The kinesthetic demonstration consisted of a continuous scan along the entire arm, starting at the shoulder and moving toward the wrist, performed on a static, non-moving arm. This ensured a well-defined baseline trajectory that could later be adapted to simulate real MSK-US conditions, where sonographers must continuously adjust for patient movement [5].

During both the VSI and MSK-US kinesthetic demonstrations, pose data of the probe end-effector was recorded at 200 Hz to obtain the trajectory:

$$\mathcal{T} = \{T_i\}_{i=1}^{n}, \tag{1}$$

where each pose T_i consists of:

$$T_i = \begin{bmatrix} \mathbf{p}_i \\ \mathbf{q}_i \end{bmatrix} \tag{2}$$

with $\mathbf{p}_i = (x_i, y_i, z_i)^\top \in R^3$ representing the position of the US probe, and $\mathbf{q}_i = (q_{w,i}, q_{x,i}, q_{y,i}, q_{z,i})^\top \in H$ as the orientation given as a unit quaternion. For trajectory adaptation using LTE, we define the set of Cartesian waypoints as:

$$\mathcal{T}_p = \{\mathbf{p}_i\}_{i=1}^{n}, \quad \mathbf{p}_i \in R^3, \tag{3}$$

and the set of orientation waypoints as:

$$\mathcal{T}_o = \{\mathbf{q}_i\}_{i=1}^{n}, \quad \mathbf{q}_i \in H, \tag{4}$$

where $\mathcal{T}_p$ represents the positional data and $\mathcal{T}_o$ the orientational data of the trajectory.

B. Surface Extraction

To adapt the initial demonstration obtained through the LTE framework, an estimate of displacement for corresponding points on the surface was needed. As a first step, the (depth) images that captured changes in the scene were processed. This began with extracting the region of interest (ROI) from the camera's RGB images, which was achieved through either color-based segmentation or keypoint detection, depending on the experimental setting.

For **rigid and generalization phantom-based adaptation**, the ROI was extracted using HSV-based segmentation. A predefined range of HSV values corresponding to the phantom's specific color was used to generate a bounding box that isolated the region of interest. The 2D points within this bounding box were then projected into the depth image to obtain their 3D coordinates, forming a dense point cloud representation of the phantom surface. Afterward, a filtering step was applied to remove the table plane from the point cloud.

Non-rigid arm motion tracking, required a more robust approach, as color-based segmentation was unsuitable for the dynamic nature of the non-rigid movements. Instead, landmark detection was performed using Google's Mediapipe[1], a deep learning-based framework that extracts anatomical landmarks from images [63]. The four detected landmarks corresponding to the shoulder, elbow, wrist, and thumb were projected into the depth image to obtain their 3D coordinates. These four 3D points formed the point cloud representation of the arm, which served as the basis for estimating surface displacement and subsequently trajectory adaptation.

Regardless of the extraction method, all information was transformed into the robot's base frame, providing an intuitive global reference frame (Fig. 3). Determining the robot end-effector's pose in the robot base frame was necessary to track the trajectory, $\mathcal{T}$. This was accomplished by performing forward kinematics using the Eigen C++ library and the iiwa_ros URDF[2]. To find the transformation between the camera frame, $\{C\}$, and the robot's base frame, $\{B\}$, the OpenCV Python library was used to obtain the pose of the ArUco marker's center in the camera frame, resulting in the transformation matrix T_A^C. The transform between the ArUco marker, $\{A\}$, and robot base frame, $\{B\}$, was a pure known translation: $T_A^B = [0.0, -0.425, 0.017]^T$m. The total transform from C to B was given by:

$$T_C^B = T_A^B \cdot {T_A^C}^{-1} \tag{5}$$

This transformation ensured that all detected points were consistently represented in the robot's base frame for further processing.

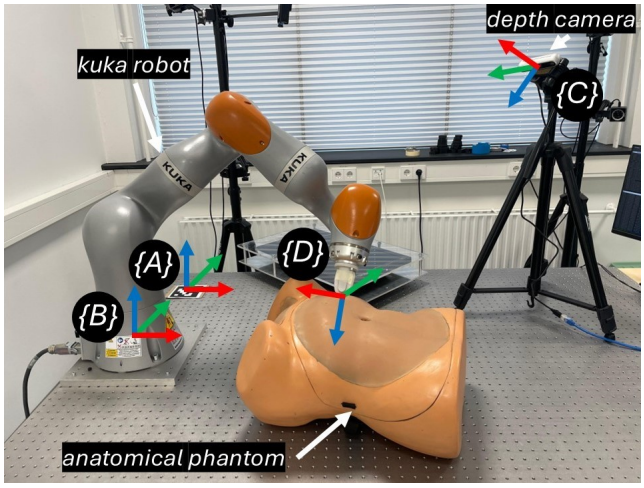


Fig. 3: Experimental setup including one lower abdomen phantom. $\{A\}$, $\{B\}$, $\{C\}$, and $\{D\}$ represent the ArUco marker, robot base, camera, and end-effector frames, respectively. The phantom's dense point cloud was obtained from RGB-D images and transformed into the robot base frame using T_A^B, T_A^C, and T_C^B. The robot end-effector's frame, D, was determined relative to the base frame using forward kinematics.

C. Displacement Estimation

To determine whether registration and trajectory adaptation were necessary, consecutive point cloud representations

[1] https://github.com/google-ai-edge/mediapipe

[2] https://github.com/epfl-lasa/iiwa_ros

of the surface were compared against a predefined threshold. Since surface changes varied between application scenarios, different measures were used. In phantom-based scenarios, registration was triggered when the Chamfer distance between consecutive point clouds exceeded 0.05 m. Non-rigid arm motion tracking relied on the sum of pairwise Euclidean distances between corresponding keypoints, skipping registration if this sum was below 0.05 m.

Once registration was initiated, the source and target point clouds were defined to estimate the displacement between them. For the registration of the surface point clouds, let us define the **source point cloud** as:

$$\mathcal{P}_s = \{\mathbf{p}_i^s\}_{i=1}^{M}, \quad \mathbf{p}_i^s = (x_i^s, y_i^s, z_i^s)^\top \in R^3 \tag{6}$$

where $\mathcal{P}_s$ consists of M 3D points $\mathbf{p}_i^s$, each represented by its spatial coordinates (x_i^s, y_i^s, z_i^s). Similarly, we define the **target point cloud** as:

$$\mathcal{P}_t = \{\mathbf{p}_j^t\}_{j=1}^{N}, \quad \mathbf{p}_j^t = (x_j^t, y_j^t, z_j^t)^\top \in R^3 \tag{7}$$

where $\mathcal{P}_t$ consists of N 3D points $\mathbf{p}_j^t$ with corresponding spatial coordinates (x_j^t, y_j^t, z_j^t). For the registration of $\mathcal{P}_s$ and $\mathcal{P}_t$, different approaches were applied depending on the experimental setting. In all cases, the goal was to establish correspondences and estimate the displacements between consecutive surface representations required for trajectory adaptation (Fig. 4).

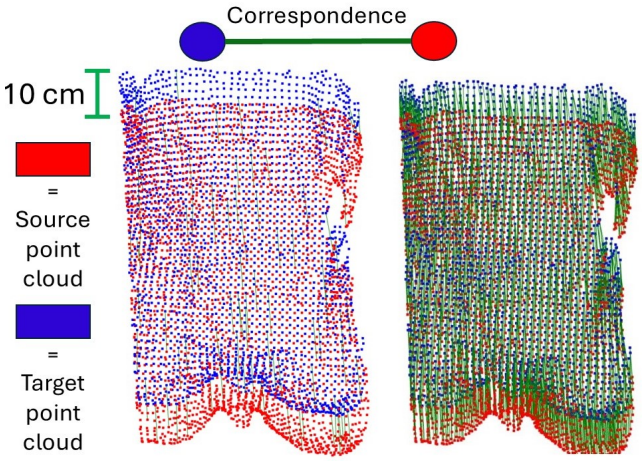


Fig. 4: Top-view of correspondences after registration of phantom source and target point clouds translated by 0.1m. Corresponding points are connected through a green line. Left: every 30th correspondence is shown for visibility. Right: all correspondences are shown.

For both **rigid and generalization phantom-based adaptation**, a four-step registration process was applied. First, statistical outliers were removed, and voxel-grid downsampling was performed to reduce computational complexity. Secondly, the centers of the point clouds were aligned. The key difference between the two cases lies in the initial correspondence estimation. For rigid transformations, Open3D's RANSAC registration method, based on feature matching, was performed. Fast Point Feature Histograms (FPFH) were used as features [64]. FPFH is a local geometric descriptor that encodes shape characteristics around each point, which helps to find correspondences. In contrast, for **generalization** requiring non-rigid transformations, Coherent Point Drift (CPD) was used [65]. By modeling one point cloud as a probability distribution, CPD iteratively aligns it to another, allowing deformations while preserving the overall structure. In both cases, Iterative Closest Point (ICP) was applied as a final step to find the correspondences [66]. For the **non-rigid arm motion tracking**, explicit point cloud registration was unnecessary, as the four tracked anatomical landmarks inherently defined surface correspondences. These landmarks provided a direct measure of surface displacement, eliminating the need for additional registration steps.

