

# Gesture-based Steering Framework For Redundant Soft Robots

Jiewen Lai, *Member, IEEE*, Bo Lu, Kaicheng Huang, and Henry K. Chu, *Member, IEEE*

**Abstract**—Traditional soft robot steering using different devices, such as a mouse or joystick, provides convenient ways to move the robot's tip around in omni-direction, but they may need to be more user-friendly for redundant soft robots as the null space motion can be difficult to teleoperate. In this work, we propose an intuitive approach as a framework to steer a two-segment soft-bodied robot in 3-D space using motile gestures. A lightweight hand-tracking device is employed as the input device to define the position and orientation of the robot tip, and the tip motion can be synchronized from the hand gesture – which is estimated and enhanced by a Kalman Filter (KF) and Low-pass Filter with Spherical Linear Interpolation (LPF-SLERP) – in real-time. The inverse kinematics (IK) can be rapidly computed based on the convex optimization online. Different hand gestures can be understood to execute the prescribed constrained robot motion. A hardware-in-the-loop simulator and extensive prototype experiments are presented to verify the performance and robustness of the framework. Experiment results show that the master-follower control runs with a steady 11 Hz update rate, and the measured robot's configurations are within 2 mm (for a 100-mm slender manipulator) in either axis in free space steering. This work contributes an intuitive, comprehensible, and low-cost steering framework for redundant soft robots with multi-mode motion, rapid IK computation, and satisfactory accuracy.

**Index Terms**—Soft Robotics, Tele-operation, Motion Control, Hardware-in-the-Loop Simulation.

## I. INTRODUCTION

**S**OFT-BODIED robots are usually composed of soft-touch materials, making the robots compliant and dexterous. These features are desirable in many applications, such as robot-assisted minimally invasive surgery [1]–[3], human-robot interaction [4], and object manipulation [5].

Unlike conventional robotic arms with rotary joints, extrinsically-actuated soft robots are, in most cases, driven by cables or tendons [2]. The full control of a soft robot's tip requires inverse kinematics (IK) mapping from the actuator space to the task space with the configuration space that

Manuscript received; revised; accepted. Date of publication; date of current version. The work was supported in part by the Hong Kong Research Grants Council (15213621); in part by the National Natural Science Foundation of China (62203315); in part by the Natural Science Foundation of Jiangsu Province of China (BK20220490); in part by CUHK IdeaBooster Fund Award (IDBF23ENG06); and in part by CUHK Direct Grant (4055213). (*Corresponding authors:* Henry K. Chu and Bo Lu.)

J. Lai and H. K. Chu are with the Department of Mechanical Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong. J. Lai is also with the Department of Electronic Engineering, The Chinese University of Hong Kong, Shatin, NT, Hong Kong, China. (email: [jiewen.lai@cuhk.edu.hk](mailto:jiewen.lai@cuhk.edu.hk); [henry.chu@polyu.edu.hk](mailto:henry.chu@polyu.edu.hk)).

B. Lu is with the School of Mechanical and Electrical Engineering, Robotics and Micro-systems Center, Soochow University, Suzhou, China. (email: [blu@suda.edu.cn](mailto:blu@suda.edu.cn))

K. Huang is with the Department of Materials Science and Engineering, Southern University of Science and Technology, Shenzhen, China. (e-mail: [huangkc@sustech.edu.cn](mailto:huangkc@sustech.edu.cn)).

This article has supplementary materials provided by the authors.

Digital Object Identifier

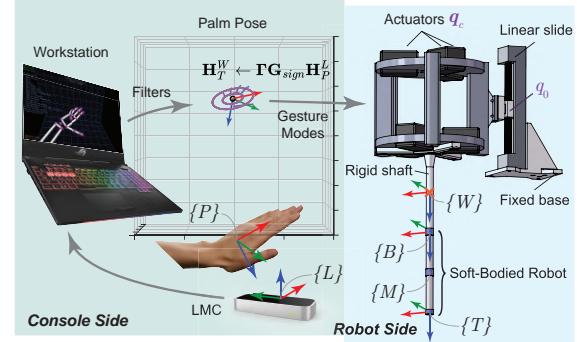


Fig. 1. A conceptual schematic shows the steering of a two-segment soft-bodied robot using hand gestures via an LMC device. **Notation** -  $\{P\}$ : Palm frame;  $\{L\}$ : LMC frame;  $\{W\}$ : World frame (spatially fixed);  $\{B\}$ : SoRo Base frame;  $\{M\}$ : SoRo Middle frame;  $\{T\}$ : SoRo Tip frame. Note that  $\{P\}$  w.r.t.  $\{L\}$  is made proportionally equivalent to  $\{T\}$  w.r.t.  $\{W\}$ .

geometrically relates them [6]. Hence, steering a multisegment soft robot can sometimes be challenging. Unless the desired tip orientation is defined, a soft robot with two articulated segments may have multiple inverse solutions for a given desired tip position [3]. Despite the difficulties, it is relatively straightforward to assign a spatial tip frame to the robot tip using articulated input devices with multiple dimensions, such as the 3-DOF Novint Falcon, the 6-DOF Geomagic Touch, and the 7-DOF Master Tool Manipulator (MTM) from the da Vinci surgical system. However, it is not straightforward to control the robot *configuration* with a single input device when the robot has more controllable DOFs than that of the input device. If additional constraints are required during the operation, it may be beyond the capabilities of these devices.

A Leap Motion®Controller (LMC) is an optical hand tracking module that captures the movements of hands with an average accuracy of 0.7 mm [7], [8]. It discerns 27 distinct hand elements, like the position and orientation of the palm, the pose of each finger, left/right hand, etc., allowing a markerless and complex gesture input for the control of rigid-bodied robots. Similar to other input devices, the majority of these studies [9]–[12] mainly focus on utilizing the LMC to teleoperate a gripper or an end-effector of the robots through different gestures. With a rich amount of sensing data available in one gesture, LMC provides a promising application in the simultaneous control of multiple segments of a hyper-redundant soft robot. A conceptual schematic in Fig. 1 illustrates the main idea of this work.

## A. Related Works

A number of works have been done towards autonomous soft-bodied robots. Without any interactive inputs, the robot's

motion can be automatically adjusted to fit the requirements based on the prescribed task and real-time feedback [13]–[16]. Based on the model predictive control, the soft continuum robots can accurately follow the pre-assigned tip trajectory [17], [18], while the tip orientations are usually not taken care of. Optimization-based motion planner can be utilized to control the tip orientation while performing tip trajectory tracking and obstacle avoidance task [19]. By learning from the physical simulation, redundant soft robots can deliver autonomous collision-free navigation with the whole body motion being constrained [20].

A real-time interactive control usually deploys a master-follower (MF) mechanism [21] for the steering. For a soft pneumatic manipulator, the user can control it using several syringes that pressurize the internal air chambers [22]. However, this method is far from intuitive, and it would be incomprehensible when there is more than one segment. By exploiting the IK, user-friendly mastering methods, such as Graphical User Interface (GUI)-based mouse-clicking [23]–[25], and joystick/game-pad steering [26]–[28] have been proposed, most of which are for surgical operation. However, these input devices with limited controllable DOF restrict many possible robot motions, unless with multiple of them [29]. Here, we argue that the intuitiveness in teleoperation is highly dependent on the DOFs of the master devices, and it should have proportionable DOFs of the master hand in an appropriate form. A 7-DOF MTM is one of the examples of exceptional intuitiveness, as it is mechanically designed to mimic the 3-D wrist motion with sufficient DOFs.

More intuitive MF systems have been presented. Ouyang *et al.* [30] designed an interactive control system to manipulate a multisegment continuum robot by defining the robot body's curve using a Geomagic Touch. It is user-friendly to draw a curve for robot pose control, but it is not ideal when continuous motion is needed. Alternatively, one can use a kinematically similar [31], or even identical robots [32], [33] to master-control the follower-robot with highly synchronized motion in real-time. Hwang and Kwon [31] tailor-made a master device to control a dual-arm surgical continuum robot. The master device was developed with a kinematic structure similar to that of the surgical instrument to compress the learning curve in the time axis. In [34], a wearable device was used to control the unidirectional bending of a single-segment soft robot with bending feedback from three flexible strain sensors attached to the soft body. To ultimately eliminate the user's learning effort, Frazelle *et al.* [32] developed a kinematically similar continuum robot as a master device to control the follower-side robot. Furukawa *et al.* [33] proposed a similar method to control a soft bending robot by using two identical robots with wire encoders to compose an MF system, where the motion of the master robot would be synchronized with that of the follower robot in real-time. More recently, teleoperation of soft robot grippers using a sensor-integrated glove with real-time finger-tip haptic feedback has been proposed [35].

However, most of the existing MF mechanisms for soft robot steering are limited to single segment [22]–[24] or insufficient DOF [17], [26], [27] cases. In contrast, some cutting-edge works for controlling the two-segment robots usually come with complicated manipulation strategies or multiple input devices [29], [30], resulting in a long learning curve and complication in designing new controller devices. In addition,

many MF mechanisms used in rigid robots aim to guide the robots in the task space, whereas the operation of soft robots, according to our survey summarized in Table S1, is defined in the configuration or joint space.

### B. Contributions

This work presents a gesture-based steering framework for redundant soft robots with a gesture motion sensing device. For the case study, we investigate the teleoperation of a two-segment cable-driven soft robot under this framework. A human hand can provide different gestures and wrist motions as semantic inputs. Using an LMC, the informative gestures and hand motions in the air can be compiled into executable commands for the robot system in real-time. Compared to other MF methods, this work highlights the multi-mode mapping of motion from hand behaviors to robot motions in 3-D space. The robot's tip pose can be synchronized as desired, and a specific configuration can be achieved. Compared with traditional input devices such as keyboards and joysticks, an LMC can provide simultaneous 3-D position and orientation input in an intuitive, straightforward way. The framework contributes to soft robots' full potential in delivering more sophisticated motions and seamless transitions between different configurations and modes (one/dual segment control) during teleoperation. Proximally constrained motions can be easily assigned. Obstacle avoidance can also be realized by incorporating the soft robot simulator. The framework provides insight to enlighten the soft robot teleoperations in a somatic way, especially for the redundant soft robots that require multi-mode steering.

In short, this work contributes a gesture-based steering framework for redundant soft robots, featuring:

- 1) An LMC-based MF-controlled system for steering a redundant cable-driven soft robot using hand behaviors on real-time optimization that necessitates rapid inverse kinematics solutions from task space to joint space.
- 2) A multi-mode gesture–robot mapping that enables the robot to perform complex poses in response to single-hand gesture semantics and wrist motion; that enables constrained robot motion like fixing positional changes in a particular axis or locking the proximal segment through the gesture changes with real-time optimization.
- 3) A filtering algorithm (Kalman filter and low-pass filter with spherical linear interpolation, LPF-SLERP) that allows continuous 3-D gesture inputs to smoothen the robot motion and seamless mode transition.

