

Intuitive Telemanipulation of Hyper-Redundant Snake Robots within Locomotion and Reorientation using Task-Priority Inverse Kinematics

Tim-Lukas Habich, Melvin Hueter, Moritz Schappler and Svenja Spindeldreier¹

Abstract—Snake robots offer considerable potential for endoscopic interventions due to their ability to follow curvilinear paths. Telemanipulation is an open problem due to hyper-redundancy, as input devices only allow a specification of six degrees of freedom. Our work addresses this by presenting a unified telemanipulation strategy which enables follow-the-leader locomotion *and* reorientation keeping the shape change as small as possible. The basis for this is a novel shape-fitting approach for solving the inverse kinematics in only a few milliseconds. Shape fitting is performed by maximizing the similarity of two curves using Fréchet distance while simultaneously specifying the position and orientation of the end effector. Telemanipulation performance is investigated in a study in which 14 participants controlled a simulated snake robot to locomote into the target area. In a final validation, pivot reorientation within the target area is addressed.

I. INTRODUCTION

Minimally invasive surgery (MIS) has changed medicine sustainably. Intraluminal procedures offer a reduction of incisions and postoperative pain which accelerates recovery. This is made possible by access via natural orifices, resulting in two key demands: Endoscopes require maneuverability to reach the target area and sufficient stiffness for tissue manipulation. Conventional flexible endoscopes have high compliance at the actuated distal end due to the shaft's passive structure [1]. Identical problems occur in the boroscopy, e.g. of aircraft engines. Reaching the compressor blades is possible due to the flexible tube, but the necessary stiffness for surface processing is lacking. One solution is a change from a passive shaft to a fully actuated structure. Continuum robots [2] and hyper-redundant (discrete) snake robots [3], as the one shown in Fig. 1(a), fulfill these requirements. Numerous works focus on snake-motion imitation, e.g. realizing sidewinding and lateral undulation via environmental contact [4] or the repulsion at the environment [5]. These approaches are not applicable for MIS: contact with the tissue should be reduced to avoid injuries. Locomotion should be provided by an external feeder and minimization of tissue contact realized by *fitting the snake's shape* to the anatomical pathway.

For shape control, snake robots can be modeled using a continuous backbone curve that takes into account macroscopic geometric features [3], [6]. In [7], a parameterized spatial curve is used to specify a desired shape. A similar approach generates the desired shape by interpolating shape control points and successively aligning the robot along the curve [8]. Tang et al. [9] also use waypoints as input, form

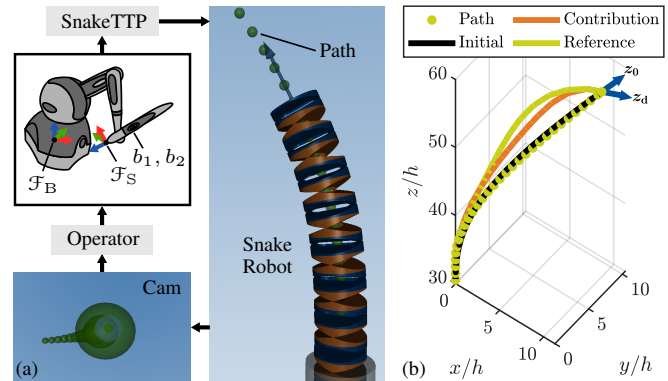


Fig. 1. (a) Intuitive telemanipulation with SnakeTTP using an input device (frames: base \mathcal{F}_B , stylus \mathcal{F}_S). The snake's tip is controlled by orientation input ${}^B R_S$. A camera at the distal end provides visual feedback. Buttons activate the feeder for locomotion (b_1) or realize pivot reorientation (b_2). Green spheres indicate a desired curvilinear path to be followed. (b) Less shape change is realized during reconfiguration compared to a reference.

the target shape using spline interpolation and determine the joint angles with kinematic considerations. Alternatively, an optimization problem can be formulated to minimize the distances between the current link positions and target positions on a given backbone [10]. All these approaches are not suitable for telemanipulating snake robots for one reason: the assumption of a known desired snake shape. The target shape is not known a priori and must be created incrementally via user input. This is true both during locomotion into the target area and during *reorientation within the target area*, as pivot movements will inherently change the shape of the robot.

To counter this, follow-the-leader (FTL) motion was formulated [11]. Controlling the snake tip by a joystick enables whole-body collision avoidance by having all links spatially follow the tip. Many groups [12]–[14] have researched FTL movements for MIS or inspection tasks based on this idea. Palmer et al. [15] propose a tip-following method for continuum robots using sequential quadratic programming. The user can control the tip via a two-axis gamepad and path-tracking is implemented using a geometric cost function.

However, telemanipulating via gamepads or joysticks with two degrees of freedom (DoF) in a classical FTL fashion [11] is not promising for two reasons: First, such devices allow only rudimentary haptic feedback (vibration). A precise feedback of the 3D interaction force between instrument and tissue is impossible. This is essential for robot-assisted surgery, as it reduces intraoperative injuries and improves surgical performance [16]. Kwok et al. [17] therefore use a six-DoF haptic input device for telemanipulation and realize

¹All authors are with the Leibniz University Hannover, Institute of Mechatronic Systems, 30823 Garbsen, Germany, tim-lukas.habich@imes.uni-hannover.de

motion planning with dynamic active constraints. To consider tissue deformation, pre- and intraoperative imaging (e.g. computer tomography of the entire environment) is required. Such imaging is not always available and telemanipulation of snake robots for MIS or inspection of industrial machines should be possible without such external sensing.