D. LTE Trajectory Transportation

To adapt the initial trajectory, $\mathcal{T}$, LTE was used to modify the positional waypoints, $\mathcal{T}_p$, based on the correspondences and the displacements found through the registration pipeline. For this work, the LTE implementation of [67] was used as a basis[3]. LTE solves an optimization problem that ensures modified waypoints preserve smoothness while respecting imposed positional constraints [61]. In LTE, we represent $\mathcal{T}_p$ as an undirected graph $\mathcal{G}$, with vertices $\mathcal{V}$ and edges $\mathcal{E}$ defining the connectivity between the vertices. Each point $\mathbf{p}_i$ in $\mathcal{T}_p$ corresponds to the vertex v_i in $\mathcal{V}$. The set of neighboring points of a vertex v_i, is given by

$$\mathcal{N}(v_i) = \{v_j \in \mathcal{V} \mid (v_i, v_j) \in \mathcal{E}\} \tag{8}$$

To capture the local properties of the trajectory, we convert the waypoints in Cartesian space $\mathbf{p}_i$ into Laplacian coordinates Δ. Δ is the concatenation of all δ_i which, for the vertex v_i, is given as:

$$\delta_i = \sum_{v_j \in \mathcal{N}(v_i)} w_{ij} \left(\frac{\mathbf{p}_i - \mathbf{p}_j}{\sum_{v_j \in \mathcal{N}(v_i)} w_{ij}} \right), \tag{9}$$

where w_{ij} is the edge weight between vertices v_i and v_j, set to 1 in our approach. To obtain Δ, the Laplacian matrix $L \in R^{n \times n}$ is used:

$$L_{ij} = \begin{cases} 1, & \text{if } i = j, \\ -\frac{w_{ij}}{\sum_{v_k \in \mathcal{N}(v_i)} w_{ik}}, & \text{if } v_j \in \mathcal{N}(v_i), \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

All Laplacian coordinates are then given by $\Delta = L\mathbf{T}_p$, where $\mathbf{T}_p$ represents the matrix form of the waypoint set $\mathcal{T}_p$. Once the Laplacian coordinates were computed, LTE found the adapted trajectory by solving a least-squares problem. The adapted trajectory, $\mathcal{T}_p^*$, was found by solving:

$$\mathbf{T}_p^* = \arg\min_{\mathbf{T}_p^*} \|L\mathbf{T}_p^* - \Delta\|^2 + \lambda \|\bar{P}\mathbf{T}_p^* - \bar{\mathbf{C}}\|^2 \tag{11}$$

[3] https://github.com/franzesegiovanni/gaussian_process_transportation

where $\bar{P}$ is a diagonal binary matrix where each entry $\bar{P}_{kk}$ is set to 1 if waypoint $\mathbf{p}_k^*$ in $\mathcal{T}_p^*$ is constrained, and 0 otherwise. $\bar{C}$ contains the target positions for the constrained waypoints in $\mathcal{T}_p^*$. λ is a regularization weight, controlling the trade-off between preserving local shape properties and adhering to the new displacement constraints.

In this work, for all use cases, the displacement constraints, $\bar{C}$, in LTE were determined based on the displacement between corresponding points in the source point cloud, $\mathcal{P}_s$, and the target point cloud, $\mathcal{P}_t$. The selection of which waypoints from the original trajectory, $\mathbf{p}_k \in \mathcal{T}_p$, influence the adaptation process is given by:

$$\mathcal{K} = \{k \mid \mathbf{p}_k \in \mathcal{T}_p\}. \tag{12}$$

For **rigid and generalization phantom-based adaptation**, where point clouds were dense, a nearest-neighbor search using a KDTree was used to identify waypoints in $\mathcal{T}_p$ that were within a predefined distance threshold of 0.03 m from $\mathcal{P}_s$, indicating physical contact with the surface. Not all contact points were used as constraints, as this would overly restrict the adaptation and leave little room to preserve local properties during LTE optimization. Instead, every z^{th} contact point was selected, where z was a tunable parameter. A smaller z enforced stricter positional tracking by adding more constraints, while a larger z provided greater flexibility in preserving the original trajectory's shape. For **non-rigid arm motion tracking**, where the representation consisted of sparse keypoints, the Hungarian algorithm was applied to optimally assign four waypoints from $\mathcal{T}_p$ to the four keypoints in $\mathcal{P}_s$ [68]. The adaptation was then constrained such that the corresponding waypoints in $\mathcal{T}_p^*$ aligned with these key anatomical locations.

Each selected waypoint $\mathbf{p}_k \in \mathcal{T}_p$ defined a corresponding constraint in the adapted trajectory $\mathcal{T}_p^*$. These constraints were indexed in the set $\mathcal{K}$ and for each $k \in \mathcal{K}$, the corresponding entry in $\bar{P}$ was set to 1:

$$\bar{P}_{kk} = 1, \quad \forall k \in \mathcal{K}. \tag{13}$$

The displacement for each constraint was computed based on the difference between corresponding points in $\mathcal{P}_s$ and $\mathcal{P}_t$:

$$\mathbf{d}_k = \mathbf{p}_m^t - \mathbf{p}_\ell^s, \quad \forall k \in \mathcal{K}, \tag{14}$$

where $\mathbf{p}_\ell^s \in \mathcal{P}_s$ and $\mathbf{p}_m^t \in \mathcal{P}_t$ denote corresponding points in the source and target point clouds, respectively. The selected constraints were then applied to the adapted trajectory $\mathcal{T}_p^*$, modifying the corresponding waypoints as

$$\bar{\mathbf{C}}_k = \mathbf{p}_k + \mathbf{d}_k, \quad \forall k \in \mathcal{K}. \tag{15}$$

Additionally, to maintain trajectory continuity and ensure smooth motion transitions, the robot's current position was added as a constraint. This ensured that the current robot position was always a part of $\mathcal{T}_p^*$. Fig. 5 shows a transportation following from the registration procedure from Fig. 4.

As previously described, orienting the ultrasound probe perpendicular to the scanning surface allows for optimal propagation of ultrasound waves [12]. To compute the orientational component of the adapted trajectory $\mathcal{T}^*$, surface normals of the target point cloud, $\mathcal{P}_t$, were first estimated using the Open3D Python library. For each positional waypoint in the adapted trajectory, $\mathcal{T}_p^*$, the closest corresponding point in $\mathcal{P}_t$ within a distance threshold of 0.03 m was identified, using the same KDTree method as in LTE. The surface normal of all these points in $\mathcal{P}_t$ defined the probe's orientation. Specifically, the orientational component $\mathbf{q}_i$ of each waypoint was aligned with the surface normal of the closest point but oriented towards the surface. For all points not within the distance threshold, the orientation was kept equal to the original orientation in $\mathcal{T}$. Combining the positional waypoints obtained through LTE and the surface normal-based orientations, resulted in the adapted trajectory $\mathcal{T}^*$.

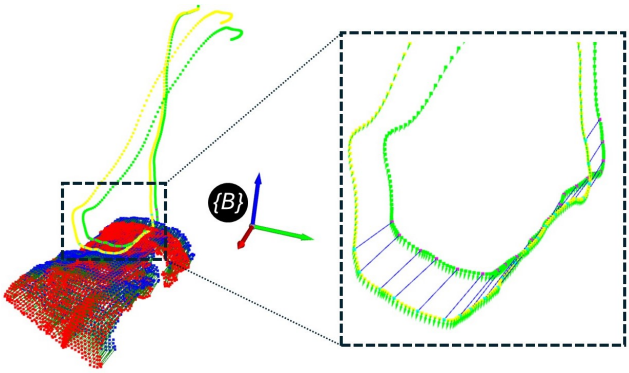


Fig. 5: Laplacian Trajectory Editing (LTE) adaptation based on point cloud registration. The original (green) and adapted (yellow) trajectories, derived from the registration between the source and target point clouds (Fig. 4), are shown with axes indicating the robot's base frame $\{B\}$. The callout shows a visualization of $\bar{P}$ and $\bar{C}$. Magenta points indicate selected waypoints ($\bar{P}$), while blue lines represent the displacement constraints ($\bar{C}$).

III. EXPERIMENTAL VALIDATION

A. Experimental Design and Procedures

1) Experimental Setup: To validate the proposed system, a series of experiments were conducted using a robotic manipulator, a depth camera, and a silicone phantom or a human arm (Fig. 3). A Kuka LBR iiwa 14 R820 (KUKA AG, Augsburg, Germany) [69] robotic manipulator was equipped with a 3D-printed mock ultrasound probe and controlled via the KUKA Fast Robot Interface (FRI). An Intel RealSense D455 (Intel Corporation, Santa Clara, CA, USA) depth camera was mounted above the workspace at an incline to capture real-time RGB-D data. Computations were distributed across two machines. The first was a laptop equipped with an Intel Core i7-8750H processor, 16GB of RAM, a NVIDIA Quadro P1000 GPU and operating under Ubuntu 20.04 with ROS Noetic. This machine processed camera data in real time and executed the Laplacian Trajectory Editing (LTE) pipeline. The second machine was a dedicated desktop

responsible for running the robotic manipulator's controller. It featured a 12th Gen Intel Core i7-12700K processor, 32GB of RAM, integrated Intel UHD Graphics 770 and operated under Ubuntu 20.04.6 LTS with ROS Noetic.

2) Robot Control: To demonstrate real-time adaptability in ultrasound scanning, the robotic manipulator maintained contact with the scan surface while following the adapted trajectory. A certain contact force is needed to allow the propagation of ultrasound waves through human tissue. However, safety must be guaranteed in any clinical setting. That is why a compliant Cartesian impedance controller was implemented to ensure safe interaction. The controller is defined as follows:

$$\boldsymbol{\tau} = \mathbf{J}^{\top}(\mathbf{K}(\mathbf{p}_{\text{ref}} - \mathbf{p}) + \mathbf{D}(\mathbf{J}\dot{\mathbf{q}}) + \mathbf{F}_{\text{ext}}) \tag{16}$$

where $\boldsymbol{\tau} \in R^{7\times 1}$ represents the joint torques sent to the robot. The matrix $\mathbf{J} \in R^{6\times 7}$ is the robot Jacobian matrix. The term $\mathbf{K} \in R^{6\times 6}$ is the Cartesian stiffness matrix, regulating the compliance of the manipulator. The vectors $\mathbf{p}_{\text{ref}}, \mathbf{p} \in R^3$ represent the reference and current Cartesian positions of the tip of the ultrasound probe end-effector, respectively. The damping matrix $\mathbf{D} \in R^{6\times 6}$ introduces velocity-dependent damping forces to improve stability, where $\dot{\mathbf{q}} \in R^7$ is the joint velocity vector. The resulting control law ensured that the end-effector followed the desired Cartesian trajectory while maintaining compliance. The final term, the external force, is given by $\mathbf{F}_{\text{ext}} \in R^6$. Translational stiffness was set to 300.0 N/m with a damping ratio of 0.7, yielding damping values in N·s/m. Similarly, rotational stiffness was 10.0 Nm/rad with the same damping ratio, resulting in damping values in Nm·s/rad. The external force was set equal to 2.0 N balancing proper surface contact and patient safety [16], [17]. Furthermore, this force was only active when the controller was within a 0.1 m bound of the surface's z-height, where the height was obtained from the depth images. This method ensured the force control term was only enabled when making contact.