To the best of our knowledge, this will be the first work presenting the use of single-hand gestures to maneuver a redundant soft robot in 3-D space.

## II. SOFT ROBOT KINEMATICS—A COMPRESSIBLE CURVATURE MODELING APPROACH

In this work, a two-segment cable-driven soft robot (CDSR) was used. The dimension and assembly are shown in Fig. 2, and the physical properties are shown in Table I (note:  $E$  - Young's Modulus;  $K_a$  - axial stiffness;  $K_T$  - bending stiffness;  $L_k$  - original length of the  $k$ th segment). Each soft segment was actuated by 3 cables. The proximal segment had 6 cable passages, of which 3 of them were for the distal segment. A

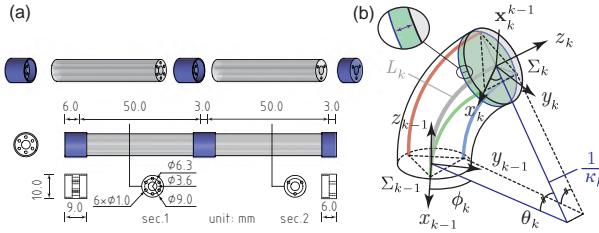


Fig. 2. (a) Assembly of the 3-D printed cable-driven soft robot [3]. The soft material (body parts) is Agilus30 PolyJet Photopolymer from Stratasys. The blue caps are made from PLA. (b) Sketch of the  $k$ th compressible soft segment (Adapted from [19]).

main hollow passage was spared at the core. The cable coupling design benefits a compact and miniaturized manipulator structure but requires real-time actuation decoupling for valid control. The soft robot has a total mass of 5 grams.

TABLE I  
PARAMETERS OF THE COMPRESSIBLE SOFT SEGMENT

$E$ [MPa]	$K_a$ [Nmm $^{-1}$ ]	$K_T$ [Nmmrad $^{-1}$ ]	$L_k$ [mm]
0.8	0.8550–0.9907	5.0211–5.5790	50

The soft robot is modeled under the compressible curvature assumption [19], which is a modified piece-wise constant curvature (PCC) model that takes the cable actuation-led undesirable axial compression of each soft segment, as well as the cascade compression, into consideration (see Fig. 2(b)). In an in-extensible PCC model, the independent  $k$ th segment can be geometrically described in the configuration space, which includes the bending angle  $\theta_k$ , bending direction  $\phi_k$ , and curvature  $\kappa_k$ . This method neglects the case of a cable-driven soft-bodied manipulator with noteworthy strain and segment coupling caused by the actuation mechanism itself. In a compressible curvature model [19], the decoupled  $(n-1)$ th segment configuration parameterized by bending angle  $\hat{\theta}_{n-1}$ , bending direction  $\hat{\phi}_{n-1}$ , and curvature  $\hat{\kappa}_{n-1}$  are given by

$$\begin{aligned} \hat{\theta}_{n-1} &= \frac{1}{K_{T,n-1}} (\mathcal{M}_{n-1}^2 + K_{T,n}^2 \theta_n^2) \\ &+ 2\mathcal{M}_{n-1} K_{T,n} \theta_n \cos(\phi_{n-1} - \phi_n + \zeta)^{\frac{1}{2}}, \end{aligned} \quad (1)$$

$$\begin{aligned} \hat{\phi}_{n-1} &= \pi - \text{atan2}\{\mathcal{M}_{n-1} \cos \phi_{n-1} + K_{T,n} \theta_n \cos(\phi_n - \zeta), \\ &\mathcal{M}_{n-1} \sin \phi_{n-1} + K_{T,n} \theta_n \sin(\phi_n - \zeta)\}, \end{aligned} \quad (2)$$

$$\hat{\kappa}_{n-1} = \frac{\hat{\theta}_{n-1}}{s_{n-1}}, \quad (3)$$

respectively, where  $s_{n-1}$  denotes the resultant compressed length of  $(n-1)$ th segment. In (1) and (2),  $\mathcal{M}_{n-1} = r\mathcal{F}_{n-1}$ , and  $\mathcal{F}_{n-1} = (f_{n-1,1}^2 + f_{n-1,2}^2 + f_{n-1,3}^2 - f_{n-1,1}f_{n-1,2} - f_{n-1,1}f_{n-1,3} - f_{n-1,2}f_{n-1,3})^{\frac{1}{2}}$ .

The  $\zeta$  is the installation angle between the neighboring segment (here  $\zeta = \pi$ ). The  $r$  denotes the equidistance between the neural axis and cable axis, and  $f_{k,i}$  represents the force of  $i$ th cable acting on the  $k$ th segment. As indicated in [19], since the in-extensible cables are always kept in tension throughout the operation, the tensile forces of cables are analogously directed as the motor actuation based on the Hooke's law, i.e.,  $\Delta f_{k,i} \propto \Delta q_{k,i}$ . Note that the decoupled

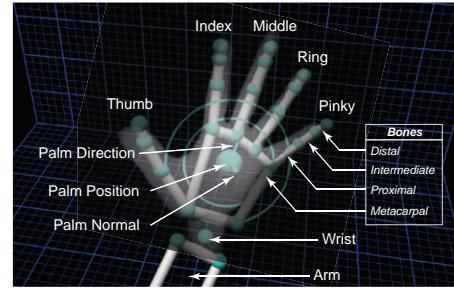


Fig. 3. The 3-D skeletal model of a left hand in LMC.

rule for the distal-most segment is given by  $\hat{\psi}_n = \psi_n$  where  $\psi_n = [\theta_n \ \phi_n \ \kappa_n]^T$ .

### III. HAND-LMC-ROBOT MAPPING

#### A. Hand Detection and Frames Transformation

An LMC uses two monochromatic IR cameras and three IR LEDs to observe a roughly hemispherical area and reconstruct the 3-D skeletal model of the recognized hands (see Fig. 3). It provides rich data on a motile hand, including the palm, fingers, velocities, etc. With a steady update rate of about 115 fps, the LMC can provide real-time input for a responsive MF mechanism. Although our human hands are dexterous with many DOFs, only a number of them will be monitored and selected as the steering input.

To provide an intuitive maneuvering experience, we make the motion of the robot tip correspond to the motion of the palm of either hand, i.e., the palm frame  $\{P\}$  with respect to (w.r.t.) the LMC frame  $\{L\}$  should coincide with the tip frame  $\{T\}$  w.r.t. the world frame  $\{W\}$ . As the hand movement has a broader range than that of the robot tip, the hand positioning should be proportionally scaled down with the tip orientation unchanged. The relationship can be expressed using the multiplication of homogeneous transformation matrices as

$$\underbrace{\begin{bmatrix} \langle \mathbf{R}_T^W \rangle & \mathbf{x}_T^W \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}}_{\mathbf{H}_T^W} := \underbrace{\begin{bmatrix} \mathbf{I}_{3 \times 3} & c_x \\ \mathbf{0}_{1 \times 3} & c_y \\ \mathbf{0}_{1 \times 3} & \frac{150}{z_P^L} + c_z \end{bmatrix}}_{\Gamma} \underbrace{\begin{bmatrix} \langle \mathbf{R}_P^L \rangle & \mathbf{x}_P^L \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}}_{\mathbf{H}_P^L}, \quad (4)$$

where

$$\mathbf{x}_P^L = [x_P^L \ y_P^L \ z_P^L]^T \quad (5)$$

represents the palm's position w.r.t.  $\{L\}$ , and  $\mathbf{H}_A^B$  denotes the homogeneous transformation matrix of  $\{A\}$  w.r.t.  $\{B\}$ ,  $\Gamma$  is a scaling matrix for position (see Sec. IV-B for more), the  $\mathbf{I}_{3 \times 3}$  returns the 3-by-3 identity matrix, and  $\langle \cdot \rangle$  denotes the normal-direction operator for the rotation matrix (i.e., only extracting palm's normal by using the function `metaData.hands(1).palm.normal()`), such that twist motion of the palm along its normal direction can be excluded.

#### B. Hand Position Estimation using Kalman Filter

Due to the unintentional low-frequency hand trembling, the hand-tracking data can be noisy even though the LMC has a high tracking accuracy. The detection noises follow

the Gaussian distribution, and its state-space model is linearly defined [36]. Therefore, the Kalman Filter (KF) can be employed for state estimation of position. KF is a recursive algorithm that uses a series of noisy measurements observed over time to estimate the unknown state of a linear dynamic system that tends to be more accurate than those using a single measurement. The KF describes the estimated state of a discrete-time controlled linear dynamic system using the linear stochastic difference equation [37]

$$\mathbf{x}_t = \mathbf{F}_t \cdot \mathbf{x}_{t-1} + \mathbf{B} \cdot \mathbf{u}_t + \mathbf{w}_t, \quad (6a)$$

$$\mathbf{z}_t = \mathbf{H} \cdot \mathbf{x}_t + \mathbf{v}_t, \quad (6b)$$

where  $\mathbf{x}_t$  denotes the state factor at time step  $t$ , corresponding to  $\mathbf{x}_P^W$  and  $\dot{\mathbf{x}}_P^W$  of the hand motion input in (4).  $\mathbf{F}_t$  is the state transition matrix that is applied to the state vector at the previous time step.  $\mathbf{B}$  is the control-input matrix which is neglected,  $\mathbf{z}_t$  denotes the measurement vector, and  $\mathbf{H}$  is the observation matrix that relates the state to the measurement. The random variables  $\mathbf{w}_t$  and  $\mathbf{v}_t$  denote the process and measurement noise, respectively, and they are assumed to be independent of each other, with Gaussian probability distributions  $p(\mathbf{w}) \sim \mathcal{N}(0, Q)$  and  $p(\mathbf{v}) \sim \mathcal{N}(0, R)$ , where  $Q$  and  $R$  represent the covariance of process noise and measurement noise, respectively.