Second, a reorientation of the snake tip at a *constant* end-effector (EE) position is not possible during FTL. Such a pivot motion (with instrument tip as pivot point) is important if the target area (e.g. a polyp to be ablated) is outside the camera view and the *pointing direction* should be adjusted. This reconfiguration task requires online-planned shape fitting with consideration of position and orientation constraints. In [18], six different approaches for telemanipulating snake robots during ear-nose-throat surgery are compared and promising results regarding real-time capability and intuitiveness are achieved using the pseudo-inverse of the Jacobian. Due to mechanical coupling, there is only a low degree of redundancy (robot with eight actuated DoF). The tip is controlled by a six-DoF interface, though there is no formulation of shape adaptation for whole-body collision avoidance. The pivot movement is also not addressed.

Intuitive telemanipulation of highly articulated structures is still considered an open problem in robotics [19] and the core of our work, illustrated in Fig. 1. Our contributions are

- 1) formulating a novel shape-fitting method using task-priority inverse kinematics [20],
- 2) defining two different shape-fitting tasks using point-to-point correspondences and Fréchet distance and
- 3) presenting a unified method for telemanipulating snake robots for locomotion *and* reorientation (Algorithm 3).

Using the Fréchet distance in motion planning of redundant robots is not entirely new. Within [21], [22], path planning is performed such that a desired task-space path is followed using the Fréchet distance, even though obstacles are present. However, the authors only consider the EE path of a seven-DoF manipulator which is not sufficient for hyper-redundant snake robots. The latter requires setting a six-DoF EE pose while simultaneously maintaining a desired snake shape to realize whole-body collision avoidance. This is the core of our work. To the best of the authors' knowledge, this is the first time that shape fitting is solved using the Fréchet distance as similarity measure between the desired and the current backbone. The paper is structured as follows: Section II describes our telemanipulation strategy, followed by validations (Sec. III) and conclusions (Sec. IV).

II. SNAKE-ROBOT TELEMANIPULATION

After formulating the kinematics (II-A), shape fitting with orientation constraints (II-B) and the online motion planning (II-C) are presented using a snake robot with alternating single-axis pitch and yaw joints. The method is easily extensible to robots with universal (two-DoF) joints.

A. Kinematics Preliminaries

Snake robots with n discrete revolute joints are a serial chain of actuators. The Denavit-Hartenberg (DH) notation

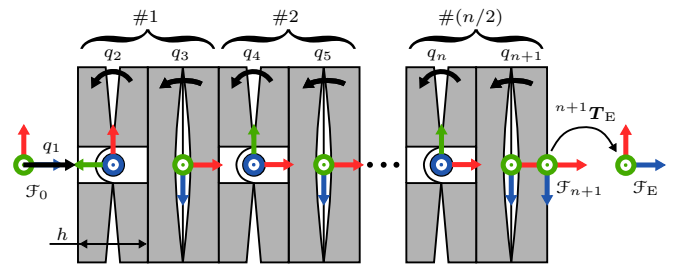


Fig. 2. Kinematic chain with n rotational actuators of height h , linear feeder q_1 , frames \mathcal{F}_i and arbitrary tool transformation ${}^{n+1}\mathbf{T}_E$

can be used to determine forward kinematics. Figure 2 illustrates the structure neglecting the actuation principle. A feeder q_1 for locomotion leads to a robot with $(n+1)$ -DoF. The longitudinal rotation $\zeta = \pi/2$ between each joint allows spatial motion for each module consisting of two actuators. The DH parameters for $n/2$ modules are given in Table I. The transformation ${}^0\mathbf{T}_i(q_i)$ depends on the first i coordinates $\mathbf{q}_i = [q_1, \dots, q_i]^T$ and contains the orientation ${}^0\mathbf{R}_i$ and position \mathbf{p}_i of link i . The six-DoF pose is denoted as

$$\mathbf{x}_i = \mathbf{f}_i(\mathbf{q}_i) = \begin{bmatrix} \mathbf{p}_i \\ \varphi_{xyz}({}^0\mathbf{R}_i) \end{bmatrix} \in \mathbb{R}^6. \quad (1)$$

Without loss of generality, the (intrinsic x-y-z" Tait-Bryan angles φ_{xyz} represent orientation. For positioning link i , the inverse kinematics solution is first defined on velocity level. We distinguish between the analytical Jacobian $\mathbf{J}_i = [\mathbf{J}_{i,t}^T, \mathbf{J}_{i,r}^T]^T \in \mathbb{R}^{6 \times i}$ (t: translational, r: rotational) and the Jacobian $\tilde{\mathbf{J}}_i \in \mathbb{R}^{6 \times i}$ of the task residual $\tilde{\mathbf{x}}_i$. Since for a 6D pose specification the naive difference between the feedback task coordinate \mathbf{x}_i and the desired quantity $\mathbf{x}_{i,d}$ is only appropriate for position and not for orientation errors, the two different Jacobians are necessary [23]. By Moore-Penrose pseudo-inversion (\dagger), the desired minimum-norm joint angle velocities are obtained by

$$\dot{\mathbf{q}}_{i,d} = \mathbf{J}_i^\dagger \dot{\mathbf{x}}_{i,d} + \tilde{\mathbf{J}}_i^\dagger \mathbf{\Lambda} \tilde{\mathbf{x}}_i \quad (2)$$

with gains $\mathbf{\Lambda} = \text{diag}(\lambda) \in \mathbb{R}^{6 \times 6}$. Closed-loop inverse kinematics (CLIK) [24] avoids the long-term drift whereby we consider only the feedback term, i.e. $\dot{\mathbf{x}}_{i,d} \equiv \mathbf{0}$. Including the feedforward component is out of the scope of our work and could further accelerate convergence.