3) Experimental Scenarios: The validation consisted of three experiments increasing in complexity, with a static demonstration replay without a scanning surface to serve as a baseline for trajectory execution. In the first experiment, a single phantom was translated, rotated, and tilted to assess the system's ability to adapt to rigid movements. While a direct transformation found through the point cloud registration could suffice for this case, LTE was applied as an initial proof-of-concept and to serve as a control before testing more complex adaptations. The second experiment adapted a demonstration from one abdominal phantom to another with a different shape (male and one female) to evaluate generalization to anatomical variations. Finally, in the third experiment, non-rigid arm movement was introduced to simulate dynamic patient movement and assess the system's adaptation capabilities, reflecting the probe adjustments required in MSK-US procedures. Each scenario included five trials, yielding five replay trials, 15 rigid adaptation trials (five each for translation, rotation, and tilt), five generalization trials, and five non-rigid adaptation trials.

4) Evaluation Metrics: Performance was evaluated both quantitatively and qualitatively, with video material available[4]. This work aimed to develop a RUSS capable of adapting to patient movements in real-time during ultrasound procedures while preserving smoothness. To this end, the baseline performance of replaying the initial demonstration was first evaluated in III-B.1. Subsequently, the system's ability to handle dynamic patient motion was evaluated using success rates and trajectory adaptation accuracy in III-B.2. This evaluation was conducted specifically for the non-rigid case, as it introduced the most complex and deformed transformations. Since non-rigid adaptation was the most challenging, its accuracy was a strong performance indicator in less complex scenarios. To ensure that adaptation occurred while preserving smoothness, positional tracking error and jerk were analyzed, reflecting the deviation between the robot's actual and reference trajectory and the smoothness of motion, respectively, detailed in III-B.3. The real-time feasibility of adaptations was evaluated through the computation times, examining image processing, point cloud registration, and trajectory adaptation speed in III-B.4. Additionally, registration accuracy, measured by the mean point-to-surface distance and overlap ratio, was analyzed in III-B.5 to assess the quality of point cloud alignment during surface displacements.

B. Results

1) Baseline Performance: When replaying a recorded demonstration from the VSI protocol five times (Fig. 6), the trajectory tracking mean and maximum errors were 0.018 ± 0.001 m and 0.079 ± 0.002 m, respectively. The total mean jerk recorded in the motion was 4.03 ± 0.1 m/s^3, and the maximum jerk was 19.8 ± 1.4 m/s^3.

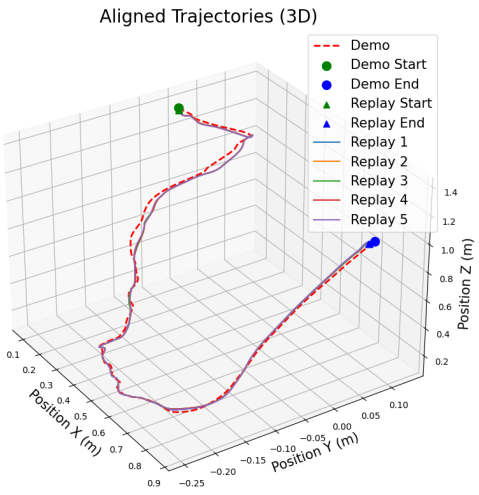


Fig. 6: The red-dotted line represents a VSI protocol demonstration, while the colored lines show recorded poses during replay. Due to close alignment, individual trajectories are difficult to distinguish.

[4]https://youtu.be/XwbCbb_WA7Q

2) Adaptation Success and Accuracy: For all experimental conditions, the success and failure outcomes are presented (Table I). A failure case here was defined as the robotic manipulator not being able to complete the motion and disconnecting. The system successfully adapted in most trials, with occasional failures occurring in the rigid and non-rigid adaptation scenarios.

TABLE I: Success/Failure Results Across All Experiments

Trial #	Rigid Movements			Generalization	Non-Rigid
	T 20 cm	R 45°	Tilt 30°		
1	S	S	F	S	S
2	S	F	S	S	S
3	S	F	S	S	S
4	S	S	F	S	F
5	S	S	S	S	S

Next to adaptation success, we report its accuracy. Adaptation accuracy was quantified in the non-rigid arm motion tracking scenario by comparing the adapted trajectory to a predefined reference line drawn with blue physiotherapy tape on the human arm (Fig. 7). This reference, a blue line visible in the camera view, represented the ideal scanning path that the adapted trajectory should follow after accounting for surface movement. The initial MSK-US demonstrated trajectory was obtained by asking an operator to manually follow this blue line on a static arm, ensuring that the intended scan path aligned with the reference. The position of the blue line was extracted in the robot's base frame, enabling a direct comparison with the transported trajectory. Accuracy was measured as the deviation between corresponding points on the adapted trajectory and the reference line.

To evaluate adaptation accuracy, we report the root mean squared error (RMSE), which quantifies the average deviation between corresponding points on the adapted trajectory and the reference line. The initial error between the operator's demonstration and the blue line was averaged over 30 frames, resulting in a mean deviation of 0.021 ± 0.002 m. Over 54 LTE adaptations, the RMSE between the blue line and the adapted trajectory was computed (Fig. 8). The mean RMSE across all adaptations was 0.026 ± 0.012 m.

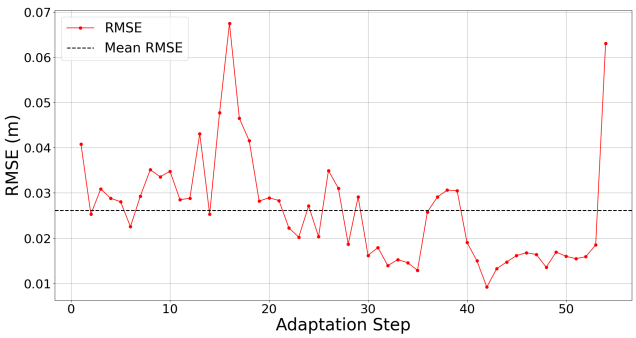


Fig. 8: RMSE between the adapted trajectory and the moving blue line reference over 54 adaptation steps. The mean RMSE across all adaptations is 0.026 ± 0.012 m.

3) Trajectory Tracking and Smoothness Analysis: We evaluated tracking performance using the positional tracking error, measuring deviations between the executed trajectory and the adapted reference trajectory. Furthermore, jerk analysis quantified motion smoothness, where lower jerk values indicated smoother movements.

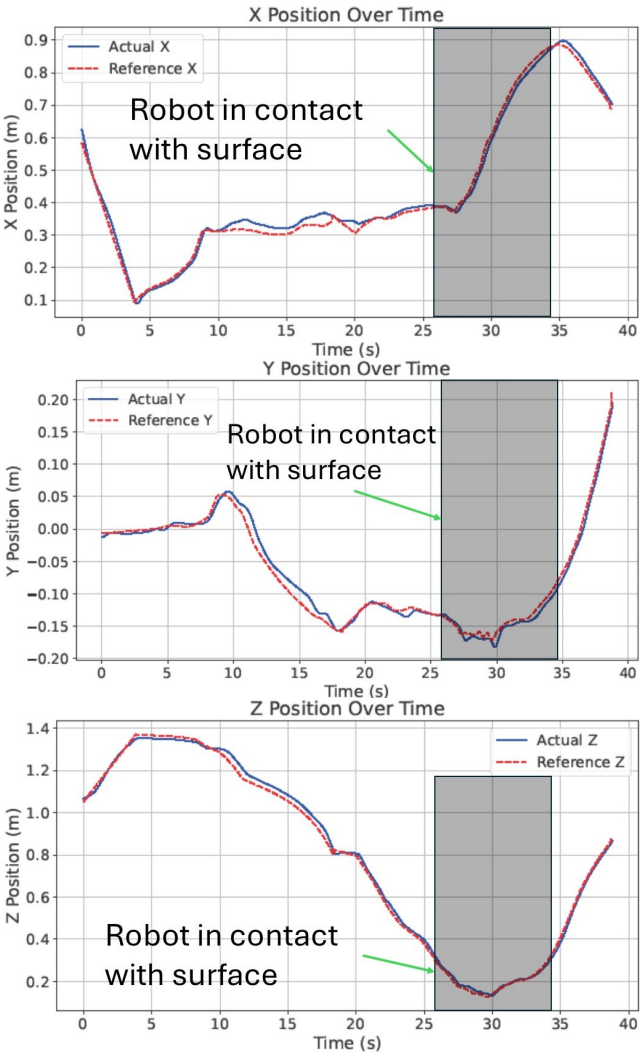


Fig. 9: Tracking errors across the X, Y, and Z dimensions. Each plot shows the deviation between the executed trajectory and the adapted reference trajectory over time. These results correspond to a 'Translation 20 cm' trial.

For each individual trial, the tracking error at every timestep was recorded (Fig. 9). Across scenarios, both mean and maximum positional tracking errors remained low compared to the static demonstration replay (Fig. 10). However, errors increased in the translational rigid scenario, generalization, and non-rigid adaptation, with non-rigid adaptations showing the highest spread. Notably, the maximum positional tracking errors were highest in the non-rigid case.