For valid input, the user's hand shall work within the interaction zone of the LMC (depth up to 60 cm; 140×120° typical field of view [7]). Therefore, in a limited volume of space, we can assume that the infinitesimal motion of the hand satisfies either uniform linear motion (constant velocity) or uniform acceleration/deceleration (constant acceleration) motion. Thus, in 3-D space, the hand motion in a discrete-time system can be formulated as a state transition function:

$$\begin{bmatrix} x_{P,t}^L \\ \dot{x}_{P,t}^L \\ \ddot{x}_{P,t}^L \\ y_{P,t}^L \\ \dot{y}_{P,t}^L \\ \ddot{y}_{P,t}^L \\ z_{P,t}^L \\ \dot{z}_{P,t}^L \\ \ddot{z}_{P,t}^L \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \\ & & \ddots \\ & & 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ & & 0 & 1 & \Delta t \\ & & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{F}_t} \begin{bmatrix} x_{P,t-1}^L \\ \dot{x}_{P,t-1}^L \\ \ddot{x}_{P,t-1}^L \\ y_{P,t-1}^L \\ \dot{y}_{P,t-1}^L \\ \ddot{y}_{P,t-1}^L \\ z_{P,t-1}^L \\ \dot{z}_{P,t-1}^L \\ \ddot{z}_{P,t-1}^L \end{bmatrix} + \begin{bmatrix} 0 & 0 & w_x & 0 & 0 & w_y & 0 & 0 & w_z \end{bmatrix}^\top, \quad (7)$$

where  $\Delta t$  denotes the sampling time and  $(w_x, w_y, w_z)$  represents the process noise of the hand acceleration [36]. The observation function is given by

$$\begin{bmatrix} m_{x,t} \\ m_{\dot{x},t} \\ m_{y,t} \\ m_{\dot{y},t} \\ m_{z,t} \\ m_{\dot{z},t} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{H}} \begin{bmatrix} x_{P,t}^L \\ \dot{x}_{P,t}^L \\ \ddot{x}_{P,t}^L \\ y_{P,t}^L \\ \dot{y}_{P,t}^L \\ \ddot{y}_{P,t}^L \\ z_{P,t}^L \\ \dot{z}_{P,t}^L \\ \ddot{z}_{P,t}^L \end{bmatrix} + \mathbf{v}_t, \quad (8)$$

where the LHS of (8) denotes the ideal positional measurement.

As a recursive estimator, the KF updates the current estimated state from the *a priori* estimated state incorporated with the current noisy measurement. The updating process of the KF is given as follows -

- 1) Predict the *a priori* error covariance matrix  $\mathbf{P}_t^-$

$$\mathbf{P}_t^- = \mathbf{F}_t \mathbf{P}_{t-1}^- [\mathbf{F}_t]^\top + \mathbf{Q}. \quad (9)$$

- 2) Compute the Kalman gain matrix  $\mathbf{K}_t$

$$\mathbf{K}_t = \mathbf{P}_t^- [\mathbf{H}]^\top \left\{ \mathbf{H} \mathbf{P}_t^- [\mathbf{H}]^\top + \mathbf{R} \right\}^{-1}. \quad (10)$$

- 3) Update the *a posteriori* state error covariance matrix  $\mathbf{P}_t^-$

$$\mathbf{P}_t = (\mathbf{I} - \mathbf{K}_t \mathbf{H}) \mathbf{P}_t^-. \quad (11)$$

- 4) Update the *a posteriori* state estimation via measurement

$$\mathbf{x}_t = \mathbf{x}_t^- + \mathbf{K}_t (\mathbf{z}_t - \mathbf{H} \mathbf{x}_t^-), \quad (12)$$

where  $\mathbf{x}_t^-$  is the *a priori* state estimation from (6a).

Herein, at each time step, the palm position from the *a posteriori* state estimation ( $\mathbf{x}_P^L \subseteq \mathbf{x}_t$ ), instead of the raw measurement, would be adopted for the MF control.

### C. Hand Orientation Filtering using LPF-SLERP

The hand orientation  $\langle \mathbf{R}_P^L \rangle$  defined in (4) returns a 3-by-3 rotation matrix for the normal direction of the palm. Since the LMC does not provide any angular velocities, we used a low-pass filter with spherical linear interpolation (LPF-SLERP) to create sequences of orientations in the form of quaternions ( $\mathbf{q} \in \mathbb{R}^4$ ) to smoothen the orientation input for stable robot motion. The quaternion interpolation  $\mathbf{q}'$  between  $\mathbf{q}_0$  and  $\mathbf{q}_1$  from SLERP is given by [38]

$$\mathbf{q}'(h) = \text{SLERP}(\mathbf{q}_0, \mathbf{q}_1, h) := \mathbf{q}_0 \cos(h\Delta\delta) + \mathbf{q}_1 \sin(h\Delta\delta) \quad (13)$$

where  $\mathbf{q}_\perp$  denotes the ortho-normalization of  $\mathbf{q}_1$  against  $\mathbf{q}_0$ , and  $\Delta\delta$  is the angle between the two quaternions in  $\mathbb{R}^4$ :

$$\mathbf{q}_\perp = \frac{\mathbf{q}_1 - (\mathbf{q}_0^\top \mathbf{q}_1) \mathbf{q}_0}{\|\mathbf{q}_1 - (\mathbf{q}_0^\top \mathbf{q}_1) \mathbf{q}_0\|}, \quad (14)$$

$$\Delta\delta = \arccos(\mathbf{q}_0^\top \mathbf{q}_1). \quad (15)$$

The interpolation parameter  $h$  is specified as a normalized value in the closed-interval of  $[0, 1]$ . This parameter could be dynamically updated using an LPF as

$$h_{LPF} = \frac{\|\mathbf{q}_0, \mathbf{q}_1\|_D}{\pi} \cdot h_{RL} + b_{low}, \quad (16)$$

where  $\|\cdot, \cdot\|_D$  returns the angular distance between the quaternions in radians,  $h_{RL} \in [0, 1]$  denotes the range limit, and  $b_{low}$  is the low-pass parameter. Then, the current quaternion can be updated by the LPF-SLERP as  $\mathbf{q}_1 := \text{SLERP}(\mathbf{q}_0, \mathbf{q}_1, h_{LPF})$  such that a smooth change in orientation input can be obtained. In principle, an LPF-SLERP-based method for filtering hand orientation will be sufficient and computationally efficient [39] for online motion generation for robotic manipulation [40].

#### D. Gesture Shortcuts: Constrained Motion Control

When using a mouse or a conventional haptic device to steer a redundant soft robot, it is challenging to define many special motions, especially the null space motions. This work uses a series of simple gesture shortcuts to define different constraints for the robot when following the trajectory from the input devices, achieving multi-mode control that significantly facilitates the teleoperation of multisegment soft robots. A finger is considered extended if the finger keeps straight. On the contrary, a finger is considered not extended when it is curled towards the palm. On this basis, we employed a boolean vector to represent a gestured hand as

$$\mathcal{G} = [\mathcal{G}_{\text{thumb}} \quad \mathcal{G}_{\text{index}} \quad \mathcal{G}_{\text{middle}} \quad \mathcal{G}_{\text{ring}} \quad \mathcal{G}_{\text{pinky}}]^T, \quad (17)$$

where

$$\mathcal{G}_{\text{finger}} = \begin{cases} 1, & \text{extended} \\ 0, & \text{curled.} \end{cases} \quad (18)$$

The assignment of shortcut gesture should consider the comfortability—for example, the  $[0, 1, 0, 1, 0]^T$  gesture will be a torment. Although, theoretically, a single hand could provide up to 120 combinations of hand gestures, the selected gestures should be comfortable to pose and easy to memorize for the user. As shown in Fig. 4, in addition to the general palm mode, three additional simple gestures were pre-assigned to the shortcuts for different constrained robot motions:

**Palm:** General control of the position and orientation of tip frame  $\{T\}$  using the position and orientation of palm frame  $\{P\}$  with the twist motion of our hand (i.e., rotating the palm w.r.t. its center) neglected.

**V-Sign:** Constrained control of the position and orientation of tip frame  $\{T\}$  using the position and orientation of palm frame  $\{P\}$  with tip position being constrained at  $\mathbf{x}_T^W = [0 \ 0 \ Z]^T$  where  $Z$  is the only positional variable.

**Thumbs-up:** Constrained control of the position of tip frame  $\{T\}$  using the position of palm frame  $\{P\}$  with tip position being constrained at  $\mathbf{x}_T^W = [0 \ 0 \ Z]^T$  where  $Z$  is the only positional variable, and tip orientation being fixed to become identical to the world frame  $\{W\}$ , i.e., forming a straight configuration.

**Fist-pump:** Constrained control of the position and orientation of tip frame  $\{T\}$  using the position and orientation of palm frame  $\{P\}$  with the proximal segment being locked as the final updated pose before the detection of the fist. In other words, moving a fist-pump in mid-air will only control the distal segment, with the proximal segment being frozen.

With that being defined, equation (4) can be modified as

$$\mathbf{H}_T^W := \mathbf{\Gamma} \mathbf{G}_{\text{sign}} \mathbf{H}_P^L, \quad \text{if } \exists \mathcal{G}_{\text{finger}} = 1, \quad (19)$$

where

$$\mathbf{G}_{\text{sign}} = \underbrace{\begin{bmatrix} \mathbf{I}_{3 \times 3} & 1 \\ 0_{1 \times 3} & 1 \end{bmatrix}}_{\text{palm}} \text{ or } \underbrace{\begin{bmatrix} \mathbf{I}_{3 \times 3} & 0 \\ 0_{1 \times 3} & 1 \end{bmatrix}}_{\text{v-sign}} \text{ or } \underbrace{\begin{bmatrix} \langle \mathbf{R}_P^L \rangle^T & 0 \\ 0_{1 \times 3} & 1 \end{bmatrix}}_{\text{thumbs-up}}. \quad (20)$$

The fist-pump gestures have an alternative computation scheme, which will be introduced in the following subsection.

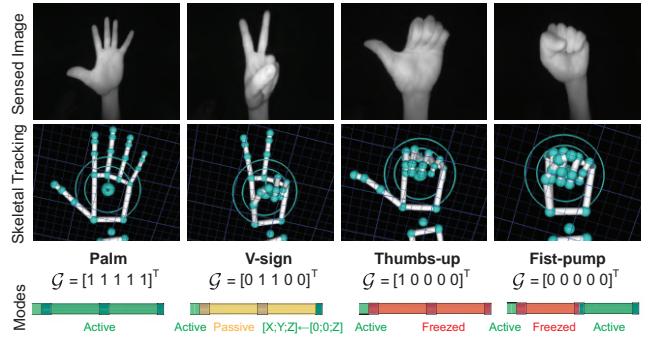


Fig. 4. Gesticulation definition for different constraints of robot motion.