TABLE I
DH PARAMETERS OF THE SNAKE ROBOT

i	θ_i	d_i	a_i	α_i
1	0	$q_1 + h/2$	0	$-\pi/2$
2	$q_2 - \pi/2$	0	h	$\pi/2$
3	q_3	0	h	$-\pi/2$
4	q_4	0	h	$\pi/2$
\vdots	\vdots	\vdots	\vdots	\vdots
$n-1$	q_{n-1}	0	h	$-\pi/2$
n	q_n	0	h	$\pi/2$
$n+1$	q_{n+1}	0	$h/2$	0

B. Shape Fitting using Task-Priority Inverse Kinematics

1) *Task-Priority*: Forming a desired snake shape requires the positioning of a large number of robot links i . In addition to the EE positioning task, the shape-fitting task has to be defined. The task-priority approach [25], [26] is suitable for fusing these tasks. In contrast to the approach of a weighted sum of the resulting joint angle velocities of each task, a well-defined priority order of the tasks is made possible by null-space projection. For intuitive telemanipulation, the positioning of the robot tip controlled by the user must be executed with high priority. Consequently, shape fitting takes place in the null space of the end-effector task. We use the singularity-robust task-priority framework [27] which decouples algorithmic singularities from kinematic singularities by avoiding inversion of the null-space projector. Considering k tasks¹ $\sigma_j \in \mathbb{R}^{m_j}$ with descending priority order [20], the desired joint angle velocity results in

$$\dot{q}_d = \mathbf{J}_{\mathcal{T}_1}^\dagger \Lambda_1 \tilde{\sigma}_1 + \mathbf{N}_1 \mathbf{J}_{\mathcal{T}_2}^\dagger \Lambda_2 \tilde{\sigma}_2 + \sum_{j=3}^k \mathbf{N}_{j-1}^A \mathbf{J}_{\mathcal{T}_j}^\dagger \Lambda_j \tilde{\sigma}_j \quad (3)$$

with task Jacobians $\mathbf{J}_{\mathcal{T}_j}$, null space projectors

$$\mathbf{N}_1 = \mathbf{I}_n - \mathbf{J}_{\mathcal{T}_1}^\dagger \mathbf{J}_{\mathcal{T}_1}, \mathbf{N}_{j-1}^A = \mathbf{I}_n - \mathbf{J}_{j-1}^A \mathbf{J}_{j-1}^A, \quad (4)$$

gains $\Lambda_j = \text{diag}(\lambda_j) \in \mathbb{R}^{m_j \times m_j}$ and augmented Jacobians

$$\mathbf{J}_{j-1}^A = [\mathbf{J}_{\mathcal{T}_1}^\dagger, \mathbf{J}_{\mathcal{T}_2}^\dagger, \dots, \mathbf{J}_{\mathcal{T}_{j-1}}^\dagger]^\dagger. \quad (5)$$

2) *End-Effector Pose*: As already mentioned, the tip is positioned on top priority ($j=1$) since it is directly controlled by the operator. Additionally, arbitrary safety functions could be defined on higher priorities, e.g. avoiding too abrupt movements due to incorrect input of the operator. Table II shows an overview of the task definitions used in this work. The combination of tasks is done by stacking the Jacobians and residuals. For taking a desired EE position $\mathbf{p}_{E,d} \in \mathbb{R}^3$, a 3T task is defined which can be extended by an orientation task if necessary. This can also be three-dimensional (3R), so that the full six-DoF pose of the tip can be specified. Alternatively, we consider the relevant case of a pointing task (2R), in which the rotation around the longitudinal axis of the end effector is arbitrary. For image evaluation of the endoscope camera or manipulation with a rotationally symmetrical tool, a specification of the roll angle is obsolete. Moreover, in the considered robot type no rolling motion is possible by the actuators. Omitting the longitudinal rotation as a task (3T2R) thus improves convergence if the given six-DoF pose is not achievable. The Tait-Bryan angle residual of the difference rotation $\tilde{\varphi}_\circ(\mathbf{d}\mathbf{R}_E)$ between current orientation ${}^0\mathbf{R}_E$ and desired orientation ${}^0\mathbf{R}_d$ is used as task error [23]. Depending on the task, this residual is three-dimensional ($\tilde{\varphi}_{zyx}$) or two-dimensional ($\tilde{\varphi}_{yx}$), neglecting the first component. Thus, also two different task Jacobians result by multiplying the transformation \mathbf{T}_{zyx} (3R) or \mathbf{T}_{yx} (2R) with

¹Instead of \mathbf{x}_i and $\tilde{\mathbf{x}}_i$, σ_j and $\tilde{\sigma}_j$ are used as the task variable and task residual respectively, since these do not generally represent the ($m_j = 6$)-DoF pose task. Note the difference between task number j and link index i which still denotes the number of the considered segment.

TABLE II

TASK DEFINITIONS: OBJECTIVE, RESIDUAL, JACOBIAN, DOF

Task j	σ_j	$\tilde{\sigma}_j$	$\tilde{\mathbf{J}}_j$	m_j
3T [28]	\mathbf{p}_i	$\mathbf{p}_{i,d} - \sigma_j$	$\mathbf{J}_{i,t}$	3
1T [28]	$\ \mathbf{p}_{i,d} - \mathbf{p}_i\ _2$	$0 - \sigma_j$	$\frac{(\mathbf{p}_{i,d} - \mathbf{p}_i)^\top}{-\sigma_j} \mathbf{J}_{i,t}$	1
3R [23]	$\varphi_{xyz}({}^0\mathbf{R}_i)$	$\tilde{\varphi}_{zyx}(\mathbf{d}\mathbf{R}_i)$	$\mathbf{T}_{zyx} \mathbf{J}_{i,r}$	3
2R [29]	$\varphi_{xy}({}^0\mathbf{R}_i)$	$\tilde{\varphi}_{yx}(\mathbf{d}\mathbf{R}_i)$	$\mathbf{T}_{yx} \mathbf{J}_{i,r}$	2
Fréchet	$d_F(\mathbf{P}_d, \mathbf{P})$	$0 - \sigma_j$	\mathbf{J}_F	1

the rotational part of the analytical Jacobian $\mathbf{J}_{i,r}$ of link i . A detailed derivation can be found in [29].