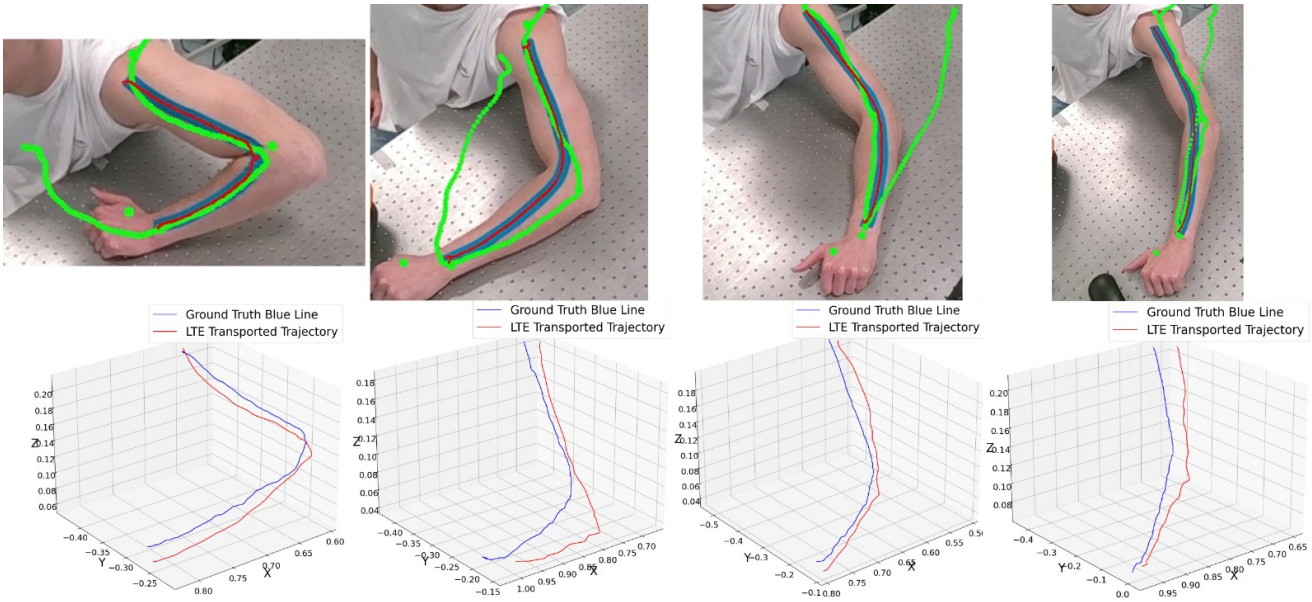


Fig. 7: Illustration of trajectory adaptation accuracy evaluation. The blue line represents the ground truth reference drawn on the arm, while the red line corresponds to the adapted trajectory transported using Laplacian Trajectory Editing (LTE). The deviation between the two lines quantifies adaptation accuracy.

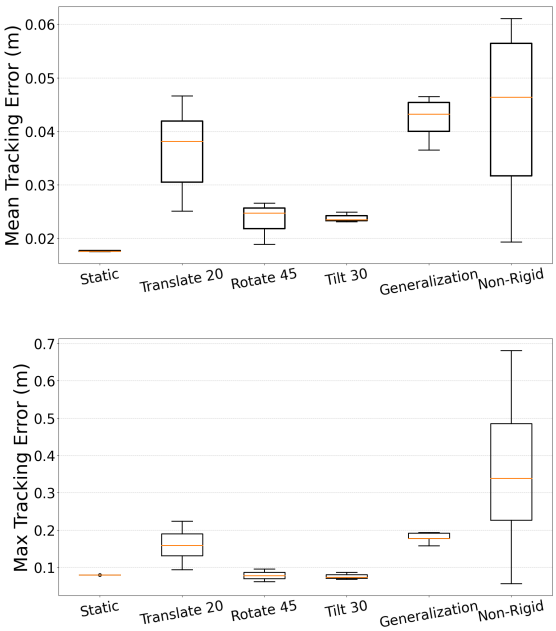


Fig. 10: Mean and maximum positional tracking errors across experimental conditions, with five trials per scenario. Errors were low in static replay but increased in translational rigid, generalization, and non-rigid adaptations. Non-rigid cases exhibited the highest spread and maximum errors.

Mean jerk remained low and consistent in the static, generalization, and tilt scenarios (Fig. 11). Higher variability was observed in translational and rotational rigid adaptations, with occasional spikes. Non-rigid adaptations showed the highest mean jerk values, with some trials exceeding 20 m/s³. Maximum jerk varied across conditions, with the highest values appearing in translational rigid and non-rigid adaptations, exceeding 500 m/s³ in some cases.

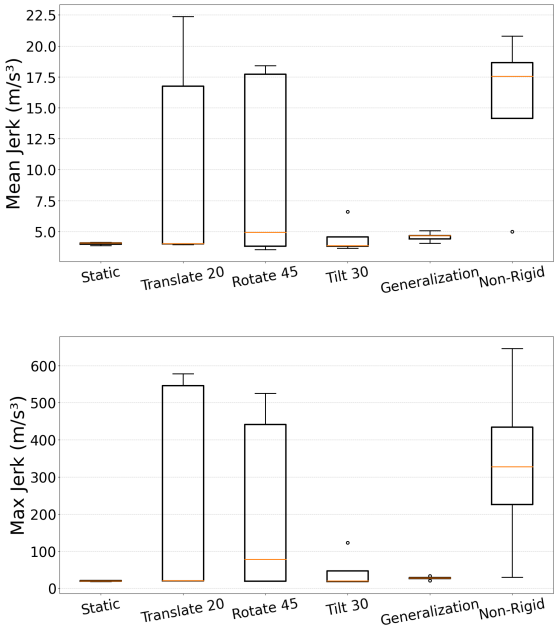


Fig. 11: Mean and maximum jerk values across experimental conditions, with five trials per scenario. Non-rigid adaptations exhibited the highest mean jerk, often exceeding 20 m/s³. Translational and rotational rigid and non-rigid cases showed the highest maximum jerk.

4) Computational Performance: To assess computational feasibility, we measured the execution times for three key processing steps: image processing, displacement estimation, and LTE-based trajectory adaptation. These steps correspond directly to components (b), (c), and (d) of the proposed framework (Fig. 2). Each step consists of several sub-components, and the reported times represent the aggregated, averaged computation time across these components (Table II).

Image processing times remained similar in the rigid and generalization scenarios, at 942 ms and 1090 ms, respectively, while the non-rigid scenario had a lower processing time of 55.2 ms. The displacement estimation time showed significant variation across scenarios, with the generalization case taking considerably longer than rigid and non-rigid adaptations. Lastly, LTE-based trajectory adaptation required 240 ms in the rigid scenario, 463 ms in the generalization scenario, and 415 ms in the non-rigid scenario.

TABLE II: Computation Times (ms) Across Scenarios

Component	**Rigid (ms)**	**Generalization (ms)**	**Non-Rigid (ms)**
Image Processing	942	1,090	55.2
Displacement Estimation	1,090	40,500	0.380
LTE Trajectory Adaptation	240	463	415

5) Point Cloud Registration Accuracy: Since most scenarios relied on point cloud registration for estimating surface displacements, registration accuracy directly influenced system performance. To assess registration quality, we measured the mean point-to-surface distance and overlap ratio between the registered point clouds (Table III). Only scenarios where registration took place and successful trials were included. Across all tested scenarios, the mean point-to-surface distance ranged from 10.0 ± 0.5 mm to 14.7 ± 0.3 mm. Maximum distances varied more significantly, with the highest deviations occurring in the Tilt 30° scenario (97.3 ± 27.2 mm). Overlap ratios remained consistently high, exceeding 90% across all conditions, with the generalization scenario achieving full overlap (100%).

TABLE III: Point Cloud Registration Accuracy Metrics Across Scenarios

Metric	**T20 cm**	**R 45°**	**Tilt 30°**	**Generalization**
Point-to-Surface Mean Distance (mm ± SD)	10.0 ± 0.5	12.9 ± 0.3	14.7 ± 0.3	11.2 ± 0.1
Point-to-Surface Max Distance (mm ± SD)	51.0 ± 8.1	52.7 ± 3.2	97.3 ± 27.2	69.0 ± 9.8
Overlap Ratio (% ± SD)	98.1 ± 2.5	90.7 ± 2.9	97.6 ± 0.1	100.0 ± 0.0

IV. DISCUSSION

The following discussion evaluates the performance of the proposed framework in IV-A, discusses its limitations in IV-B, and concludes with recommendations for future work in IV-C.

A. Static, Rigid and Non-rigid Performance Comparison

Most trials were successful in that the trajectory could be adapted and followed by the robot, but in some cases, jerky motion led to velocity or torque limit violations, triggering the robot's safety stop and causing a disconnect. This was not due to adaptation, but rather to kinematic singularities, which LTE does not account for. Since LTE modifies the trajectory only in Cartesian space, it does not consider workspace constraints, joint limits, or singularity avoidance. As a result, when the adapted trajectory approached a singularity, the robot's joint velocities increased unpredictably, leading to unstable motion. These jerky motions were localized to small trajectory sections but significantly increased the overall mean and maximum jerk values (Fig. 11). However, when excluding these short high-jerk segments, the mean jerk closely aligns with the baseline value of 4.03 ± 0.1 m/s^3, suggesting that most of the trajectory remained smooth across trials.

For successful non-rigid trials, the error between the blue reference line and the adapted trajectory remained stable, within 0.035 and 0.015 m for most adaptations (Fig. 8). The mean RMSE after adaptation (0.026 ± 0.012 m) was slightly higher than the initial deviation in the operator-drawn line (0.021 ± 0.002 m), indicating effective adaptations with minor deviations. However, some trials showed errors exceeding 0.06 m, while others had lower RMSE than the initial demonstration, highlighting variability in adaptation accuracy influenced by arm movement (Fig. 8). This variability can be attributed to LTE optimization's properties: when the arm is fully extended, the constrained keypoints align in a straight line, causing the local Laplacian coordinates to become zero. This configuration minimizes the LTE optimization objective, making a straight trajectory the optimal solution. However, when the arm bends near 45°, the new keypoint configuration deviates from a straight path. LTE, which minimizes changes in local Laplacian coordinates rather than enforcing global linearity, distributes adjustments across all the unconstrained points. As a result, the trajectory curves inward from the shoulder to the elbow and outward from the elbow to the wrist instead of forming distinct straight-line segments (Fig. 7, leftmost plot). This effect is amplified by the sparse use of MediaPipe keypoint constraints relative to the trajectory length.

Once the trajectory was adapted, the system demonstrated consistent and low tracking errors, with a mean and maximum tracking error of 0.018 ± 0.001 m and 0.079 ± 0.002 m during static replay (Fig. 6). Across adaptation scenarios, mean tracking errors remained low but were slightly higher in trials where the probe made contact with the surface, likely due to force interactions (Fig. 10). In the non-rigid case, mean tracking errors were further influenced by high maximum errors, which resulted from parts of the reference trajectory extending beyond the robot's reachable workspace. At these points, the robot fully stretched in the target direction without being able to reach the pose, inflating both maximum and mean tracking errors. Even with

these inflations, mean tracking errors remained below 0.05 m, indicating reliable tracking. However, further analysis is needed to determine whether these errors indeed stem from the trajectory extending beyond the robot's workspace or result from, for example, differences between human tissue and rigid phantoms.