#### E. Optimization-based Constrained Motion MF Control

As demonstrated in Fig. 1, the soft manipulator is perpendicularly mounted on a linear slide to form an insertion–retraction motion (i.e.  $q_0 \geq 0$ ). At an instantaneous moment of  $t$ , the actuation  $\mathbf{q}_{[t]}$  can be noted as

$$\mathbf{q}_{[t]} = [q_0 \ q_{1,1} \ q_{1,2} \ q_{1,3} \ q_{2,1} \ q_{2,2} \ q_{2,3}]^T, \quad (21)$$

where  $\mathbf{q}_c = [q_{1,1} \ \dots \ q_{2,3}]^T \in \mathbb{Q}^6$  is defined as the cable actuation. The homogeneous transformation matrix of the tip w.r.t. the world frame can be expressed as

$$\begin{aligned} \mathbf{H}_{\text{tip}}^0(\mathbf{q}) &= \mathbf{H}_{\text{base}}^0(q_0) \cdot \prod_{k=1}^2 \mathbf{H}_k^{k-1}(\mathbf{q}_c) \\ &= \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathcal{H} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{R}}_1^0 & \widehat{\mathbf{x}}_1^0 \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{R}}_2^1 & \widehat{\mathbf{x}}_2^1 \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}, \end{aligned} \quad (22)$$

where  $\mathcal{H} = [0, 0, q_0]^T$  denotes the slide's translation, and

$$\widehat{\mathbf{x}}_k^{k-1} = \mathbf{x}_k^{k-1}(\widehat{\psi}_k) = \frac{1}{\widehat{\kappa}_k} \begin{bmatrix} \cos \widehat{\phi}_k (1 - \cos \widehat{\theta}_k) \\ \sin \widehat{\phi}_k (1 - \cos \widehat{\theta}_k) \\ \sin \widehat{\theta}_k \end{bmatrix}, \quad (23)$$

$$\widehat{\mathbf{R}}_k^{k-1} = \text{Rot}(\bar{\mathbf{z}}_{k-1}, \widehat{\phi}_k) \cdot \text{Rot}(\bar{\mathbf{y}}_{k-1}, \widehat{\theta}_k) \cdot \text{Rot}(\bar{\mathbf{z}}_k, -\widehat{\phi}_k). \quad (24)$$

Given a linear tip velocity  $\dot{\mathbf{x}} \in \mathbb{R}^3$  in frame  $\{W\}$ , the corresponding velocity of actuation that drives the robot tip can be computed by

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger \dot{\mathbf{x}} + (\mathbf{I}_{7 \times 7} - \mathbf{J}^\dagger \mathbf{J}) \dot{\mathbf{q}}_{\mathcal{N}}, \quad (25)$$

where  $\mathbf{J} \in \mathbb{R}^{3 \times 7}$  is a wide Jacobian matrix inferring a redundant robot, and  $(\cdot)^\dagger$  denotes the right pseudo inverse operator. The  $\dot{\mathbf{q}}_{\mathcal{N}}$  projects the null space of  $\mathbf{J}$ . In a discrete-time system, the changes in the tip position in an infinitesimal period of time  $\Delta x$  can be approximated based on the changes of the infinitesimal actuation  $\Delta \mathbf{q}$ , such that  $\Delta \mathbf{q} = \mathbf{J}^\dagger \Delta \mathbf{x}$ , which may provide multiple solutions due to the system redundancy. Therefore, we used the damped least square (DLS) method [41] to deal with the IK to get rid of the singularity and null space problem under the suitable parameters. In our case, the DLS that solves the IK can be formulated by converging the

changes of tip positions, orientation, and joint motion as the following convex least-square optimization:

$$\begin{aligned} \arg \min_{\Delta q} & \underbrace{\|\mathbf{J}\Delta q - \Delta x\|_2^2}_{\text{position}} + \sum_{i=1}^3 \underbrace{\|\Omega_i(\Delta q_c) - \Delta \Omega_i\|_2^2}_{\text{orientation}} + \lambda^2 \|\Delta q_c\|_2^2 \\ \text{s.t. } & \mathbf{A} \cdot \Delta q \leq \mathbf{b} \quad \text{and} \quad \mathbf{q}_{\min} \leq \mathbf{q} \leq \mathbf{q}_{\max}, \end{aligned} \quad (26)$$

where  $\Delta \Omega_i = \Delta [\alpha \quad \beta \quad \gamma]^T$  denotes the desired instantaneous change of tip orientation, and  $\Omega_i(\Delta q_c)$  denotes the optimized change of tip orientation, all in Euler representation. The robot motion is bounded by the least motion associated to a non-zero damping constant  $\lambda \in \mathbb{R}^+ \setminus \{0\}$ . Based on the workspace, accuracy, and hardware limitation, we empirically set the inequality constraints as  $\mathbf{A} = \text{diag}[10^{-3}, 1, 1, 1, 1, 1]$  and  $\mathbf{b} = 10^{-2} \cdot \text{ones}(7, 1)$ , with the bounded conditions in  $\mathbf{q}_{\min} = -[0, 2, 2, 2, 2, 2, 2]^T$  and  $\mathbf{q}_{\max} = [60, 0, 0, 0, 0, 0, 0]^T$ . Similarly, the fist gesture, which defines the proximal segment to be locked, can be utilized to formulate an optimization problem by converging the changes of tip and mid-point positions and joint motion as

$$\begin{aligned} \arg \min_{\Delta q} & \underbrace{\|\mathbf{J}\Delta q - \Delta x\|_2^2}_{\text{position}} + \underbrace{\|\mathbf{J}_M \Delta q - \Delta x_M\|_2^2}_{\text{mid position}} + \underbrace{\lambda^2 \|\Delta q_c\|_2^2}_{\text{least motion}} \\ \text{s.t. } & \mathbf{A} \cdot \Delta q \leq \mathbf{b} \quad \text{and} \quad \mathbf{q}_{\min} \leq \mathbf{q} \leq \mathbf{q}_{\max}, \end{aligned} \quad (27)$$

where  $\mathbf{J}_M \Delta q$  kinematically computes the mid-point position.

With instant optimization, the consistent update of the change of desired tip position and orientation generates the real-time output of required actuation that flexes the soft robot to the proper configuration with the designated tip pose. The shortcut motions can be defined by formulating a case-switching logic as shown in Algorithm 1. Here we use  $\tilde{\mathbf{H}}$  to note the noisy raw input, and  $\mathbf{H}$  for the aftermath of denoising.

#### IV. HARDWARE-IN-THE-LOOP SIMULATION EXPERIMENT

The framework's steering performance was examined through HIL simulation, where an LMC was used as the input device to teleoperate a virtual two-segment CDSR. A Matlab-based CDSR simulator has been developed in previous work [19]. The simulator requires the cable and slide actuation as the input and demonstrates a two-segment soft robot in real-time 3-D animation. The official LeapSDK Orion 4.0.0 features a C-style API called LeapC for accessing the hand-tracking data from the LMC. In this work, the LeapC was compiled by Microsoft Visual C++ 2019 in Matlab environment using the mex-file from Matleap [42]. To achieve real-time control, we solved the optimization functions using fmincon in Matlab and interpreted the program in C. After examining different nonlinear optimization algorithms, we empirically employed the sequential quadratic programming (SQP) method as the option in fmincon for the sake of balancing computational speed and accuracy. The optimization rate was higher than the update rate of hand detection, ensuring real-time control, while the 3-D simulator was updated at about 15 Hz.

##### A. Hand Tracking

Before using the LMC for the experiment, we need to ensure reliable accuracy in hand tracking in both position and

---

#### Algorithm 1: Gesture-based rapid IK optimization

---

```

Input: Palm pose  $\tilde{\mathbf{H}}_P^L$ , Gesture Recognition  $\mathcal{G}$ .
Result: Actuation  $\mathbf{q}_{[t]} \in \mathbb{R}^{7 \times 1}$  at frame  $t$ .
1 Initialization;
2 while True do
3   Query  $\tilde{\mathbf{H}}_P^L \leftarrow \text{LMC.Matleap}(1)$ ; % One-hand detection.
4   if isEmpty( $\tilde{\mathbf{H}}_P^L$ ) == 0 then
5     Compute  $\mathbf{H}_T^W \xleftarrow{\text{KF\&LPF-SLERP}} \tilde{\mathbf{H}}_T^W$ ; %
      Real-time data estimation & filtering.
6     Compute  $\mathbf{H}_T^W \leftarrow \Gamma \mathbf{H}_P^L$ ; % Frames Transformation.
7     Assign  $\mathbf{x}_{desired}$  as  $\mathbf{x}_T^W$ ; % Desired position.
8     Assign  $\Omega_{desired}$  as  $\text{rotm2eul}(\langle \mathbf{R}_T^W \rangle)$ ; % Desired
      normal orientation in euler.
9     Query  $\mathcal{G} \leftarrow \text{get.Fingers}()$ ; % Gesture recognition.
10    if all( $\mathcal{G}$ )  $\not\subseteq \mathcal{O}$  then
11      % Palm, V-sign, Thumbs-up, and others but fist.
12       $\mathbf{H}_T^W \leftarrow \Gamma \mathbf{G}_{sign} \mathbf{H}_P^L$ ;
13      Calculate  $\mathbf{x}_T^W = \text{forwardKine}(\mathbf{q}_{current})$ ;
14      Calculate  $\Delta \mathbf{x} = \mathbf{x}_{desired} - \mathbf{x}_T^W$ ;
15      Calculate  $\Delta \Omega = \Omega_{desired} - \Omega_T^W$ ;
16      Compute
         $\Delta q \leftarrow \arg \min_{\Delta q} \{objFun(\Delta \mathbf{x}, \Delta \Omega)\}; (26)$ 
        Set  $\mathbf{q}_{current} \leftarrow \mathbf{q}_{previous} + \Delta q$ ;
        Update  $\mathbf{q}_{previous} \leftarrow \mathbf{q}_{current}$ ;
17    else
18      % Fist: Proximal segment frozen.
19      Calculate  $\mathbf{x}_M^W, \mathbf{x}_T^W = \text{forwardKine}(\mathbf{q}_{current})$ ;
20      Calculate  $\Delta \mathbf{x}_M = \mathbf{x}_{M,desired}^W - \mathbf{x}_M^W$ ;
21      Calculate  $\Delta \mathbf{x} = \mathbf{x}_{desired} - \mathbf{x}_T^W$ ;
22      Compute  $\Delta q \leftarrow$ 
         $\arg \min_{\Delta q} \{objFun_{pxFixed}(\Delta \mathbf{x}, \mathbf{x}_M^W)\}; (27)$ 
        Set  $\mathbf{q}_{current} \leftarrow \mathbf{q}_{previous} + \Delta q$ ;
        Update  $\mathbf{q}_{previous} \leftarrow \mathbf{q}_{current}$ ;
23    else
24      Assign boundary/safe values as  $\mathbf{x}_T^W$  then
        compute/set  $\mathbf{q}_{current}$ ;
25    Set  $\mathbf{q}_{[t]} \leftarrow \mathbf{q}_{current}$ ; (21)

```

---

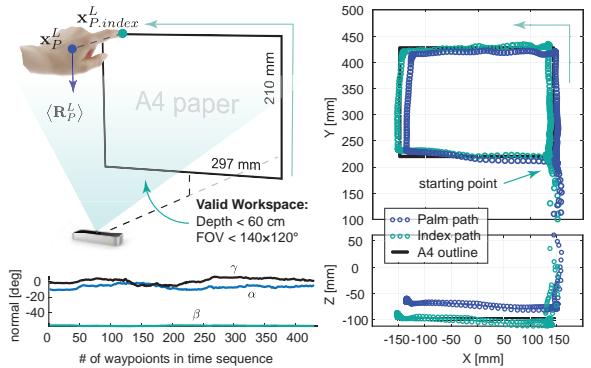


Fig. 5. A test to check the tracking accuracy of the LMC equipped with our filtering algorithm. The green and blue paths reflect the tracking performance of  $\mathbf{x}_{P,index}^L$  and  $\mathbf{x}_P^L$ , respectively. The angle diagram denotes the normal orientation of the palm along with the tracking.