3) *Point-to-Point Correspondences*: Shape fitting takes place in the null space of the end-effector task. This is possible in two ways, which are presented below. For macroscopic adjustment to a desired configuration it is not necessary to position each link and, moreover, the translational positioning is sufficient. By means of 1T task every n_s -th link is positioned using the Euclidean distance between current link position \mathbf{p}_i and desired link position $\mathbf{p}_{i,d}$. Thereby, n_s serves as parameter and will be varied in III-A. The priority decreases with distance from the EE, so that at link $n - n_s$ there is the second highest priority ($j=2$), at link $n - 2n_s$ the third highest priority ($j=3$) and so on. Position \mathbf{p}_n of link n is excluded since the EE task on highest priority is already defined. Nevertheless, it is shown in III-A that too many tasks negatively influence the convergence. Note that due to the tasks for different links i , the Jacobians must be filled with zeros in the last $(n+1-i)$ columns. Hence, we use the task Jacobian

$$\mathbf{J}_{\mathcal{T}_j} = [\tilde{\mathbf{J}}_j, \mathbf{0}] \in \mathbb{R}^{m_j \times (n+1)} \forall j \in \{1, \dots, k\} \quad (6)$$

to compute $\dot{q}_d \in \mathbb{R}^{n+1}$ in (3). This shape-fitting approach is termed ‘‘Point’’ in Sec. III, since point-to-point correspondences between current and desired link positions are used.

4) *Fréchet Distance as Similarity Measure*: Using point-to-point correspondences in both the presented approach and the state-of-the-art [8]–[10] is effective as long as the desired positions of all segments are *known*. During FTL locomotion into the target area, the desired shape can be incrementally computed [11]. However, for reorientation within the target area by means of pivot movement, the desired positions of all segments are *unknown*: The shape must change due to the kinematic structure in order to establish a new pointing direction $\mathbf{z}_d = \mathbf{e}_z^{E,d}$ at constant EE position. When taking the initial shape (before reconfiguration) as the desired shape, point-to-point correspondences are disadvantageous, since the initial shape cannot be maintained if the pointing direction is changed. Especially in the tip area, there is inevitably a large shape deviation which can be seen in Fig. 1(b). To reach the position, the feeder q_1 must be actuated and the robot has to be moved forward (or backward). Consequently for positioning link i , the initial position of this segment is *not reachable*. Instead, the initial positions of following (or preceding) segments should be used as target position for link i to decrease the shape error.

Algorithm 1: Compute Jacobian \mathbf{J}_F for Fréchet task

Input : $\mathbf{q}, \delta, \mathbf{P}_d$
Output: $\mathbf{J}_F \in \mathbb{R}^{1 \times (n+1)}$

- 1 $\mathbf{P} \leftarrow$ Curve from link positions at current configuration \mathbf{q} ;
- 2 $\sigma_j \leftarrow d_F(\mathbf{P}_d, \mathbf{P})$;
- 3 **for** $i = 1$ *to* $n + 1$ **do**
- 4 $\mathbf{q}_\delta \leftarrow \mathbf{q}$;
- 5 $\mathbf{q}_\delta \leftarrow$ Add small increment δ to joint angle i ;
- 6 $\mathbf{P}_\delta \leftarrow$ Curve from link positions at configuration \mathbf{q}_δ ;
- 7 $\sigma_{j,\delta} \leftarrow d_F(\mathbf{P}_d, \mathbf{P}_\delta)$;
- 8 $\mathbf{J}_{F,i} \leftarrow (\sigma_{j,\delta} - \sigma_j) / \delta$;
- 9 **end**

To counter this, we use the Fréchet distance [30] as scalar similarity measure between two curves taking into account the positions and ordering of the points. For fast calculation, the discrete Fréchet distance² $d_F(\mathbf{P}_d, \mathbf{P})$ [31] is used as approximation. The application of this distance measure avoids the need to specify a discrete target position for each segment. Instead, the current shape \mathbf{P} and the target shape \mathbf{P}_d are described as polygonal curves whose vertices are the segment positions \mathbf{p}_i from (1). The initial shape before reconfiguration is taken as target, allowing shape fitting to be realized by minimizing the scalar similarity measure. In contrast to the shape fitting described above using point-to-point correspondences, all initial link positions are thus available to each segment i to reduce the shape error.

5) *Implementation Aspects:* To minimize the Fréchet distance, only one task has to be defined in the null space of the end-effector task. The computation of the corresponding task Jacobian \mathbf{J}_F is done by numerical differentiation with angle increment δ and is presented in Algorithm 1.

Finally, joint angle limits $q_{i,\min} \leq q_i \leq q_{i,\max}$ are considered. When a violation of joint $i = \kappa$ occurs, the corresponding joint angle q_κ is pragmatically set to its limit. This scales one entry in $\dot{\mathbf{q}}_d$ resulting in no more null-space motion [32]. Recalculating the CLIK step (3) is required, where column κ of all task Jacobians $\mathbf{J}_{\mathcal{T}_j}$ is removed [33]. The desired position change per iteration scaled by time step Δt results in $\Delta \mathbf{q}_{d,\kappa} = \dot{\mathbf{q}}_{d,\kappa} \Delta t$ and includes only the motion of the active joints. The vector κ contains binary entries and serves as an overview of which joints are active (=1) and which are inactive (=0). Within a CLIK iteration, this procedure is repeated until no limits are exceeded. After acceptance of an iteration, all joints are set back to the initial state $\kappa = \kappa_{\text{init}}$. In the case of only active joints, $\kappa_{\text{init}} = [1, \dots, 1]^T \in \mathbb{R}^{n+1}$ thus holds for n rotational actuators and one feeder.

Algorithm 2 introduces our joint-limit-aware implementation of the shape fitting as pseudo-code. The desired end-effector pose can be specified via ${}^0\mathbf{T}_{E,d}$. Depending on the shape-fitting approach, the Jacobians $\mathbf{J}_{\mathcal{T}_{1..k}}$ and residuals $\tilde{\sigma}_{1..k}$ vary. Based on this, the unified strategy for snake telemanipulation is presented in the following section.