Computation times varied significantly across scenarios. Image processing was nearly 20 times faster in the non-rigid case, completing in 55.2 ms, as MediaPipe landmark detection required less computation than the HSV-based segmentation used for phantoms. Displacement estimation in the generalization scenario was substantially slower due to CPD registration, taking 40.5 seconds compared to 1090 ms in the rigid case and just 0.380 ms in the non-rigid case. While this makes it unsuitable for real-time applications with dense point clouds, it remains viable for generalizing preoperatively planned scan paths. In contrast, the keypoint-based displacement estimation used in the non-rigid scenario proved highly efficient. Regardless of the scenario, LTE adaptation was consistently fast, executing within 500 ms once displacements were obtained, ensuring real-time trajectory adaptation.

Point cloud registration was effective overall, confirming the suitability of the selected methods for estimating surface displacements. Across all tested scenarios, registration achieved high accuracy, with mean point-to-surface distances ranging from 10.0 ± 0.5 mm to 14.7 ± 0.3 mm and overlap ratios exceeding 90%. Notably, the generalization scenario, which employed CPD for registration, consistently achieved full overlap (100%). This could be attributed to the probabilistic nature of CPD, which inherently distributes correspondences throughout the set of points, leading to complete overlap. However, overlap alone may not be the best metric to assess the success of CPD, as it does not account for possible local misalignments. Since the primary focus of this work was on trajectory adaptation rather than registration accuracy, further refinements in registration were not extensively explored.

B. System Limitations

This proof-of-concept RUSS and the proposed framework demonstrate the potential of integrating LTE for ultrasound scan trajectory adaptation, but several limitations remain.

A key limitation is the need for manually defined constraints in LTE. While this also brings flexibility to account for non-rigid deformations, inaccuracies in surface displacement estimation directly propagate into the adapted trajectory. Even though reported registration accuracies in III-B.5 were high, this does not capture the entire story of displacement estimation. In particular, large differences in perspective between frames posed challenges, leading to errors that affected adaptation quality. More advanced techniques such as deep-learning-based registration could improve accuracy and possibly reduce computational times by replacing CPD for the non-rigid dense point cloud registration.

Additionally, selecting constraints that balance local flexibility with global trajectory preservation is nontrivial. LTE preserves local Laplacian relationships but does not explicitly enforce trajectory shape, meaning adaptations may not retain, for example, the straight-line nature of the original demonstration in this work's non-rigid experiments. Defining constraints based on surface geometry, such as normals, curvature, or anatomical landmarks, could improve consistency. Additionally, adaptive constraint placement based on trajectory deviation may better balance local flexibility with global trajectory preservation.

Next to constraint selection, the accuracy of adaptation begins with robust surface extraction. The HSV-based segmentation for phantoms and keypoint tracking for the arm are sensitive to lighting, occlusions, and anatomical variability. Errors in segmentation or keypoint localization quickly degrade adaptation quality. Future work could benefit from leveraging multi-view point cloud data.

Another limitation is the primary focus on positional adaptations rather than force and orientation control. While these were implemented, they were not extensively optimized. Both factors are critical for ultrasound image quality, and integrating force- and orientation-aware control strategies could enhance probe stability and imaging consistency [13], [15], [20], [59], [70].

Furthermore, LTE operates purely in Cartesian space, disregarding kinematic constraints such as joint limits and singularity avoidance. This occasionally led to the earlier reported unstable robot motion causing system disconnections. Incorporating workspace-aware constraints could mitigate these issues.

Finally, even with perfectly segmented and registered point clouds and reliable force and orientation control, evaluating the quality of the adapted trajectories remains challenging due to the absence of an actual ultrasound machine. Without imaging feedback, it is difficult to assess whether the system achieves clinically relevant outcomes. Integrating a functioning ultrasound device into the setup would allow for validation of the quality of the trajectory based on the ultrasound images obtained. For example, by computing the ultrasound confidence map [71], [72].

C. Future Work

Future efforts could focus on integrating a functional ultrasound device to assess clinical relevance, improving displacement estimation through surface registration robustness, and refining force and orientation control strategies to enhance probe stability. Additionally, optimizing constraint selection and adaptive constraint placement could improve LTE's balance between local flexibility and global trajectory preservation. Addressing kinematic feasibility, singularity avoidance, and real-time processing will further enhance system performance and practical deployment.

Beyond ultrasound applications, combining LTE with a real-time registration pipeline may extend to other robotic domains. Many tasks require continuous adaptation to changing, deformable, or unpredictable surfaces, where trajectory

consistency and smoothness are essential. The ability to adjust motion dynamically without repeated demonstrations makes this approach applicable to various fields, such as robotic surface cleaning, agricultural automation, and space exploration, where robots must interact with varying terrains, soft tissues, or biological surfaces. Future work could explore LTE's adaptability to these domains, particularly in real-time, closed-loop control scenarios where deformation-aware trajectory updates are required.

V. CONCLUSION

This work introduced a novel Robotic Ultrasound System (RUSS) that adapts scan trajectories in real time to both rigid and non-rigid patient movements using Laplacian Trajectory Editing (LTE). In addition to adapting to real-time movement, the system demonstrated the ability to generalize preoperatively planned trajectories across different anatomical shapes. While LTE has been applied in other areas of robotics, this study is the first to integrate it into robotic ultrasound, enabling trajectory adaptation without requiring multiple demonstrations. The system successfully adapted trajectories with a mean adaptation accuracy of 0.026 ± 0.012 m in non-rigid cases and completed LTE-based adaptations within 373 ms, ensuring real-time feasibility. Tracking performance remained stable, with mean errors below 0.05 m, even in dynamic scenarios.

Beyond ultrasound, the proposed framework provides a generalizable approach for real-time trajectory adjustments, with applications in medical imaging, surface interaction tasks, and autonomous systems requiring continuous adaptation. By enabling real-time adaptation to patient movement, this work represents a step toward more autonomous and intelligent robotic ultrasound systems, reducing operator dependency and paving the way for more consistent and accessible ultrasound imaging in clinical practice.

REFERENCES

[1] K. K. Shung, "Ultrasound: Past, Present and Future," in *The Third International Conference on the Development of Biomedical Engineering in Vietnam*, V. Van Toi and T. Q. D. Khoa, Eds. Berlin, Heidelberg: Springer, 2010, pp. 10–13.

[2] E. Bercovich and M. C. Javitt, "Medical Imaging: From Roentgen to the Digital Revolution, and Beyond," *Rambam Maimonides Medical Journal*, vol. 9, no. 4, p. e0034, Oct. 2018. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6186003/

[3] R. Kojcev, A. Khakzar, B. Fuerst, O. Zettinig, C. Fahkry, R. DeJong, J. Richmon, R. Taylor, E. Sinibaldi, and N. Navab, "On the reproducibility of expert-operated and robotic ultrasound acquisitions," *International Journal of Computer Assisted Radiology and Surgery*, vol. 12, no. 6, pp. 1003–1011, Jun. 2017. [Online]. Available: https://doi.org/10.1007/s11548-017-1561-1

[4] S. Bianchi and C. Martinoli, *Ultrasound of the musculoskeletal system*, ser. Medical radiology. Berlin: Springer-Verlag, 2007.

[5] C. Martinoli, "Musculoskeletal ultrasound: Technical guidelines," *Insights into imaging*, vol. 1, pp. 99–141, Jul. 2010.

[6] D. Won, J. Walker, R. Horowitz, S. Bharadwaj, E. Carlton, and H. Gabriel, "Sound the Alarm: The Sonographer Shortage is Echoing Across Healthcare," *Journal of Ultrasound in Medicine*, vol. n/a, no. n/a, Jan. 2024, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jum.16453. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/jum.16453

[7] J. T. Finnoff, "The Evolution of Diagnostic and Interventional Ultrasound in Sports Medicine," *PM&R*, vol. 8, no. 3S, pp. S133–S138, 2016, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1016/j.pmrj.2015.09.022. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1016/j.pmrj.2015.09.022

[8] O. J. Pallotta and A. Roberts, "Musculoskeletal pain and injury in sonographers, causes and solutions," *Sonography*, vol. 4, no. 1, pp. 5–12, 2017, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/sono.12093. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sono.12093

[9] K. Evans, S. Roll, and J. Baker, "Work-Related Musculoskeletal Disorders (WRMSD) Among Registered Diagnostic Medical Sonographers and Vascular Technologists: A Representative Sample," *Journal of Diagnostic Medical Sonography*, vol. 25, no. 6, pp. 287–299, Nov. 2009, publisher: SAGE Publications Inc STM. [Online]. Available: https://doi.org/10.1177/8756479309351748

[10] S. E. Salcudean, Z. Jiang, and N. Navab, "Robotic ultrasound imaging: State-of-the-art and future perspectives," *Medical Image Analysis*, vol. 89, p. 102878, Oct. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S136184152300138X

[11] A. Kaminski, A. Payne, S. Roemer, D. Ignatowski, and B. K. Khandheria, "Answering to the call of critically ill patients: Limiting sonographer exposure to COVID-19 with focused protocols," *Journal of the American Society of Echocardiography : official publication of the American Society of Echocardiography*, 2020. [Online]. Available: https://works.bepress.com/bijoy-khandheria/98/

[12] B. Ihnatsenka and A. P. Boezaart, "Ultrasound: Basic understanding and learning the language," *International Journal of Shoulder Surgery*, vol. 4, no. 3, pp. 55–62, Jul. 2010.

[13] Z. Jiang, M. Grimm, M. Zhou, J. Esteban, W. Simson, G. Zahnd, and N. Navab, "Automatic Normal Positioning of Robotic Ultrasound Probe Based Only on Confidence Map Optimization and Force Measurement," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1342–1349, Apr. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8963620/

[14] Z. Jiang, M. Grimm, M. Zhou, Y. Hu, J. Esteban, and N. Navab, "Automatic Force-Based Probe Positioning for Precise Robotic Ultrasound Acquisition," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 11, pp. 11 200–11 211, Nov. 2020, conference Name: IEEE Transactions on Industrial Electronics. [Online]. Available: https://ieeexplore.ieee.org/document/9257181

[15] D. Raina, A. Mathur, R. M. Voyles, J. Wachs, S. Chandrashekhara, and S. K. Saha, "RUSOpt: Robotic UltraSound Probe Normalization with Bayesian Optimization for In-Plane and Out-Plane Scanning," in *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*, Aug. 2023, pp. 1–7, iSSN: 2161-8089. [Online]. Available: https://ieeexplore.ieee.org/document/10260479

[16] M. Dhyani, S. C. Roll, M. W. Gilbertson, M. Orlowski, A. Anvari, Q. Li, B. Anthony, and A. E. Samir, "A pilot study to precisely quantify forces applied by sonographers while scanning: A step toward reducing ergonomic injury," *Work (Reading, Mass.)*, vol. 58, no. 2, pp. 241–247, 2017.