orientation after the filtering. In order to do that, we can use an object with a known size as a reference and finger-track its outline. Based on the intrinsic accuracy of 0.7 mm that was learned from a LMC's study [8], the initial parameters of the KF were empirically determined as  $P = 0.7 \cdot \text{eye}(9)$  with the sampling time  $\Delta t \approx 30$  ms being updated. The initial matrices

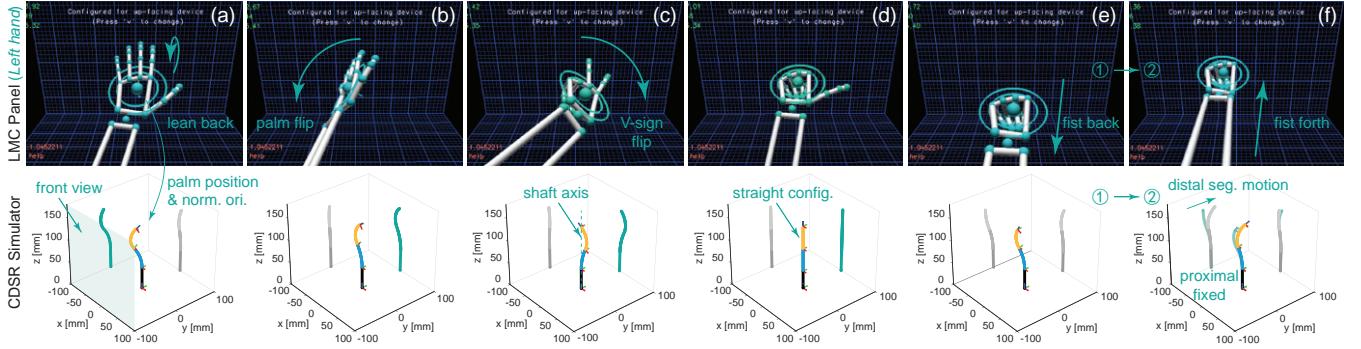


Fig. 6. Steering a cable-driven soft robot in the HIL simulation based on real-time gesture control. [Please refer to the video for better visualization] (a) **Palm gesture** with the direction pointed to the lower front and (b) to the right w.r.t. the user's view. (c) **V-sign gesture** with the direction tilted to the left. The X and Y position of the robot tip was constrained to 0 and 0, but the orientation and Z axis were maneuverable. (d) **Thumbs-up gesture** that forced the robot to maintain a straight configuration, where only Z-axis was controllable. (e-f) **Fist gesture** that fixed the proximal segment to the last-updated pose as much as possible. The position of the fist was used to control the position of the distal tip, from ① to ②.

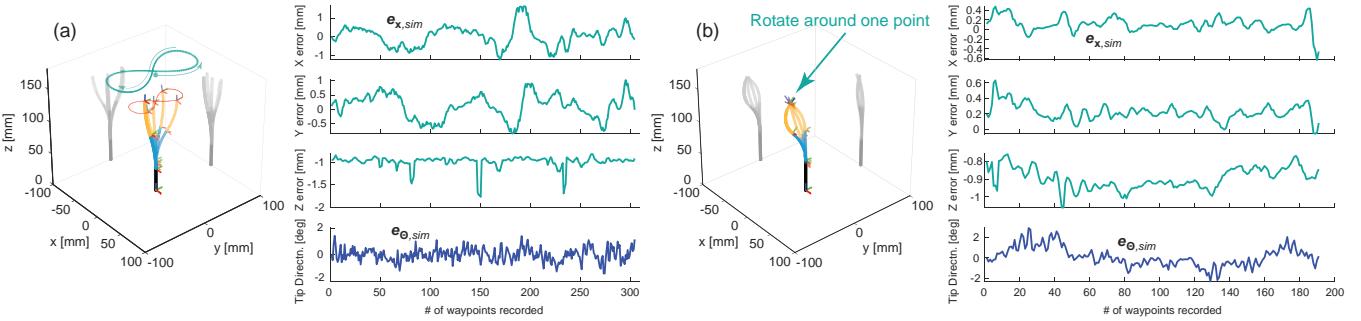


Fig. 7. Using the regular palm mode to (a) steer an  $\infty$ -path with consistent palm direction in the simulator, and (b) steering the tip to rotate around a desired point. The  $e_{x,sim}$  and  $e_{\Theta,sim}$  indicates that the HIL simulator could project an accurate configuration response to the gesture input.

$Q$  and  $R$  can be determined by the adaptive method described in [43]. As shown in Fig. 5, we used a soft A4 paper to test the tracking accuracy of the LMC with our filtering method. The sampling rate was 115 Hz (i.e., factory setting of the LMC). It can be seen that the sensed  $\mathbf{x}_{P,index}^L$  and  $\mathbf{x}_P^L$  can well-fit the dimension of the paper's edges. Here, the accuracy for finger tracking is higher than palm tracking because, for the finger, only one finger is being tracked by the LMC, and the finger can easily follow the outer edge of the paper as closely as possible to complete one full turn. On the contrary, for palm tracking, the intrinsic LMC tracking algorithm relies on tracking multiple features to compute the palm position. Besides, the exact palm position cannot be easily visualized by the user, so a discrepancy exists in the measured palm path. Also, the result also implies that it is difficult for our human hand to draw a perfectly straight line in mid-air but can stabilize the palm orientation relatively better. Therefore, a scaling matrix for position  $\Gamma$ , which has been previously mentioned in (4), would be necessary to reduce the input sensitivity in positioning. To avoid abnormal robot operations, we set the boundary for the allowable spatial position of a hand as  $30 \times 30 \times 30$  cm, and 10 cm above the LMC device. If the hand accidentally moves out of this boundary, or if the LMC cannot detect the hand, the last valid input will be used and fed to the robot.

### B. Robot Steering in HIL Simulator

We developed a reproducible HIL simulation platform that allows the real-time control of both the position and orientation

of a two-segment CDSR using a single LMC device. At each time step, the instantaneous output  $\mathbf{q}_{[t]}$  generated by algorithm 1 was assigned to the simulator. The simulator received the actuation command and animated the robot motion forwardly calculated using compressible curvature kinematics (Sec. II). A series of demonstrations are shown in Fig. 6 and in video S1 as well. The positional parameter of the scaling matrix  $\Gamma$  were set as  $c_x = 0.2$ ,  $c_y = -0.2$ , and  $c_z = -0.2$ , with  $\lambda = 1$  based on the subjective evaluation of the maneuvering sensitivity and comfortableness. The  $c$  values act as a tuning parameter to amplify or reduce the hand motion of the robot.  $\Gamma$  matrix shall be customized based on the prototype, so as the shortcuts.

The error between the tip position assignment  $\mathbf{x}_T^W$  and the tip position of the simulator  $\mathbf{x}_{T,sim}^W$  is given by  $e_{x,sim} = \mathbf{x}_T^W - \mathbf{x}_{T,sim}^W$ , and the orientation error is given by  $e_{\Theta,sim} = \Theta_T^W - \Theta_{T,sim}^W$ , where  $\mathbf{x}_{T,sim}^W(\mathbf{q}_{[t]})$  and  $\Theta_{T,sim}^W(\mathbf{q}_{[t]})$  compute the simulated tip position and orientation that employ the forward kinematics with the input of  $\mathbf{q}_{[t]}$ , respectively. We give an example in Fig. 7(a), where an  $\infty$ -path was plotted as a reference guide and the author tried to maneuver the robot tip to draw the path with consistent palm direction using the regular mode ( $\mathcal{G} = [1 \ 1 \ 1 \ 1 \ 1]^T$ ). The error diagram shows that  $e_{x,sim}$  and  $e_{\Theta,sim}$  were minor (within 1 mm and  $2^\circ$ ). In Fig. 7(b), we manipulated the tip to locate a fixed point of  $[-15, -20, 140]$  mm in  $\{W\}$ , and simultaneously rotated the tip direction around this point (i.e., null space motion) using the same regular mode. The error diagrams are also

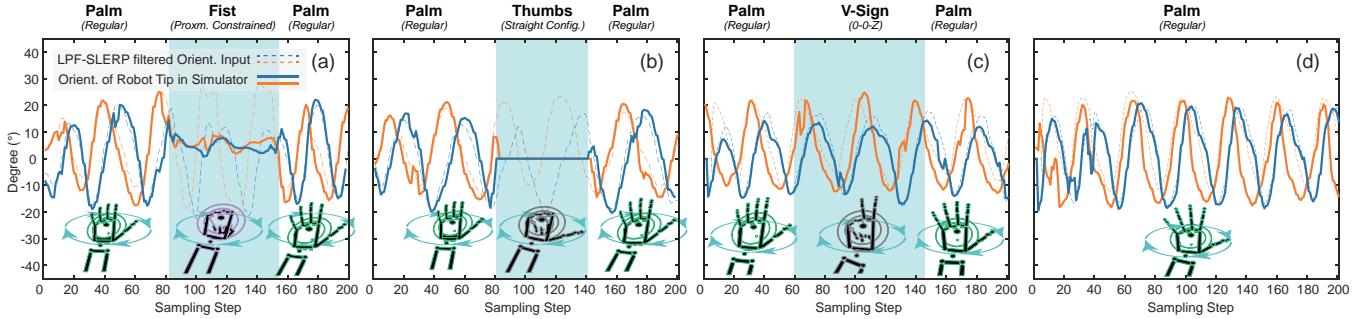


Fig. 8. Switching gestures while continuously changing the robot tip's orientation. The dashed lines represent the gesture-defined filtered orientation input ( $\alpha$  and  $\gamma$  in Euler form), and the solid lines represent the orientation of the robot tip in the simulator computed by forward kinematics from joint space – demonstrating seamless orientation transition with the filtering regardless of the sudden gesture change. (a) Palm-Fist-Palm; (b) Palm-Thumbs-Palm; (c) Palm-V-sign-Palm; (d) Regular palm modes.

found satisfactory with acceptable deviations.

As switching input modes via gesture shortcuts may be frequently used in our work, it would be interesting to see how the LPF-SLERP and simulator react to the orientation transition during gesture switching. Therefore, we tried switching input modes by changing the gesture of a dynamically rotating hand<sup>1</sup>. As shown in Fig. 8(a), we attempted to rotate the orientation input while changing the modes from “palm” to “fist” to “palm.” Since the yaw motion of the palm’s normal is invalid (see Eq. (4)), only the rolling and pitching are demonstrated. The result shows seamless, smooth transition performance during the gesture switch. When the proximal segment’s motion was restricted from any motion, the IK solver could still evolve and adapt to orientation input at full capacity while constraining the robot’s body. Fig. 8(b) depicts the result of defining similar gestures switch but using “thumbs-up” mode that calls for a straight robot configuration in the middle instead, and Fig. 8(c) for the “V-sign” mode that constrains the position assignment of  $X - Y$  plane to zero. These results provide insight into how our controller handles sudden changes in the robot’s tip orientation seamlessly during the transition between gesture modes enabled by the filters; that deviations could be gradually approximated to the desired input over time.