C. Telemanipulated Locomotion and Reorientation

One contribution of our work is a unified strategy for telemanipulated locomotion and reorientation using task-priority

²www.github.com/mp4096/discrete-frechet-distance

Algorithm 2: Shape fitting using task-priority

Input : $\mathbf{q}, \mathbf{P}_d, {}^0\mathbf{T}_{E,d}, \kappa_{\text{init}}$
Output: \mathbf{q}_d

- 1 **for** iter = 1 *to* n_{iter} **do**
- 2 $\mathbf{J}_{\mathcal{T}_{1..k}}, \tilde{\sigma}_{1..k} \leftarrow$ Update with \mathbf{q} ;
- 3 flag \leftarrow Set joint-limit flag to 0;
- 4 $\kappa \leftarrow \kappa_{\text{init}}$ Reset joint states;
- 5 **while** flag = 0 **do**
- 6 $\dot{\mathbf{q}}_{d,\kappa} \leftarrow$ CLIK step (3) with active joints κ ;
- 7 $\Delta \mathbf{q}_{d,\kappa} \leftarrow \dot{\mathbf{q}}_{d,\kappa} \Delta t$ with limited step size;
- 8 $\kappa \leftarrow$ First joint index with exceeded limit or 0;
- 9 **if** $\kappa = 0$ **then**
- 10 flag \leftarrow Set joint-limit flag to 1;
- 11 Add $\Delta \mathbf{q}_{d,\kappa}$ to active joints κ in \mathbf{q} ;
- 12 **else**
- 13 Lock q_κ on limit in \mathbf{q} ;
- 14 $\kappa \leftarrow$ Set joint κ to 0;
- 15 **end**
- 16 **end**
- 17 **end**
- 18 $\mathbf{q}_d \leftarrow \mathbf{q}$;

Algorithm 3: SnakeTTP

Input : $\mathbf{q}, b_1, b_2, {}^B\mathbf{R}_S$
Output: \mathbf{q}_d

- 1 $\varphi_{xy} \leftarrow$ Pitch and yaw angles of ${}^B\mathbf{R}_S$;
- 2 $\kappa_{\text{init}} \leftarrow$ Set states of active joints to 1;
- 3 **if** b_1 **then**
- 4 // FTL locomotion
- 5 $\mathbf{q} \leftarrow$ Update pointing direction using φ_{xy} ;
- 6 $\mathbf{P}_d \leftarrow$ Curve from desired link positions [11];
- 7 ${}^0\mathbf{T}_{E,d} \leftarrow$ Desired EE pose with φ_{xy} and \mathbf{P}_d ;
- 8 $\mathbf{q}_d \leftarrow$ Shape fitting (point-to-point correspondences);
- 9 **else if** b_2 **then**
- 10 // Pivot reorientation
- 11 $\mathbf{P}_d \leftarrow$ Curve from current link positions with \mathbf{q} ;
- 12 ${}^0\mathbf{T}_{E,d} \leftarrow$ Desired EE pose with φ_{xy} and \mathbf{P}_d ;
- 13 $\mathbf{q}_d \leftarrow$ Shape fitting (Fréchet distance);
- 14 **else**
- 15 // Telemanipulate last module
- 16 $\mathbf{q} \leftarrow$ Update pointing direction using φ_{xy} ;
- 17 $\mathbf{q}_d \leftarrow \mathbf{q}$;
- 18 **end**

inverse kinematics. Algorithm 3 presents the approach with user input $\{b_1, b_2, {}^B\mathbf{R}_S\}$ and will be denoted SnakeTTP (snake telemanipulation using task-priority) in the following. Intuitive telemanipulation is realized by the six-DoF input device shown in Fig. 1(a). It would be possible to control both feeder q_1 and pointing direction e_z^E by changing the pose of the stylus ${}^B\mathbf{T}_S$. However, for the one-DoF feeder it is convenient to control it separately via buttons $b_\circ \in \{0, 1\}$. This allows the user to focus completely on setting the desired pointing direction by changing orientation ${}^B\mathbf{R}_S$. Thus, a change of e_z^S by the user directly results in a changed pointing direction of the snake robot e_z^E in steady state. Depending on the buttons pressed by the operator, there are three different cases which are presented below.

If no button is pressed, no locomotion takes place. Only the last module is controlled, resulting in a reorientation by changing the pointing direction. Determining the pitch and yaw angles $\varphi_{xy}({}^B\mathbf{R}_S)$, the joint angles of the last module

$\{q_n, q_{n+1}\}$ can be changed under consideration of joint angle limits. Consequently, no pivot movement occurs in this type of reorientation and the EE position changes.

Telemanipulated FTL motion is done by pressing button b_1 . An incremental calculation of each target position $\mathbf{p}_{i,d}$ for the next time step based on the current configuration is necessary. We use a successive approach starting at the tip [11]. The points $\mathbf{p}_{E,d}$ and $\mathbf{p}_{n,d}$ describe the desired position of the end effector and the link n , respectively. Estimating those quantities is realized by moving Δs in the pointing direction \mathbf{e}_z^E . Since two rotational actuators are necessary for spatial movement, a target position is consecutively determined for every second link. The position \mathbf{p}_i is calculated by placing the center of a sphere of radius r_s at \mathbf{p}_{i+2} and computing the intersection between the sphere and the robot structure represented as a piecewise linear curve. Thereby, r_s is the distance between the link i and link $i+2$. The position \mathbf{p}_i is then used as the new sphere center to determine \mathbf{p}_{i-2} . The calculation continues downward along the robot structure until the base is reached. The result is a matrix

$$\mathbf{P}_{d,\text{all}} = [\mathbf{p}_{E,d}, \mathbf{p}_{n,d}, \mathbf{p}_{n-2,d}, \dots, \mathbf{p}_{2,d}]^T \quad (7)$$

containing target positions for all $n/2$ modules and one EE position. The number of positions used for shape fitting is controlled via parameter n_s to obtain \mathbf{P}_d (see II-B).