[17] R. Tsumura, J. W. Hardin, K. Bimbraw, A. V. Grossestreuer, O. S. Odusanya, Y. Zheng, J. C. Hill, B. Hoffmann, W. Soboyejo, and H. K. Zhang, "Tele-Operative Low-Cost Robotic Lung Ultrasound Scanning Platform for Triage of COVID-19 Patients," *IEEE robotics and automation letters*, vol. 6, no. 3, pp. 4664–4671, Jul. 2021.

[18] M. Dyck, A. Sachtler, J. Klodmann, and A. Albu-Schaffer, "Impedance Control on Arbitrary Surfaces for Ultrasound Scanning Using Discrete Differential Geometry," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7738–7746, Jul. 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9803208/

[19] J. Wang, C. Lu, Y. Lv, S. Yang, M. Zhang, and Y. Shen, "Task Space Compliant Control and Six-Dimensional Force Regulation Toward Automated Robotic Ultrasound Imaging," *IEEE Transactions on Automation Science and Engineering*, pp. 1–12, 2023, conference Name: IEEE Transactions on Automation Science and Engineering. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10148635

[20] J. Fu, I. Burzo, E. Iovene, J. Zhao, G. Ferrigno, and E. De Momi, "Optimization-Based Variable Impedance Control of Robotic Manipulator for Medical Contact Tasks," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–8, Mar. 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10456916/

[21] P. Jarosik and M. Lewandowski, "Automatic Ultrasound Guidance Based on Deep Reinforcement Learning," in *2019 IEEE International Ultrasonics Symposium (IUS)*, Oct. 2019, pp. 475–478, iSSN: 1948-5727. [Online]. Available: https://ieeexplore.ieee.org/document/8926041

[22] H. Hase, M. F. Azampour, M. Tirindelli, M. Paschali, W. Simson, E. Fatemizadeh, and N. Navab, "Ultrasound-Guided Robotic Navigation with Deep Reinforcement Learning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2020, pp. 5534–5541, iSSN: 2153-0866. [Online]. Available: https://ieeexplore.ieee.org/document/9340913

[23] K. Li, J. Wang, Y. Xu, H. Qin, D. Liu, L. Liu, and M. Q.-H. Meng, "Autonomous Navigation of an Ultrasound Probe Towards Standard Scan Planes with Deep Reinforcement Learning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, May 2021, pp. 8302–8308, iSSN: 2577-087X. [Online]. Available: https://ieeexplore.ieee.org/document/9561295

[24] G. Ning, X. Zhang, and H. Liao, "Autonomic Robotic Ultrasound Imaging System Based on Reinforcement Learning," *IEEE Transactions on Biomedical Engineering*, vol. 68, no. 9, pp. 2787–2797, Sep. 2021, conference Name: IEEE Transactions on Biomedical Engineering. [Online]. Available: https://ieeexplore.ieee.org/document/9336261

[25] K. Li, Y. Xu, J. Wang, D. Ni, L. Liu, and M. Q.-H. Meng, "Image-Guided Navigation of a Robotic Ultrasound Probe for Autonomous Spinal Sonography Using a Shadow-aware Dual-Agent Framework," *IEEE Transactions on Medical Robotics and Bionics*, vol. 4, no. 1, pp. 130–144, Feb. 2022, arXiv:2111.02167 [cs]. [Online]. Available: http://arxiv.org/abs/2111.02167

[26] Y. Bi, Z. Jiang, Y. Gao, T. Wendler, A. Karlas, and N. Navab, "VesNet-RL: Simulation-based Reinforcement Learning for Real-World US Probe Navigation," May 2022, arXiv:2205.06676 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2205.06676

[27] C. Shen, Z. Deng, J. Wang, S. Wang, and C. Chen, "Towards Autonomous Robotic Ultrasound Scanning Using the Reinforcement Learning-Based Volumetric Data Navigation Method," in *2023 WRC Symposium on Advanced Robotics and Automation (WRC SARA)*, Aug. 2023, pp. 335–340, iSSN: 2835-3358. [Online]. Available: https://ieeexplore.ieee.org/document/10261822

[28] G. Ning, H. Liang, X. Zhang, and H. Liao, "Autonomous Robotic Ultrasound Vascular Imaging System With Decoupled Control Strategy for External-Vision-Free Environments," *IEEE Transactions on Biomedical Engineering*, vol. 70, no. 11, pp. 3166–3177, Nov. 2023, conference Name: IEEE Transactions on Biomedical Engineering. [Online]. Available: https://ieeexplore.ieee.org/document/10135134

[29] C. Luo, Y. Chen, H. Cao, M. A. A. Sibahee, W. Xu, and J. Zhang, "Multi-Modal Autonomous Ultrasound Scanning for Efficient Human–Machine Fusion Interaction," *IEEE Transactions on Automation Science and Engineering*, pp. 1–12, 2024, conference Name: IEEE Transactions on Automation Science and Engineering. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10456580

[30] Z. Jiang, Y. Bi, C. Qian, Z. Zhang, and N. Navab, "Autonomous Path Planning for Intercostal Robotic Ultrasound Imaging Using Reinforcement Learning," Apr. 2024, arXiv:2404.09927 [cs]. [Online]. Available: http://arxiv.org/abs/2404.09927

[31] R. Huang, H. Yanwei, W. Xiao, C. Wang, H. Liu, and Z. Sun, "Towards Fully Autonomous Ultrasound Scanning Robot With Imitation Learning Based on Clinical Protocols," *IEEE Robotics and Automation Letters*, vol. PP, pp. 1–1, Mar. 2021.

[32] X. Deng, Y. Chen, F. Chen, and M. Li, "Learning Robotic Ultrasound Scanning Skills via Human Demonstrations and Guided Explorations," in *2021 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec. 2021, pp. 372–378, arXiv:2111.01625 [cs]. [Online]. Available: http://arxiv.org/abs/2111.01625

[33] X. Deng, Z. Lei, Y. Wang, W. Cheng, Z. Guo, C. Yang, and M. Li, "Learning ultrasound scanning skills from human demonstrations," *Science China Information Sciences*, vol. 65, no. 8, p. 184201, Aug. 2022. [Online]. Available: https://link.springer.com/10.1007/s11432-021-3363-0

[34] D. Raina, S. Chandrashekhara, R. Voyles, J. Wachs, and S. K. Saha, "Robotic Sonographer: Autonomous Robotic Ultrasound using Domain Expertise in Bayesian Optimization," in *2023 IEEE*

International Conference on Robotics and Automation (ICRA), May 2023, pp. 6909–6915. [Online]. Available: https://ieeexplore.ieee.org/document/10161542

[35] X. Deng, J. Jiang, W. Cheng, and M. Li, "Learning Autonomous Ultrasound via Latent Task Representation and Robotic Skills Adaptation," Jul. 2023, arXiv:2307.13323 [cs]. [Online]. Available: http://arxiv.org/abs/2307.13323

[36] W. Si, C. Guo, N. Wang, M. Yang, R. Harris, and C. Yang, "A Unified Deep Imitation Learning and Control Framework for Robot-Assisted Sonography," in *2023 IEEE International Conference on Development and Learning (ICDL)*, Nov. 2023, pp. 318–323. [Online]. Available: https://ieeexplore.ieee.org/document/10364497

[37] G. Ning, H. Liang, X. Zhang, and H. Liao, "Inverse-Reinforcement-Learning-Based Robotic Ultrasound Active Compliance Control in Uncertain Environments," *IEEE Transactions on Industrial Electronics*, vol. 71, pp. 1686–1696, Mar. 2023.

[38] Z. Jiang, Y. Bi, M. Zhou, Y. Hu, M. Burke, and N. Navab, "Intelligent robotic sonographer: Mutual information-based disentangled reward learning from few demonstrations," *The International Journal of Robotics Research*, p. 02783649231223547, Jan. 2024. [Online]. Available: http://journals.sagepub.com/doi/10.1177/02783649231223547

[39] Q. Huang, J. Lan, and X. Li, "Robotic Arm Based Automatic Ultrasound Scanning for Three-Dimensional Imaging," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1173–1182, Feb. 2019, conference Name: IEEE Transactions on Industrial Informatics. [Online]. Available: https://ieeexplore.ieee.org/document/8472788

[40] Z. Jiang, Z. Li, M. Grimm, M. Zhou, M. Esposito, W. Wein, W. Stechele, T. Wendler, and N. Navab, "Autonomous Robotic Screening of Tubular Structures based only on Real-Time Ultrasound Imaging Feedback," Jun. 2021, arXiv:2011.00099 [cs]. [Online]. Available: http://arxiv.org/abs/2011.00099

[41] Z. Jiang, Y. Gao, L. Xie, and N. Navab, "Towards Autonomous Atlas-based Ultrasound Acquisitions in Presence of Articulated Motion," Aug. 2022, arXiv:2208.05399 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2208.05399

[42] J. Tan, Y. Li, B. Li, Y. Leng, J. Peng, J. Wu, B. Luo, X. Chen, Y. Rong, and C. Fu, "Automatic Generation of Autonomous Ultrasound Scanning Trajectory Based on 3-D Point Cloud," *IEEE Transactions on Medical Robotics and Bionics*, vol. 4, no. 4, pp. 976–990, Oct. 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9922662/

[43] M. Chen, Y. Huang, J. Chen, T. Zhou, J. Chen, and H. Liu, "Fully Robotized 3D Ultrasound Image Acquisition for Artery," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, May 2023, pp. 2690–2696. [Online]. Available: https://ieeexplore.ieee.org/document/10161148