## V. PROTOTYPE EXPERIMENT & DISCUSSION

The prototype setup is shown in Fig. 1 and Fig. 2, which has been used in [19]. To ensure open-loop accuracy, we used servo motors with 4096 pulse/rev positioning feedback (Dynamixel XM430-W350, ROBOTIS) for the robot joints. An Intel D435 RGB-D camera was used to form an eye-to-hand visual measurement platform to collect experimental data for evaluation. To measure the tip pointing direction, we used an additional blue marker, such that the pointing direction can be approximately computed as  $\Theta_{T,mea}^W = \text{atan2}(\|\mathbf{x}_T^W - \mathbf{x}_{marker}^W\|_{XY}, \|\mathbf{x}_T^W - \mathbf{x}_{marker}^W\|_Z)$ , where  $\|\mathbf{x}_A - \mathbf{x}_B\|_*$  denotes the distance between  $A$  and  $B$  in the sub-scripted plane or axis expressed in  $\{W\}$ .

### A. Instantaneity and Synchronization

The performance in teleoperation tasks depends on a number of factors. Our hand motion does not provide a high input

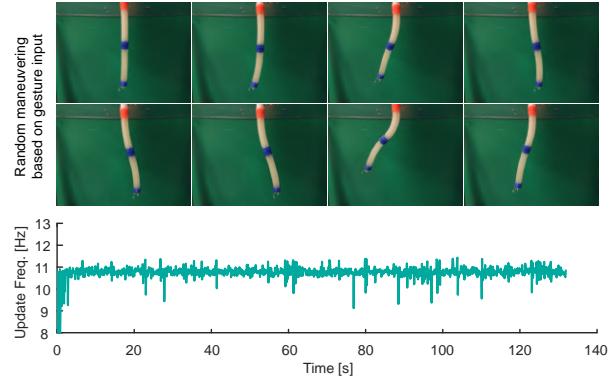


Fig. 9. Up: video snapshot of the robot’s 3-D motion based on real-time gesture input. The video was taken using a monocular camera. Down: update frequency for each while-loop of computation and motors actuation.

accuracy, but it can work together with our eyes or other sensors in a closed-loop way to alleviate the problem. Nevertheless, the latency between the input command and the robot action could lead to error or even failure in the teleoperation. In the experiment, the process of IK computation and robot actuation was updated at about 11 Hz during the manipulation, with about 0.09-second delay between hand motion and robot motion, enabling an MF control in highly responsive synchronization. Based on random steering (including the gesture shortcuts and different hand speeds) for about 130 seconds, we plotted Fig. 9 showing the update frequency of the while-loop, which included the IK computation and motors actuation. The result demonstrates a stable steering performance with rapid response, which is one of the most critical factors for an MF robotic system.

### B. Accuracy: Simulator v. Prototype

We first examined tip positioning accuracy and hand signal filtering effectiveness by maneuvering the robot tip to (i) the desired spatial position for an 8-second stay and (ii) the desired path as in the simulator. Hence, the error terms for evaluation could be defined as:

$$\begin{cases} e_{\mathbf{x},exp} = \|\mathbf{x}_{T,sim}^W - \mathbf{x}_{T,mea}^W\| \\ e_{\Theta,exp} = \|\Theta_{T,sim}^W - \Theta_{T,mea}^W\| \end{cases} . \quad (28)$$

The experiment results of (i) and (ii) are shown in Fig. 10(a-b) and Fig. 10(c-d), respectively. The results indicate that the

<sup>1</sup>In the sense of rotating our wrist and switching gestures at the same time.

TABLE II  
ERRORS BETWEEN SIMULATION AND EXPERIMENT OF THE POIS ON A 100-MM LENGTH ROBOT (UNIT: MM)

Position of POI w.r.t. $\{W\}$		$\{B\}$			$\{M\}$			$\{T\}$		
Axis <sup>†</sup>		$X$	$Y$	$Z$	$X$	$Y$	$Z$	$X$	$Y$	$Z$
Config. 0	sim.	0	0	50	0	0	100	0	0	150
	exp.	1.769	0	49.97	1.436	0	100.3	0.102	0	150.249
	err.	-1.769	0	0.03	-1.436	0	-0.3	-0.102	0	-0.249
<b>err. norm</b>		1.769			1.467			0.269		
Config. 1	sim.	0	0	48	-8.639	-3.296	97.56	7.475	-2.036	146.6
	exp.	1.674	0	46.38	-6.843	-5	97	7.401	-2	147.3
	err.	-1.674	0	1.62	-1.796	1.704	0.56	0.0740	-0.036	-0.7
<b>err. norm</b>		2.330			2.538			0.705		
Config. 2	sim.	0	0	44.36	6.467	-3.478	93.98	8.355	-4.397	143.9
	exp.	1.775	-1	44.19	6.042	-2	92.7	9.157	-5	143.5
	err.	-1.775	1	0.17	0.425	-1.478	1.28	-0.802	-0.603	0.4
<b>err. norm</b>		2.044			2.001			1.080		
Config. 3	sim.	0	0	40.19	-1.332	-4.908	90.2	6.280	-6.621	139.5
	exp.	1.697	-1	41.38	-2.127	-3	91.4	5.982	-6	139.4
	err.	-1.697	1	-1.19	0.795	-1.908	-1.2	0.298	-0.621	0.1
<b>err. norm</b>		2.301			2.390			0.696		

<sup>†</sup>The  $Y$  in exp. denotes the depth measurement from the RGB-D camera, with a resolution of 1 mm.

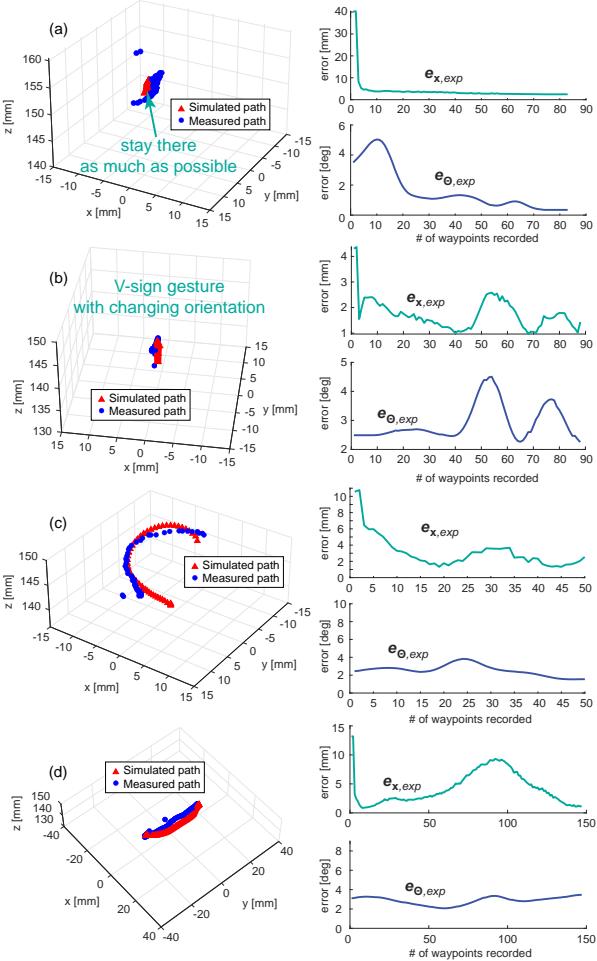


Fig. 10. (a-b) Control the robot tip to reach an arbitrary position within the workspace for a hovering. This result reflects the performance of input filtering, as our hands should be trembling in the mid-air without any support. (a) palm gesture: general control; (b) V-sign gesture: constrained motion control where the tip should be moved at  $[0, 0, Z]^\top$ . (c-d) Maneuver the robot tip in 3-D space. The simulated path and the measured path are close, with a steady-state norm positioning error at about 4 mm, and orientation (pointing direction) error within 5 degrees.

real-time pose of the robot tip being visually measured could satisfactorily align with the simulator's prediction. Therefore,

the robot tip can be well-maneuvered as desired. In the steady-state, the norm positioning errors can be within 4 mm (while in the reachable workspace), and the pointing direction error is within 5 degrees. Robot modeling and prototype fabrication should be made accountable for the errors. Since the simulator is developed based on a particular soft robot model, the real prototype may not precisely reveal the model's prediction. Besides, the fabrication and robot assembly defects cannot be eliminated for our current prototype.

We have evaluated the predicted manipulator's motion accuracy in free space. In order to do that, we measured the positional errors between the simulation and experiments of the POIs, namely,  $\{B\}$ ,  $\{M\}$ , and  $\{T\}$ , with the concerning results shown in Table II. Different static configurations were posed based on the specific gestures, as shown in Fig. 11. The results from Table II indicate that the deviations are within 2 mm on each axis. Various factors, such as gesture sensing (master side), prototype installation (follower side), and 3-D measurement (evaluation side), may contribute to the errors. It is concluded that our prototype robot could accurately reproduce the predicted motion from the simulator.

With that being said, an MF system is not necessarily as highly accurate as the simulator because the user could adjust gestures *in situ* based on his/her perception. Whenever the soft robot interacts with the environment, without stiffness compensation, there is a high chance that the robot would passively comply with the surroundings. Therefore, the effectiveness of real-world manipulation shall be examined.

### C. Maneuverability

We can interactively control the soft robot for real-world manipulation based on satisfactory synchronization and reliable accuracy in free space. In this subsection, we demonstrate the maneuverability by steering the robot to conduct some light-weighted interactive tasks.

As shown in Fig. 12, some loosely installed LEDs were used. The mission was to either stiffen or loosen the LEDs by touching them via the robot tip. Steered by the gesture motion, we could efficiently drive the robot tip to touch the desired LEDs. When the robot was in touch with the external object, it may not go as the prediction developed based on

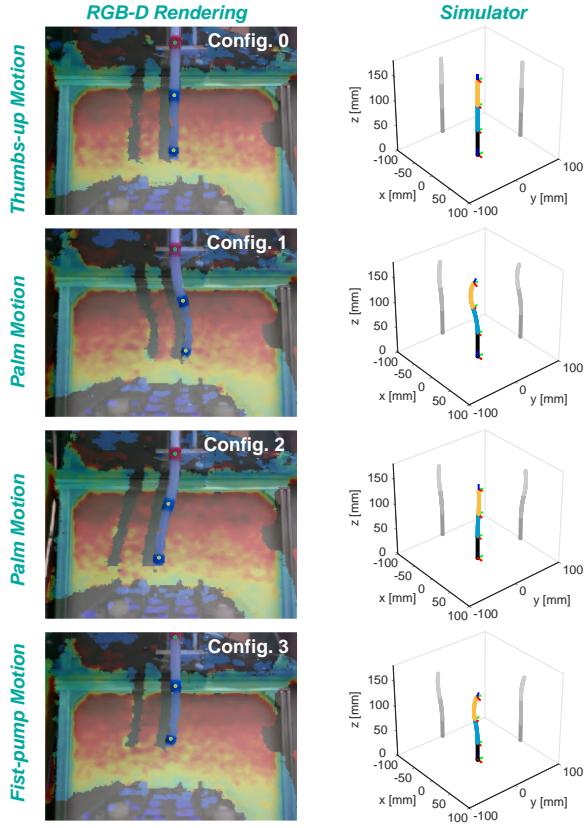


Fig. 11. Different robot configurations (poses) for Table II.

the free-space IK. However, the user can still slightly fine-tune the gesture pose by intuition for robot adjustment. The demonstration showcases the potential use of redundant soft robots to manipulate fragile objects remotely using a single gesture recognition-based input device.