After the robot moved into the target area, the operator can control the tip by means of pivot movement (button b_2). The target shape results from the current link positions to reduce the shape change due to the new pointing direction. Shape fitting is performed via 3T2R EE-task and minimization of the Fréchet distance by null-space projection.

There is one final remark to be made on the implementation. In our work we consider a snake robot which moves out of a rigid tube. This is indicated in Fig. 1(a) in the bottom part of the snake robot by the gray cylinder. Accordingly, the number of available joints increases with advanced feed position q_1 . We implement this analogously to the consideration of the joint angle limits by deleting columns of the task Jacobians. At the beginning, two rotational actuators are available for a spatial motion, so that $\boldsymbol{\kappa}_{\text{init}} = [1, \mathbf{0}, 1, 1]^T$ holds. Once q_1 has increased by one actuator height, the third actuator is activated and $\boldsymbol{\kappa}_{\text{init}} = [1, \mathbf{0}, 1, 1, 1]^T$ applies. Thus in the vector $\boldsymbol{\kappa}_{\text{init}}$ with binary entries, the available actuators are considered in each time step.

III. VALIDATION

We validate our method in three experiments using a virtual snake robot with $n=30$ rotational actuators, symmetric joint angle limits $q_{i,\text{max}} = -q_{i,\text{min}} = 30^\circ \forall i > 1$ and an actuator height $h=10$ mm. For the EE positioning errors $\{\mathcal{X}_{3T}, \mathcal{X}_{2R}, \mathcal{X}_{3R}\}$, the norm is calculated from the respective task error $\|\tilde{\boldsymbol{\sigma}}_E\|_2$ from Table II. Position and shape errors are normalized to the actuator height h to make these values more comprehensible. The proposed method was implemented using Matlab/Simulink R2021b on a 2.5 GHz Intel Core i5-10300H CPU with 16 GB of RAM running

Windows. The functions for computing the forward kinematics and Jacobians were mex-compiled. Since we do not consider an instrument at the distal end, the translational part of the tool transformation ${}^{n+1}\mathbf{T}_E$ disappears. Code³ to reproduce the results is publicly available.

A. Shape Fitting

In this section the accuracy and convergence of the two formulated task-priority approaches are compared. A set of 100 random target configurations \mathbf{q}_d without feeder ($q_1 \equiv 0$) is generated and desired link positions \mathbf{P}_d are computed using (1). Each method gets these points as input and starts in an identical configuration ($\mathbf{q} = \mathbf{0}$). Tuning the gains λ_j is out of the scope of this paper. Thus, $\lambda_j = 1 \forall j$ applies.

Figure 3 plots the shape and pose errors averaged (\diamond) for 100 shape-fitting tasks with increasing number of iterations. All strategies enable fast convergence to a desired pose. Depending on the definition of the end-effector task, the respective task error disappears. Moreover, the shape fitting done in the null space achieves for the worst method a mean Fréchet distance smaller than two actuator heights. The proposed shape fitting using the Fréchet distance shows fastest convergence and realizes a mean shape error of less than $0.8h$. Since only one scalar task is performed in the null space of the EE task, this fast convergence is possible. For point-to-point correspondences, a variation of n_s shows that the fitting is improved if every fourth link ($n_s=4$) is positioned. For $n_s=2$, the task dimension is significantly larger which affects the convergence. Hence, we will use $n_s=4$ in the following. The 3T task will be defined for the EE during FTL motion which shows fast shape convergence due to the low task dimension.

B. Telemanipulation Study

The second experiment validates the telemanipulated FTL movement. A user study was conducted with 14 participants,

³www.github.com/tlhabich/snakettp

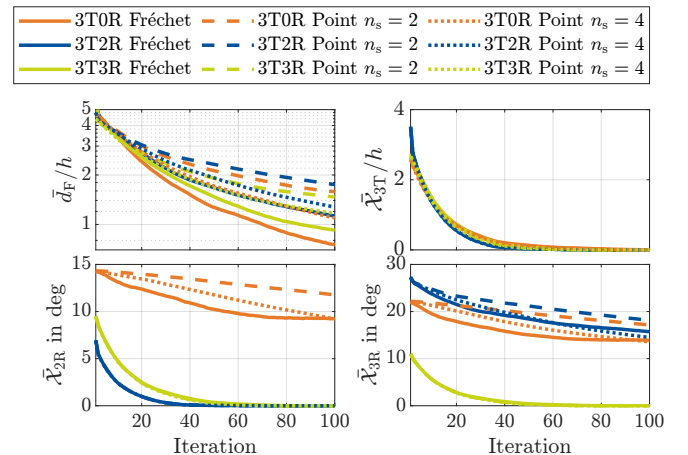


Fig. 3. Convergence of shape and pose errors averaged for 100 random desired shapes. One iteration takes on average 1.5 ms.

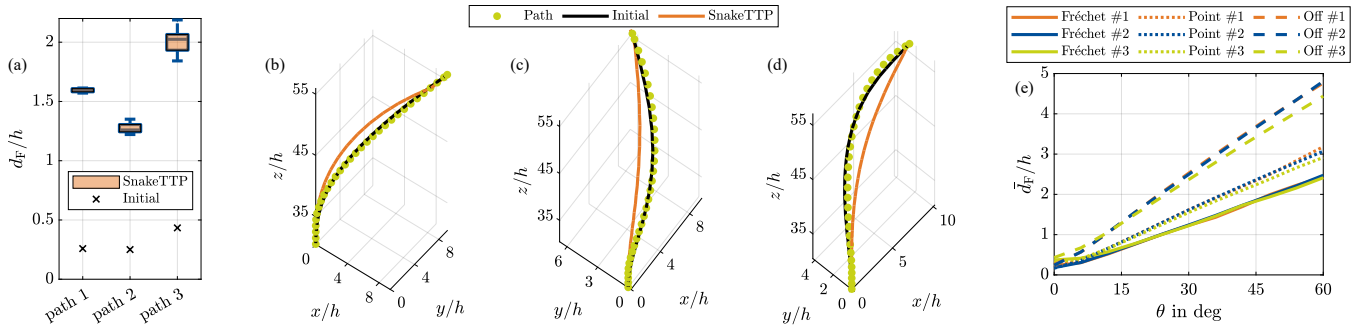


Fig. 4. (a) Normalized shape error at the end of the telemanipulation run using SnakeTTP compared to the shape-fitting method with knowledge of the entire path (Initial). (b)–(d) Spatial visualization of the three paths, the final configurations of an exemplary study participant (SnakeTTP) and the shape fitting results given knowledge of the entire paths (Initial). (e) Results of the reorientation experiment for shape fitting using Fréchet distance, point-to-point correspondences and with disabled shape fitting for three paths. The opening angle of the viewing cone θ was increased equidistantly.