[44] J. Tan, B. Li, Y. Li, B. Li, X. Chen, J. Wu, B. Luo, Y. Leng, Y. Rong, and C. Fu, "A Flexible and Fully Autonomous Breast Ultrasound Scanning System," *IEEE Transactions on Automation Science and Engineering*, vol. 20, no. 3, pp. 1920–1933, Jul. 2022, conference Name: IEEE Transactions on Automation Science and Engineering. [Online]. Available: https://ieeexplore.ieee.org/document/9830629

[45] Z. Wang, B. Zhao, P. Zhang, L. Yao, Q. Wang, B. Li, M. Q.-H. Meng, and Y. Hu, "Full-Coverage Path Planning and Stable Interaction Control for Automated Robotic Breast Ultrasound Scanning," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 7, pp. 7051–7061, Jul. 2023, conference Name: IEEE Transactions on Industrial Electronics. [Online]. Available: https://ieeexplore.ieee.org/document/9889084

[46] X. Tang, H. Wang, J. Luo, J. Jiang, F. Nian, L. Qi, L. Sang, and Z. Gan, "Autonomous ultrasound scanning robotic system based on human posture recognition and image servo control: an application for cardiac imaging," *Frontiers in Robotics and AI*, vol. 11, May 2024, publisher: Frontiers. [Online]. Available: https://www.frontiersin.org/journals/robotics-and-ai/articles/10.3389/frobt.2024.1383732/full

[47] Z. Jiang, N. Danis, Y. Bi, M. Zhou, M. Kroenke, T. Wendler, and N. Navab, "Precise Repositioning of Robotic Ultrasound: Improving Registration-based Motion Compensation using Ultrasound Confidence Optimization," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–11, Sep. 2022, arXiv:2208.05383 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2208.05383

[48] X. Zhang, H. Du, D. Guo, and Y. Zhang, "Path Planning of Breast

Ultrasound Scanning Based on Three-Dimensional Point Cloud Model," in *2024 IEEE International Conference on Mechatronics and Automation (ICMA)*, Aug. 2024, pp. 459–464, iSSN: 2152-744X. [Online]. Available: https://ieeexplore.ieee.org/document/10633205/?arnumber=10633205

[49] J. Tan, H. Qin, X. Chen, J. Li, Y. Li, B. Li, Y. Leng, and C. Fu, "Point Cloud Segmentation of Breast Ultrasound Regions to be Scanned by Fusing 2D Image Instance Segmentation and Keypoint Detection," in *2023 International Conference on Advanced Robotics and Mechatronics (ICARM)*, Jul. 2023, pp. 669–674. [Online]. Available: https://ieeexplore.ieee.org/document/10218846/?arnumber=10218846

[50] Z. Jiang, Y. Zhou, Y. Bi, M. Zhou, T. Wendler, and N. Navab, "Deformation-Aware Robotic 3D Ultrasound," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7675–7682, Oct. 2021, arXiv:2107.08411 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2107.08411

[51] Z. Jiang, Y. Zhou, D. Cao, and N. Navab, "DefCor-Net: Physics-Aware Ultrasound Deformation Correction," Aug. 2023, arXiv:2308.03865 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2308.03865

[52] D. Sun, A. Cappellari, B. Lan, M. Abayazid, S. Stramigioli, and K. Niu, "Automatic Robotic Ultrasound for 3D Musculoskeletal Reconstruction: A Comprehensive Framework," *Technologies*, vol. 13, no. 2, p. 70, Feb. 2025, number: 2 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2227-7080/13/2/70

[53] S. Schaal, "Dynamic Movement Primitives -A Framework for Motor Control in Humans and Humanoid Robotics," in *Adaptive Motion of Animals and Machines*, H. Kimura, K. Tsuchiya, A. Ishiguro, and H. Witte, Eds. Tokyo: Springer-Verlag, 2006, pp. 261–280. [Online]. Available: http://link.springer.com/10.1007/4-431-31381-8_23

[54] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann, "Probabilistic Movement Primitives," in *Advances in Neural Information Processing Systems*, vol. 26. Curran Associates, Inc., 2013. [Online]. Available: https://www.ias.tu-darmstadt.de/uploads/Publications/Paraschos_NIPS_2013a.pdf

[55] Y. Huang, L. Rozo, J. Silvério, and D. G. Caldwell, "Kernelized Movement Primitives," Mar. 2018, arXiv:1708.08638 [cs]. [Online]. Available: http://arxiv.org/abs/1708.08638

[56] S. Calinon, "A tutorial on task-parameterized movement learning and retrieval," *Intelligent Service Robotics*, vol. 9, no. 1, pp. 1–29, Jan. 2016. [Online]. Available: http://link.springer.com/10.1007/s11370-015-0187-9

[57] S. Arora and P. Doshi, "A Survey of Inverse Reinforcement Learning: Challenges, Methods and Progress," Nov. 2020, arXiv:1806.06877 [cs]. [Online]. Available: http://arxiv.org/abs/1806.06877

[58] R. Donald, B. Hertel, S. Misenti, G. Yan, and R. Azadeh, "An Adaptive Framework for Manipulator Skill Reproduction in Dynamic Environments," in *2024 21st International Conference on Ubiquitous Robots (UR)*, Jun. 2024, pp. 498–503. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10597445

[59] T. Li and N. Figueroa, "Task Generalization with Stability Guarantees via Elastic Dynamical System Motion Policies," Sep. 2023.

[60] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel, "Laplacian surface editing," in *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*. Nice France: ACM, Jul. 2004, pp. 175–184. [Online]. Available: https://dl.acm.org/doi/10.1145/1057432.1057456

[61] T. Nierhoff, S. Hirche, and Y. Nakamura, "Spatial adaption of robot trajectories based on laplacian trajectory editing," *Autonomous Robots*, vol. 40, no. 1, pp. 159–173, Jan. 2016. [Online]. Available: http://link.springer.com/10.1007/s10514-015-9442-3

[62] T. L. A. v. d. Heuvel, "Automated low-cost ultrasound: improving antenatal care in resource-limited settings," 2019, accepted: 2018-12-20T21:44:33Z. [Online]. Available: https://repository.ubn.ru.nl/handle/2066/199000

[63] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann, "MediaPipe: A Framework for Building Perception Pipelines," Jun. 2019, arXiv:1906.08172 [cs]. [Online]. Available: http://arxiv.org/abs/1906.08172

[64] R. B. Rusu, N. Blodow, and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D registration," in *2009 IEEE International Conference on Robotics and Automation*, May 2009, pp. 3212–3217, iSSN:

1050-4729. [Online]. Available: https://ieeexplore.ieee.org/document/5152473

[65] A. Myronenko and X. Song, "Point Set Registration: Coherent Point Drift," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2262–2275, Dec. 2010, conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence. [Online]. Available: https://ieeexplore.ieee.org/document/5432191/?arnumber=5432191

[66] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image and Vision Computing*, vol. 10, no. 3, pp. 145–155, Apr. 1992. [Online]. Available: https://www.sciencedirect.com/science/article/pii/026288569290066C

[67] G. Franzese, R. Prakash, and J. Kober, "Generalization of Task Parameterized Dynamical Systems using Gaussian Process Transportation," Apr. 2024, arXiv:2404.13458 [cs]. [Online]. Available: http://arxiv.org/abs/2404.13458

[68] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nav.3800020109. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800020109

[69] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppe, A. Albu-Schaeffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald, and G. Hirzinger, "The KUKA-DLR Lightweight Robot arm - a new reference platform for robotics research and manufacturing," in *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*, Jun. 2010, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/document/5756872/?arnumber=5756872

[70] J. Jiang, J. Luo, H. Wang, X. Tang, F. Nian, and L. Qi, "Force Tracking Control Method for Robotic Ultrasound Scanning System under Soft Uncertain Environment," *Actuators*, vol. 13, no. 2, p. 62, Feb. 2024, number: 2 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2076-0825/13/2/62

[71] A. Karamalis, W. Wein, T. Klein, and N. Navab, "Ultrasound confidence maps using random walks," *Medical Image Analysis*, vol. 16, no. 6, pp. 1101–1112, Aug. 2012. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1361841512000977

[72] P. Chatelain, A. Krupa, and N. Navab, "Confidence-Driven Control of an Ultrasound Probe," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1410–1424, Dec. 2017, conference Name: IEEE Transactions on Robotics. [Online]. Available: https://ieeexplore.ieee.org/document/8002599

2

APPENDIX

2.0.1. Phantoms Used In Experimental Setup

Figure 2.1 shows the two anatomical phantoms used in this study, representing different anatomical variations. The male and female phantoms differ in size and geometry, introducing variability in surface shape and curvature. This variation was used to evaluate the system's ability to generalize trajectory adaptation across different subjects. Both phantoms consist of a layer of silicone placed over a rigid surface, spanning a hole in the abdomen where intestines are located. They are typically used for laparoscopic experiments or training.