Since our robot was intended to perform tasks through teleoperation, we conducted a series of tests to simulate using soft robots for general pick-and-place tasks. As shown in Fig. 13, a metal hook was attached to the robot tip, which was then used to pick up a small object on the uneven and sticky tissue phantom. The picking process involved (i) the approaching of the tip to the object, (ii) the alignment of the hook and the object, (iii) the action that lifts the object, which was stuck on the phantom surface, and (iv) dropping. Completing such a process through a mouse or joystick could be challenging, as it requires synchronized position and orientation control in order to successfully insert the hook and pass through the object. Also, the proximal segment is temporally locked to facilitate object pickup with the distal segment, after which the robot restores to full capability during approach and drop-off. Here, multi-mode steering inputs are required to complete the task, and the result shows that a novice examiner can successfully finish the picking. For these load-carrying trials, the robot was operating at a slow speed and the quasi-static approximation remains valid without observing large oscillation. Despite the inevitable discrepancy, users can still complete the task through monitoring by providing additional motion input through the steering to compensate for the error. The results show that the steering framework can be used in real-world teleoperation without employing articulated input

devices, which is ideal for virtual reality (VR)-assisted tasks.

#### D. Obstacle Avoidance Steering in Structured Environments

Unlike the traditional haptic devices that may provide force feedback to the human hand, the LMC only reads. Therefore, our method cannot inform users of any robot-obstacle collision cases. To compensate for this drawback, we propose a method to reduce the chance of robot-obstacle collision. The robot model and the task constraints (obstacles) will first be rendered in the simulator, and the LMC inputs will execute the robot in the virtual world to ensure collision-free configurations before transferring the motion back to the actual robot.

Assuming that a sphere obstructed the robot body when the robot was trying to move in a straight line, information such as the size and the location could be evaluated through sensors (e.g., camera) and updated accordingly in the virtual world. The robot configuration that can follow the input path as closely as possible while avoiding collision can be computed by making use of the null space motion. We further implemented this principle in the real-time teleoperation based on the obstacle-avoidance optimization method [19] that continuously maximize the damped shortest distance between the obstacle surface  $O$  and robot's circumference  $S(\Delta q)$ , given the radius of obstacle  $R_{obs}$  and robot  $R_{sr}$  in a structured environment:

$$\arg \min_{\Delta q} \underbrace{\| \dots \|_2^2}_{\text{position}} + \eta \underbrace{\left\| \frac{R_{obs} + R_{sr}}{\min \| S(\Delta q) - O \|_2} \right\|_2^2}_{\text{obstacle avoidance}} + \lambda^2 \underbrace{\| \dots \|_2^2}_{\text{least motion}} . \quad (29)$$

As shown in Fig. 14, the LMC detects our hand motions and commands the robot to move in a straight line without the user paying attention to the obstacle (sphere) in front of the robot located at (0 0 120) mm w.r.t.  $\{W\}$  with a radius of 10 mm. Regardless of any hand motion that could have resulted in a collision, the soft robot can adjust its body to finish the primary tip-following task as much as possible. Experiment results indicate that the collision avoidance motion would have limited effects on the tip positioning accuracy when encountering an obstacle. The position deviations along the  $X$  and  $Y$  axes are within 2 mm, and the maximum deviations along the  $Z$  axis are within 4 mm. By increasing the damping parameter  $\eta$ , one can reduce the chance of robot-obstacle collision but at the cost of trading off the tip following accuracy. The simulation output can be further utilized in the physical robot. This application could be helpful for soft robot teleoperation in a structured environment, as users can ignore the obstructed scene and focus more on the manipulation task.

## VI. CONCLUSION

In this work, we proposed a gesture-based steering framework for redundant soft robots. Our teleportation method merits intuitiveness, instantaneity, and cost-effectiveness. Users can directly assign the palm pose as the definition of the robot tip pose for real-time maneuvering. The KF and LPF-SLERP were employed for hand motion estimation and filtering. Several shortcut gestures were provided for the robot's constrained motion. The IK can be rapidly computed based on the optimization algorithm, which is fast enough for real-time performance. We contributed an interactive LMC-virtual robot simulator that can be used to verify our idea.

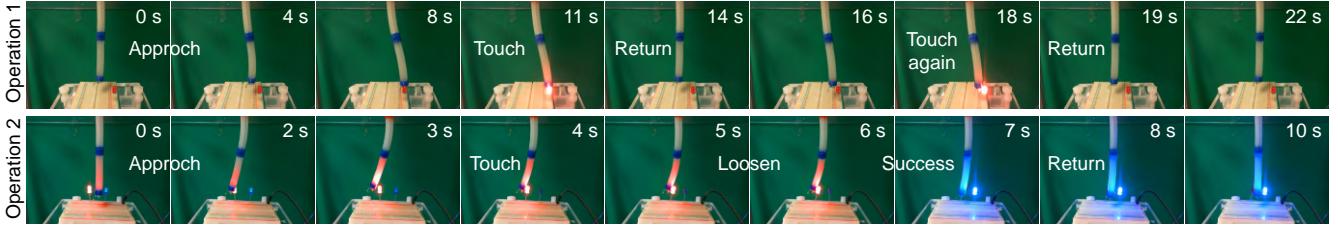


Fig. 12. Real-time MF manipulation using gesture. **Operation 1:** The red LED was loosely installed on a breadboard. We steered the soft robot to touch the LED and make it glow twice. **Operation 2:** The red and blue LEDs were connected in parallel and only the red would work (due to the different forward bias voltage). We steered the robot to loosen the red LED and made the blue one glow instead (Captured from the supplementary video S2).

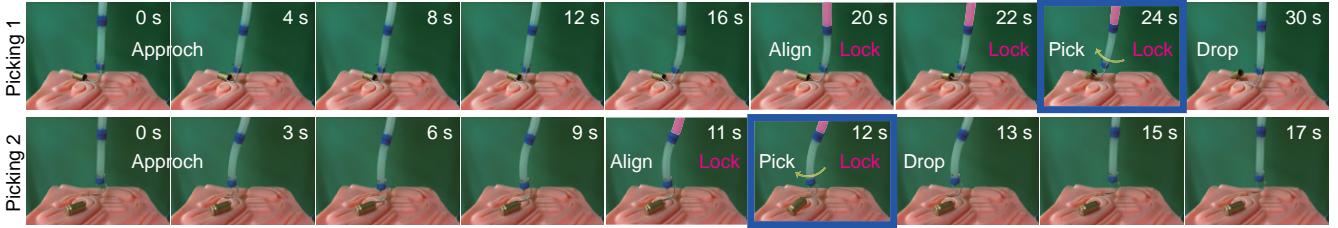


Fig. 13. Real-time master-follower manipulation using different gesture modes: picking up a 9-mm bullet case (weighs about 4 grams, which is 80% of the soft robot's mass) in different locations on an uneven and sticky tissue phantom. (Images were captured from the supplementary video S1).

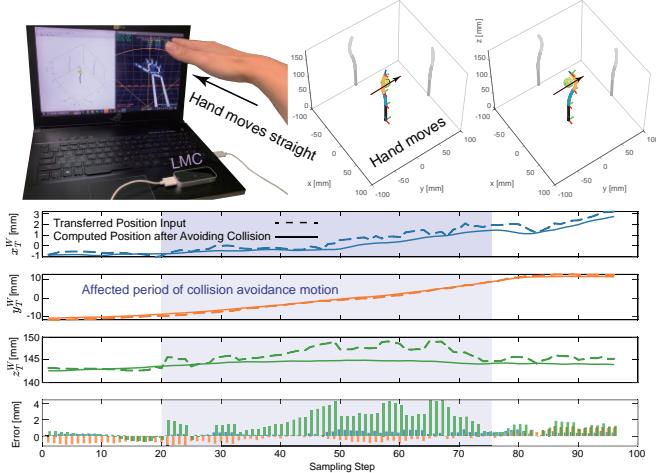


Fig. 14. Maneuvering a soft robot in the simulator with the hand moving straight – a move that could have caused a collision between the robot and the known obstacle. With the obstacle-avoidance mode on, the chance of collision can be minimized in run time due to the utilization of null space motion optimization. The collision avoidance motion affects the steering accuracy within 4 mm in  $z_T^W$ , and within 2 mm in  $x_T^W$  and  $y_T^W$ .

Prototypical experiments have also been conducted to verify the effectiveness of our method and the control accuracy of the system. Manipulations that simulate the dedicated tasks were performed, which further support the framework's feasibility.

We acknowledge the limitations, such as the limited sensing accuracy of the LMC device and the lack of haptic feedback for the operators. However, the actual accuracy can be improved by incorporating sensory feedback into the controller.

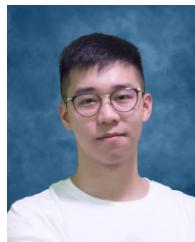
In terms of application, we envision somatic teleoperation on soft robots with more motion-sensing devices (like using the front camera of a cellphone with gesture detection algorithms). This will make robot teleoperation more accessible to a wider audience. In addition, the framework can possibly be used in VR/AR-related operations where the operators' vision is immersed in the virtual scenes. As one of the first works of a somatic MF-controlled soft robot system, we believe that the

proposed framework will inspire more practical applications of other soft robots by making them teleoperable without excessive upgrades. Future work might be done to control the manipulator's stiffness for heavy object manipulation.