all of them male engineers without any endoscopy experience. MuJoCo [34] was used as a simulation environment for the user to receive visual feedback through an endoscope camera as shown in Fig. 1(a). Three different target paths were given by green spheres for the subject to follow. The increment in pointing direction is set to $\Delta s = \frac{h}{20}$ and, therefore, few iterations are required for convergence from the current configuration to the target shape. Thus, the number of maximum iterations can be reduced to $n_{iter}=50$. The step size is limited by $\Delta t=1$ s. To increase robustness, for the maximum change of the rotational actuators $\Delta q_{r,max} = \frac{15^\circ}{n_{iter}}$ and for the feeder $\Delta q_{l,max} = \frac{2\Delta s}{n_{iter}}$ applies per CLIK step. We use 3D Systems’ Geomagic Touch as input device and the PhanTorque library [35] for Simulink integration.

The results of the study are illustrated in Fig. 4(a)–(d). For all three paths, the normalized Fréchet distance in the final configuration was determined. In addition, exemplary from one randomly chosen participant the final shapes are visualized spatially. As a comparison, the result of our shape fitting is also visualized, which receives the *entire target shape* as input (denoted as “Initial” for the next section). As expected, with knowledge of the whole path, a significantly better shape fitting is possible since the shape fitting is not created incrementally via user input. Depending on the curvature characteristics, the shape-fitting and telemanipulation performance differs. Nevertheless, macroscopic path-tracking was achieved in all telemanipulation tasks.

C. Reorientation within Target Area

In a final experiment the core of the paper will be validated: a pivot motion in order to reorient within the target area *keeping the shape change as small as possible*. For this purpose, the three target paths of the study are used again. The shape-fitting result denoted as “Initial” from Fig. 4(b)–(d) with the knowledge of the entire desired shape is given as the *initial* shape. Thus, there is a very small shape error at the beginning. The pointing direction of the initial configuration $z_0 = e_z^E$ is transformed into a new desired pointing direction $z_d = e_z^{E,d}$ using spherical coordinates $\{\theta, \phi\}$ to describe a cone-shaped area for reorientation. Thereby, $0 \leq \theta \leq \theta_{max}$ represents the opening angle of the cone and

$0 \leq \phi \leq 2\pi$ the azimuth angle. Starting at $\theta=0^\circ$, the opening angle is increased in ten equidistant steps to $\theta_{max}=60^\circ$. At each opening angle step, a complete rotation is specified in the same way by means of ϕ . Thus, 11^2 desired pointing directions z_d are specified for each path. By means of a 3T2R task for the EE, this pointing direction and the constant position can be defined to realize a pivot movement⁴.

Figure 4(e) shows the results of the experiment. The larger the deviation between the desired and initial pointing direction, the larger the Fréchet distance. The curve is almost identical for all three paths. However, using the Fréchet task allows for less shape change than using point-to-point correspondences. For the maximum orientation change considered, the shape deviation using the Fréchet approach is on average 20.1% smaller compared to the reference. Figure 1(b) shows such a reconfiguration for path #1.

IV. CONCLUSIONS

This paper presents SnakeTTP, a unified algorithm for intuitive telemanipulation realizing locomotion and pivot reorientation for endoscopic tasks. The new method based on task-priority inverse kinematics allows different position and orientation specifications at highest priority and shape fitting within the null space. The first experiment validates its fast convergence to desired EE poses and snake shapes. The proposed shape fitting using the Fréchet distance shows fastest convergence and realizes a mean shape error less than one actuator height. In a user study with 14 participants, it is shown that SnakeTTP enables online-planned locomotion with macroscopic path tracking. The last experiment demonstrates the benefit of our approach during reorientation within the target area. The novel shape-fitting approach based on the Fréchet distance reduces the shape error up to 20.1% in contrast to the classical strategy using Euclidean distance between current and desired link positions.

ACKNOWLEDGMENT

The authors acknowledge the support of this project by the German Research Foundation (Deutsche Forschungsgemeinschaft) under grant number 433586601.