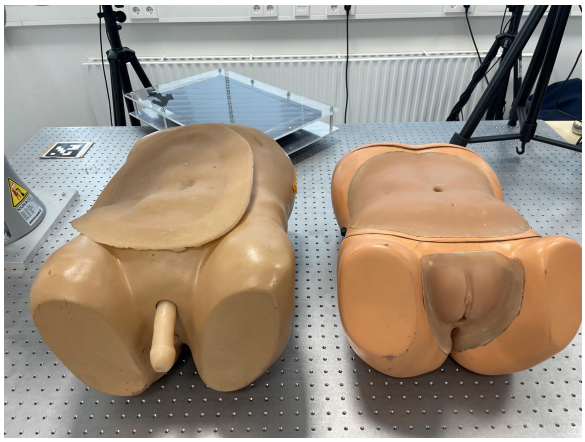
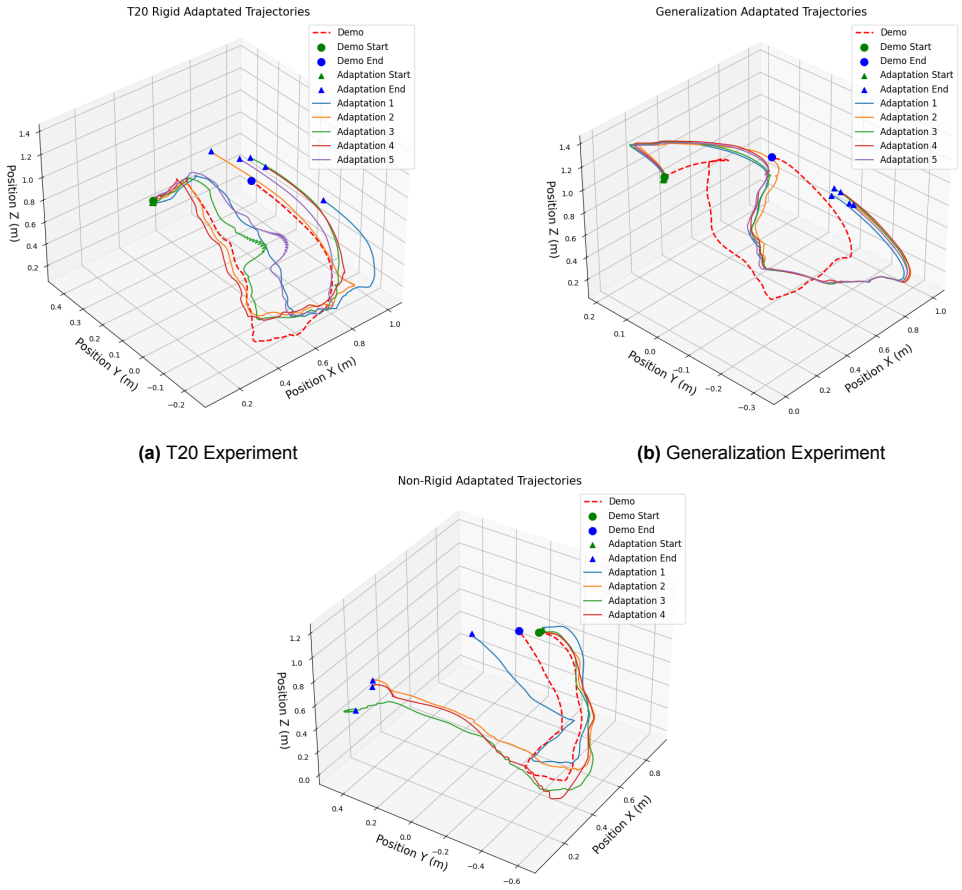


Figure 2.1: The two phantoms used in this study, illustrating differences in size and geometry that challenge the system's ability to generalize across anatomical variations.

2.0.2. Experimental Results

Figure 2.2 presents 3D plots of all trials across the T20, Generalization, and Non-Rigid experiments. Each plot visualizes the demonstrated trajectory alongside its adapted versions, highlighting how the system responded to different surface variations and movement types. In the T20 case, the green and purple lines, corresponding to adaptations 3 and 4, showed oscillatory motions similar to those discussed in Figure 2.3.



(a) T20 Experiment

(b) Generalization Experiment

(c) Non-Rigid Experiment

Figure 2.2: 3D visualizations of all trials across the T20, Generalization, and Non-Rigid experiments, showing the demonstrated and adapted trajectories in each scenario.

2.0.3. Singularities and Kinematic Limitations in Trajectory Execution

Since LTE modifies the trajectory only in Cartesian space, it does not account for workspace constraints, joint limits, or singularity avoidance. As a result, in some cases, the adapted trajectory brought the robot close to singularities, where small Cartesian displacements required large joint movements. This led to sudden spikes in joint velocity and acceleration, which, when exceeding the robot's limits, triggered safety stops and caused disconnections. These unstable motions were localized to specific trajectory segments but significantly affected acceleration profiles, as seen in Figure 2.3, where large oscillations are present.

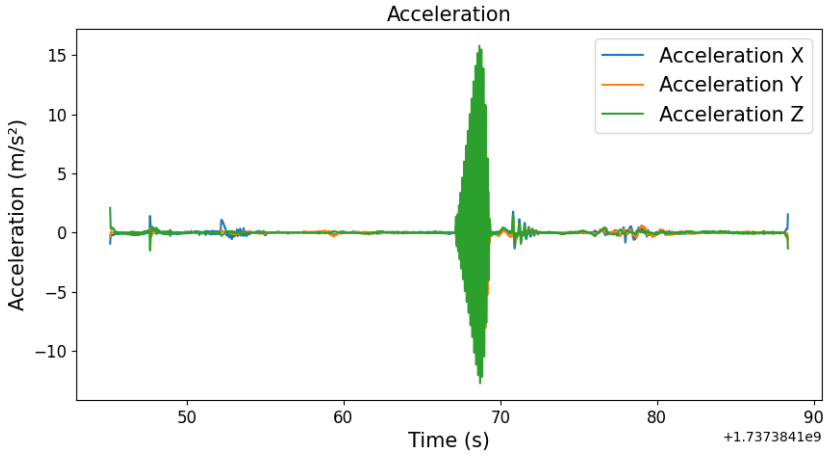


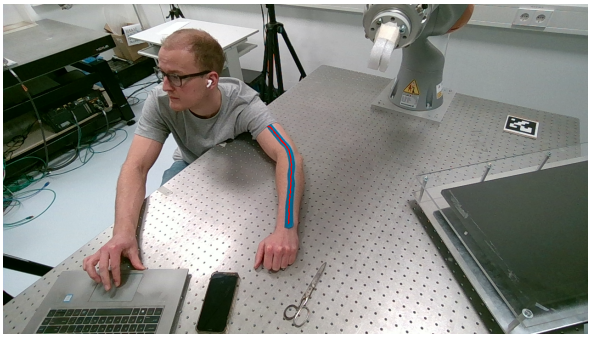
Figure 2.3: Acceleration profile of a trial where the adapted trajectory exhibited oscillatory acceleration. While this trajectory did not lead to a singularity, the repeated fluctuations in acceleration indicate unstable motion, which in some cases exceeded system tolerances and triggered a safety stop

2.0.4. Adaptation Accuracy/Blue Line Extraction

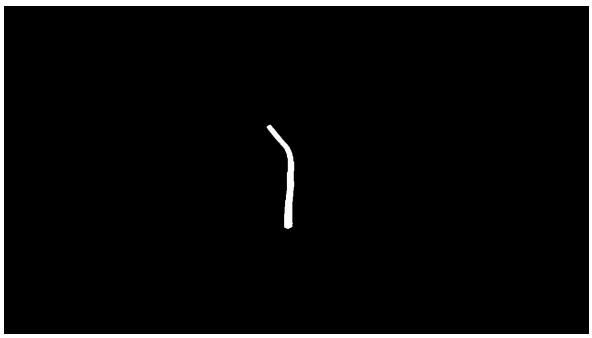
To evaluate adaptation accuracy, the blue reference line drawn with physiotherapy tape on the arm was extracted from RealSense camera images and transformed into the robot's base frame for comparison with the adapted trajectory. This process involved isolating the blue line in the image, refining its shape, and mapping it to 3D space.

The extraction pipeline applied HSV thresholding, morphological processing, and skeletonization to detect and refine the reference line. The input image was first contrast-enhanced using CLAHE [1] before being converted to HSV color space and thresholded to isolate the blue region. Morphological closing filled small gaps, and contours were filtered based on area to remove noise.

Once the blue region was segmented, a binary mask of the detected area was skeletonized to extract the centerline, which was further smoothed using cubic spline interpolation. The corresponding 3D coordinates were then obtained via depth deprojection and transformed into the robot's coordinate frame. A visual example is shown in Figure 2.4.



(a) Camera image of the detected blue reference line



(b) Corresponding mask of the detected blue line

Figure 2.4: Camera image and corresponding mask of the detected blue reference line, illustrating the segmentation process used for trajectory adaptation.

2.0.5. Computation Times Full Tables

Table 2.1 presents the computation times for each processing component across the three scenarios. Coherent Point Drift (CPD) registration in the generalization scenario is the most time-consuming step, requiring over 34 seconds. This is due to CPD's iterative optimization, which estimates point correspondences probabilistically, making it significantly slower than the rigid alignment method using RANSAC with FPFH. In contrast, Laplacian Trajectory Editing (LTE) demonstrates consistent performance across all scenarios, with fitting times ranging from 206.98 ms to 421.87 ms and trajectory updates remaining stable around 30–40 ms.

Table 2.1: Computation Times (ms) Across Scenarios

Component	**Rigid (ms)**	**Generalization (ms)**	**Non-Rigid (ms)**
RS_phantom/2D Object Detection	3.78	5.62	**X**
RS_phantom/Bounding Box-based Point Cloud Filtering	590.13	683.15	**X**
RS_phantom/Frame Extraction	14.50	10.91	**X**
RS_phantom/Point Cloud Transformation/Alignment	333.24	385.49	**X**
RS_mediapipe/Frame Extraction	**X**	**X**	8.52
RS_mediapipe/Keypoint Extraction	**X**	**X**	46.10
RS_mediapipe/PCD Publishing	**X**	**X**	0.26
RS_mediapipe/Point Cloud Generation	**X**	**X**	0.26
RS_mediapipe/Point Cloud Transformation/Alignment	**X**	**X**	0.08
PCD_rigid/FPFH Matching	797.25	**X**	**X**
PCD_rigid/Downsampling & Chamfer Distance	10.11	**X**	**X**
PCD_rigid/ICP Fine Alignment	6.55	**X**	**X**
PCD_rigid/PCD Message Handling & Preprocessing	269.82	**X**	**X**
PCD_rigid/Reordering Point Clouds	7.23	**X**	**X**
PCD_dynamic/CPD Registration	**X**	34004.28	**X**
PCD_dynamic/Downsampling & Chamfer Distance	**X**	6435.06	**X**
PCD_dynamic/ICP Fine Alignment	**X**	13.87	**X**
PCD_dynamic/PCD Message Handling & Preprocessing	**X**	0.0013	0.0019
PCD_dynamic/Reordering Point Clouds	**X**	11.12	**X**
PCD_keypoints/Keypoints entire pipeline	**X**	**X**	0.37
LTE/Fitting	206.98	421.87	383.49
LTE/Trajectory Update	33.06	40.85	31.42

References

[1] S. Pizer, R. Johnston, J. Ericksen, B. Yankaskas, and K. Muller, “Contrast-limited adaptive histogram equalization: Speed and effectiveness,” in *[1990] Proceedings of the First Conference on Visualization in Biomedical Computing*, May 1990, pp. 337–345. DOI: 10.1109/VBC.1990.109340. [Online]. Available: https://ieeexplore.ieee.org/document/109340.