## REFERENCES

- [1] M. Cianchetti, C. Laschi, A. Menciassi, and P. Dario, "Biomedical applications of soft robotics," *Nat. Rev. Mater.*, vol. 3, no. 6, pp. 143–153, 2018.
- [2] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum robots for medical applications: A survey," *IEEE Trans. Robot.*, vol. 31, no. 6, pp. 1261–1280, 2015.
- [3] J. Lai, K. Huang, B. Lu, Q. Zhao, and H. K. Chu, "Verticalized-tip trajectory tracking of a 3d-printable soft continuum soft robot: Enabling surgical blood suction automation," *IEEE/ASME Trans. Mechatron.*, vol. 27, no. 3, pp. 1545–1556, 2022.
- [4] Z. Xie, B. Chen, J. Liu, F. Yuan, Z. Shao, H. Yang, A. G. Domel, J. Zhang, and L. Wen, "A tapered soft robotic oropharyngeal swab for throat testing: A new way to collect sputa samples," *IEEE Robot. Autom. Mag.*, vol. 28, no. 1, pp. 90–100, 2021.
- [5] C. Li, Y. Yan, X. Xiao, X. Gu, H. Gao, X. Duan, X. Zuo, Y. Li, and H. Ren, "A miniature manipulator with variable stiffness towards minimally invasive transluminal endoscopic surgery," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 5541–5548, 2021.
- [6] R. J. Webster III and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *Int. J. Robot. Res.*, vol. 29, no. 13, pp. 1661–1683, 2010.
- [7] Ultraleap. Leap motion controller. [Online]. Available: [https://www.ultraleap.com/datasheets/Leap\\_Motion\\_Controller\\_Datasheet.pdf](https://www.ultraleap.com/datasheets/Leap_Motion_Controller_Datasheet.pdf)
- [8] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, "Analysis of the accuracy and robustness of the leap motion controller," *Sensors*, vol. 13, no. 5, pp. 6380–6393, 2013.
- [9] G. Du, M. Chen, C. Liu, B. Zhang, and P. Zhang, "Online robot teaching with natural human–robot interaction," *IEEE Trans. Ind. Electron.*, vol. 65, no. 12, pp. 9571–9581, 2018.
- [10] Y. Liu and Y. Zhang, "Toward welding robot with human knowledge: A remotely-controlled approach," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 769–774, 2014.
- [11] D. Kruse, J. T. Wen, and R. J. Radke, "A sensor-based dual-arm tele-robotic system," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 1, pp. 4–18, 2014.
- [12] C. Meeker, M. Haas-Heger, and M. Ciocarlie, "A continuous teleoperation subspace with empirical and algorithmic mapping algorithms for nonanthropomorphic hands," *IEEE Trans. Autom. Sci. Eng.*, 2020.
- [13] E. Amanov, T.-D. Nguyen, and J. Burgner-Kahrs, "Tendon-driven continuum robots with extensible sections—a model-based evaluation of path-following motions," *Int. J. Robot. Res.*, vol. 40, no. 1, pp. 7–23, 2021.

- [14] R. K. Katzschnittmann, C. Della Santina, Y. Toshimitsu, A. Bicchi, and D. Rus, "Dynamic motion control of multi-segment soft robots using piecewise constant curvature matched with an augmented rigid body model," in *Proc. IEEE Int. Conf. Soft Robot.*, 2019, pp. 454–461.
- [15] K.-H. Lee, D. K. Fu, M. C. Leong, M. Chow, H.-C. Fu, K. Althoefer, K. Y. Sze, C.-K. Yeung, and K.-W. Kwok, "Nonparametric online learning control for soft continuum robot: An enabling technique for effective endoscopic navigation," *Soft Robot.*, vol. 4, no. 4, pp. 324–337, 2017.
- [16] H. Mo, B. Ouyang, L. Xing, D. Dong, Y. Liu, and D. Sun, "Automated 3-d deformation of a soft object using a continuum robot," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 4, pp. 2076–2086, 2020.
- [17] D. Bruder, B. Gillespie, C. D. Remy, and R. Vasudevan, "Modeling and control of soft robots using the koopman operator and model predictive control," *arXiv preprint arXiv:1902.02827*, 2019.
- [18] Y. Gao, K. Takagi, T. Kato, N. Shono, and N. Hata, "Continuum robot with follow-the-leader motion for endoscopic third ventriculostomy and tumor biopsy," *IEEE Trans. Biomed. Eng.*, vol. 67, no. 2, pp. 379–390, 2019.
- [19] J. Lai, B. Lu, Q. Zhao, and H. K. Chu, "Constrained motion planning of a cable-driven soft robot with compressible curvature modeling," *IEEE Robot. and Autom. Lett.*, vol. 7, no. 2, pp. 4813–4820, 2022.
- [20] J. Lai, T.-A. Ren, W. Yue, S. Su, J. Y. Chan, and H. Ren, "Sim-to-real transfer of soft robotic navigation strategies that learns from the virtual eye-in-hand vision," *IEEE Trans. Ind. Inform.*, vol. 20, no. 2, pp. 2365–2377, 2024.
- [21] T. B. Sheridan, "Human–robot interaction: status and challenges," *Hum. Factors*, vol. 58, no. 4, pp. 525–532, 2016.
- [22] J. Fraś, J. Czarnowski, M. Maciąś, J. Główka, M. Cianchetti, and A. Menciassi, "New stiff-flop module construction idea for improved actuation and sensing," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 2901–2906.
- [23] X. Wang, G. Fang, K. Wang, X. Xie, K.-H. Lee, J. D. Ho, W. L. Tang, J. Lam, and K.-W. Kwok, "Eye-in-hand visual servoing enhanced with sparse strain measurement for soft continuum robots," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2161–2168, 2020.
- [24] D. Jakes, Z. Ge, and L. Wu, "Model-less active compliance for continuum robots using recurrent neural networks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2019, pp. 2167–2173.
- [25] J. Lai, K. Huang, B. Lu, and H. K. Chu, "Toward vision-based adaptive configuring of a bidirectional two-segment soft continuum manipulator," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatron.*, 2020, pp. 934–939.
- [26] H. Wang, R. Zhang, W. Chen, X. Wang, and R. Pfeifer, "A cable-driven soft robot surgical system for cardiothoracic endoscopic surgery: preclinical tests in animals," *Surg. Endosc.*, vol. 31, no. 8, pp. 3152–3158, 2017.
- [27] H. Abidi, G. Gerboni, M. Brancadoro, J. Fras, A. Diodato, M. Cianchetti, H. Wurdemann, K. Althoefer, and A. Menciassi, "Highly dexterous 2-module soft robot for intra-organ navigation in minimally invasive surgery," *Int. J. Med. Robot. Comput. Assist. Surg.*, vol. 14, no. 1, p. e1875, 2018.
- [28] Y. Zhang, T. Wang, W. He, and S. Zhu, "Human-powered master controllers for reconfigurable fluidic soft robots," *Soft Robot.*, 2023.
- [29] H.-S. Yoon, J. H. Jeong, and B.-J. Yi, "Image-guided dual master–slave robotic system for maxillary sinus surgery," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1098–1111, 2018.
- [30] B. Ouyang, Y. Liu, H.-Y. Tam, and D. Sun, "Design of an interactive control system for a multisection continuum robot," *IEEE/ASME Trans. Mechatron.*, vol. 23, no. 5, pp. 2379–2389, 2018.
- [31] M. Hwang and D.-S. Kwon, "K-flex: a flexible robotic platform for scar-free endoscopic surgery," *Int. J. Med. Robot. Comput. Assist. Surg.*, vol. 16, no. 2, p. e2078, 2020.
- [32] C. G. Frazelle, A. Kapadia, and I. Walker, "Developing a kinematically similar master device for extensible continuum robot manipulators," *J. Mech. Robot.*, vol. 10, no. 2, 2018.
- [33] S. Furukawa, S. Wakimoto, T. Kanda, and H. Hagiwara, "A soft master-slave robot mimicking octopus arm structure using thin artificial muscles and wire encoders," *Actuators*, vol. 8, no. 2, p. 40, 2019.
- [34] Y. Yamamoto, S. Wakimoto, T. Kanda, and D. Yamaguchi, "A soft robot arm with flexible sensors for master–slave operation," *Eng. Proc.*, vol. 10, no. 1, p. 34, 2021.
- [35] H. Ham, M. Park, T. Park, X. Gao, Y.-L. Park, and M. J. Park, "Teleoperation of soft robots with real-time fingertip haptic feedback using small batteries," *Adv. Mater. Technol.*, p. 2300070, 2023.
- [36] G. Du, P. Zhang, and X. Liu, "Markerless human–manipulator interface using leap motion with interval kalman filter and improved particle filter," *IEEE Trans. Ind. Inform.*, vol. 12, no. 2, pp. 694–704, 2016.
- [37] G. Welch, G. Bishop *et al.*, "An introduction to the kalman filter," 1995.
- [38] J. Sola, "Quaternion kinematics for the error-state kalman filter," *arXiv preprint arXiv:1711.02508*, 2017.
- [39] I. Cvišić, J. Česić, I. Marković, and I. Petrović, "Soft-slam: Computationally efficient stereo visual simultaneous localization and mapping for autonomous unmanned aerial vehicles," *J. Field Robot.*, vol. 35, no. 4, pp. 578–595, 2018.
- [40] R. Weitschat, A. Dietrich, and J. Vogel, "Online motion generation for mirroring human arm motion," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 4245–4250.
- [41] B. Siciliano, L. Sciavicco, G. Oriolo, and L. Villani, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.
- [42] J. Perry. Matleap. [Online]. Available: <https://github.com/jeffsp/matleap>
- [43] W. Ding, J. Wang, C. Rizos, and D. Kinlyside, "Improving adaptive kalman estimation in gps/ins integration," *J. Navig.*, vol. 60, no. 3, pp. 517–529, 2007.



**Jiewen Lai** (S'20, M'22) received his Ph.D. degree in mechanical engineering from The Hong Kong Polytechnic University, Hong Kong, in 2022. From 2022 to 2023, he was a Postdoctoral Fellow at The Chinese University of Hong Kong (CUHK). In 2023, he joined the Department of Electronic Engineering at CUHK as a Research Assistant Professor.

His research interests include soft robotics, medical mechatronics, robophysics, and robot intelligence.



**Bo Lu** received his Ph.D. degree in mechanical engineering from The Hong Kong Polytechnic University, Hong Kong, in 2019. From 2019 to 2021, he was a Postdoctoral Fellow with the CUHK T Stone Robotics Institute and the Department of Mechanical and Automation Engineering at The Chinese University of Hong Kong, Hong Kong. He is currently an Associate Professor at the School of Mechanical and Electrical Engineering, Robotics and Microsystems Center, Soochow University, China.

His current research interests include medical robotics, computer vision, and surgical automation techniques.



**Kaicheng Huang** received his Ph.D. degree in mechanical engineering from The Hong Kong Polytechnic University, Hong Kong, in 2020. From 2020 to 2023, he was a Postdoctoral Fellow with The University of Hong Kong, Hong Kong, and the Southern University of Science and Technology (SUSTech), Shenzhen, China. He is now a Senior Research Fellow with the Department of Materials Science and Engineering, SUSTech.

His research interests include automated cell patterning, robotic control, and machine learning.



**Henry K. Chu** (M'12) received the B.S. degree in mechanical engineering from the University of Waterloo, Waterloo, ON, Canada, in 2005, and the M.A.Sc. and Ph.D. degrees from the University of Toronto, Toronto, ON, Canada, both in mechanical and industrial engineering, in 2007 and 2011, respectively. He was a Postdoctoral Fellow at the University of Toronto and the City University of Hong Kong. He is currently an Associate Professor at The Hong Kong Polytechnic University, Hong Kong.

His research interests include robotic manipulation, vision-based control and automation, micro-system design, and tissue engineering.