⁴Illustration of the procedure: <https://youtu.be/oIpbxpJPoyQ>

REFERENCES

- [1] V. Vitiello, S.-L. Lee, T. P. Cundy, and G.-Z. Yang, "Emerging robotic platforms for minimally invasive surgery," *IEEE Reviews in Biomedical Engineering*, vol. 6, pp. 111–126, 2013.
- [2] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum robots for medical applications: A survey," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1261–1280, 2015.
- [3] G. S. Chirikjian, "Theory and applications of hyper-redundant robotic manipulators," Ph.D. dissertation, California Institute of Technology, 1992.
- [4] P. Liljebäck, Ø. Stavadahl, and K. Y. Pettersen, "Modular pneumatic snake robot 3D modelling, implementation and control," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 19–24, 2005.
- [5] A. A. Transeth, R. I. Leine, C. Glocker, K. Y. Pettersen, and P. Liljebäck, "Snake robot obstacle-aided locomotion: Modeling, simulations, and experiments," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 88–104, 2008.
- [6] H. Yamada and S. Hirose, "Study on the 3D shape of active cord mechanism," in *IEEE International Conference on Robotics and Automation*, 2006, pp. 2890–2895.
- [7] H. Mochiyama, E. Shimemura, and H. Kobayashi, "Shape control of manipulators with hyper degrees of freedom," *The International Journal of Robotics Research*, vol. 18, no. 6, pp. 584–600, 1999.
- [8] P. Liljebäck, K. Y. Pettersen, O. Stavadahl, and J. T. Gravdahl, "A 3D motion planning framework for snake robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 1100–1107.
- [9] L. Tang, L. Zhu, X. Zhu, and G. Gu, "Confined spaces path following for cable-driven snake robots with prediction lookup and interpolation algorithms," *Science China Technological Sciences*, vol. 63, no. 2, pp. 255–264, 2020.
- [10] T. Wang, B. Lin, B. Chong, J. Whitman, M. Travers, D. I. Goldman, G. Blekherman, and H. Choset, "Reconstruction of backbone curves for snake robots," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3264–3270, 2021.
- [11] R. L. Williams II and J. B. Mayhew IV, "Obstacle-free control of the hyper-redundant NASA inspection manipulator," *Proceedings of the Fifth National Conference on Applied Mechanisms and Robotics*, 1997.
- [12] S. Tappe, J. Pohlmann, J. Kotlarski, and T. Ortmaier, "Towards a follow-the-leader control for a binary actuated hyper-redundant manipulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 3195–3201.
- [13] H. Choset and W. Henning, "A follow-the-leader approach to serpentine robot motion planning," *Journal of Aerospace Engineering*, vol. 12, no. 2, pp. 65–73, 1999.
- [14] H. Xie, C. Wang, S. Li, L. Hu, and H. Yang, "A geometric approach for follow-the-leader motion of serpentine manipulator," *International Journal of Advanced Robotic Systems*, vol. 16, no. 5, 2019.
- [15] D. Palmer, S. Cobos-Guzman, and D. Axinte, "Real-time method for tip following navigation of continuum snake arm robots," *Robotics and Autonomous Systems*, vol. 62, no. 10, pp. 1478–1485, 2014.
- [16] H. Xin, J. S. Zelek, and H. Carnahan, "Laparoscopic surgery, perceptual limitations and force: A review," *First Canadian Student Conference on Biomedical Computing*, 2006.
- [17] K.-W. Kwok, K. H. Tsoi, V. Vitiello, J. Clark, G. C. T. Chow, W. Luk, and G.-Z. Yang, "Dimensionality reduction in controlling articulated snake robot for endoscopy under dynamic active constraints," *IEEE Transactions on Robotics*, vol. 29, no. 1, pp. 15–31, 2013.
- [18] P. Berthet-Rayne, K. Leibrandt, G. Gras, P. Fraitse, A. Crosnier, and G.-Z. Yang, "Inverse kinematics control methods for redundant snakelike robot teleoperation during minimally invasive surgery," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2501–2508, 2018.
- [19] A. L. Orekhov, C. Abah, and N. Simaan, "Snake-like robots for minimally invasive, single-port, and intraluminal surgeries," pp. 203–243, 2018.
- [20] P. Baerlocher and R. Boulic, "Task-priority formulations for the kinematic control of highly redundant articulated structures," in *Innovations in theory, practice and applications*. IEEE, 1998, pp. 323–329.
- [21] R. M. Holladay and S. S. Srinivasa, "Distance metrics and algorithms for task space path optimization," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 5533–5540.
- [22] R. Holladay, O. Salzman, and S. Srinivasa, "Minimizing task-space Fréchet error via efficient incremental graph search," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1999–2006, 2019.
- [23] C. Natale, *Interaction Control of Robot Manipulators: Six degrees-of-freedom tasks*. Springer, 2003, vol. 3.
- [24] P. Chiacchio, S. Chiaverini, L. Sciacivico, and B. Siciliano, "Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy," *The International Journal of Robotics Research*, vol. 10, no. 4, pp. 410–425, 1991.
- [25] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *The International Journal of Robotics Research*, vol. 6, no. 2, pp. 3–15, 1987.
- [26] B. Siciliano and J.-J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Fifth International Conference on Advanced Robotics 'Robots in Unstructured Environments*. IEEE, 1991, pp. 1211–1216 vol.2.
- [27] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 3, pp. 398–410, 1997.
- [28] G. Antonelli, *Underwater Robots*, 3rd ed., ser. Springer Tracts in Advanced Robotics. Springer, 2014.
- [29] M. Schappler, S. Tappe, and T. Ortmaier, "Resolution of functional redundancy for 3T2R robot tasks using two sets of reciprocal Euler angles," in *Advances in Mechanism and Machine Science*, ser. Mechanisms and Machine Science, T. Uhl, Ed. Springer, 2019, vol. 73, pp. 1701–1710.
- [30] M. M. Fréchet, "Sur quelques points du calcul fonctionnel," *Rendiconti del Circolo Matematico di Palermo*, vol. 22, no. 1, pp. 1–72, 1906.
- [31] T. Eiter and H. Mannila, "Computing discrete Fréchet distance," *Computer Science, Economics, Mathematics*, 1994.
- [32] P. Baerlocher and R. Boulic, "An inverse kinematics architecture enforcing an arbitrary number of strict priority levels," *The Visual Computer*, vol. 20, no. 6, pp. 402–417, 2004.
- [33] K. Leibrandt, P. Wisanuvej, G. Gras, J. Shang, C. A. Seneci, P. Giataganas, V. Vitiello, A. Darzi, and G.-Z. Yang, "Effective manipulation in confined spaces of highly articulated robotic instruments for single access surgery," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1704–1711, 2017.
- [34] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 5026–5033.
- [35] C. I. Aldana, E. Nuño, L. Basañez, and E. Romero, "Operational space consensus of multiple heterogeneous robots without velocity measurements," *Journal of the Franklin Institute*, vol. 351, no. 3, pp. 1517–1539, 2014.