# Learning-Based Visual-Strain Fusion for Eye-in-Hand Continuum Robot Pose Estimation and Control

Xiaomei Wang, *Member, IEEE*, Jing Dai, *Student Member, IEEE*, Hon-Sing Tong, Kui Wang, Ge Fang, Xiaochen Xie, *Member, IEEE*, Yun-Hui Liu, *Fellow, IEEE*, Kwok Wai Samuel Au, and Ka-Wai Kwok, *Senior Member, IEEE*

***Abstract*—Image processing has significantly extended the practical value of the eye-in-hand camera, enabling and promoting its applications for quantitative measurement. However, fully vision-based pose estimation methods sometimes encounter difficulties in handling cases with deficient features. In this article, we fuse visual information with the sparse strain data collected from a single-core fiber inscribed with fiber Bragg gratings (FBGs) to facilitate continuum robot pose estimation. An improved extreme learning machine algorithm with selective training data updates is implemented to establish and refine the FBG-empowered (F-emp) pose estimator *online*. The integration of F-emp pose estimation can improve sensing robustness by reducing the number of times that visual tracking is lost given moving visual obstacles and varying lighting. In particular, this integration solves pose estimation failures under full occlusion of the tracked features or complete darkness. Utilizing the fused pose feedback, a hybrid controller incorporating kinematics and data-driven algorithms is proposed to accomplish fast convergence with high accuracy. The online-learning error compensator can improve the target tracking performance with a 52.3%–90.1% error reduction compared with constant-curvature model-based control, without requiring fine model-parameter tuning and prior data acquisition.**

***Index Terms*—Camera pose estimation, fiber Bragg grating (FBG), hybrid control, online learning, visual-strain fusion.**

Manuscript received 11 December 2022; accepted 24 December 2022. This work was supported in part by the Research Grants Council of Hong Kong under Grant 17205919, Grant 17207020, Grant 17209021, and Grant T42-409/18-R, in part by Innovation and Technology Commission under Grant MRP/029/20X, and in part by the Multi-Scale Medical Robotics Center Limited, InnoHK. This article was recommended for publication by Associate Editor Guoying Gu and Editor E. Yoshida upon evaluation of the reviewers' comments. *(Corresponding author: Ka-Wai Kwok.)*

Xiaomei Wang is with the Department of Mechanical Engineering, The University of Hong Kong, Hong Kong, and also with the Multi-Scale Medical Robotics Center Limited, Hong Kong (e-mail: wangxmei@connect.hku.hk).

Jing Dai, Hon-Sing Tong, Kui Wang, Ge Fang, and Ka-Wai Kwok are with the Department of Mechanical Engineering, The University of Hong Kong, Hong Kong (e-mail: u3006581@connect.hku.hk; u3527192@connect.hku.hk; kuiwang@connect.hku.hk; fangge@hku.hk; kwokkw@hku.hk).

Xiaochen Xie is with the Department of Automation, Harbin Institute of Technology, Shenzhen, Shenzhen 518055, China, and also with the Guangdong Key Laboratory of Intelligent Morphing Mechanisms and Adaptive Robotics, Shenzhen 518055, China (e-mail: xiexiaochen@hit.edu.cn).

Yun-Hui Liu is with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: yhliu@mae.cuhk.edu.hk).

Kwok Wai Samuel Au is with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong, and also with the Multi-Scale Medical Robotics Center Limited, Hong Kong (e-mail: samuelau@cuhk.edu.hk).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TRO.2023.3240556.

Digital Object Identifier 10.1109/TRO.2023.3240556

## I. Introduction

RECENT advances in computer vision enable the detection of the robot configuration in unstructured environments [1], [2], similar to human visual perception that allows us to interpret body movement relative to our surroundings. In computer vision, camera pose estimation is a fundamental problem that has been widely studied in the areas of structure from motion (SfM), visual odometry [3], simultaneous localization and mapping (SLAM), and even commonly applied in augmented reality [4] as well as autonomous navigation [5]. Pose estimation by means of temporally coherent features in a sequence of two-dimensional/three-dimensional (2-D/3-D) images [6] can avoid the complicated integration of additional positional sensors.

However, feature-based estimation using cameras is inherently subject to the image quality, which is inevitably affected by unstable light exposure, vision occlusion, and rapid viewpoint changes [3]. This weakness is made more apparent with cameras used in the eye-in-hand configuration, where the camera (i.e., the *eye*) is fixed on the robot end-effector (the *hand*) to see its surroundings. Although the eye-in-hand approach is intuitive and provides active visual perception, it requires effective end-effector movement for pose detection [7], [8] and greatly demands for consistent robot motion patterns. In addition, as the camera usually points closer toward the objects of interest [9], the effect of local lighting variations and specular reflection will be dominant in the camera view. To compensate for the pose error induced by the lack of high-quality image features, fusing computer vision data with other sensing feedback has become a promising option.

The most prevalent type of fusion approach is to integrate cameras with inertial measurement units (IMUs) [4], [10], that is, the visual-inertial system (VINS), which has been generally developed for rigid-link robots. Detected acceleration and angular velocity could be utilized by employing statistical filtering techniques, such as extended Kalman filters [4] or learning-based fusion methods, e.g., long short-term memory [11] and

convolutional neural networks [12]. Although the extrinsic calibration and accumulated drift in VINS were widely discussed [13], [14], [15], residual and nonquasi-static vibrations in soft robots would induce increased or accumulated positional errors much more than their rigid counterparts. Note that the IMUs accuracy and reliability are also bounded by its limited acceleration/velocity sensing range. Mechanical integration of IMUs would also require tailor-made or compact packing with the camera at the soft robot's tip, whereas rigid robots have the freedom to fix the IMUs anywhere along their links. To this end, there remains a demand for alternative sensors that can directly measure the pose of soft robots.

Soft robots usually involve relatively large deformation of which the strain changes on its body surface would give strong cues to estimate its configuration. Real-time strain sensing achieved with fiber Bragg grating (FBG) optical fiber is a potential candidate that can utilize these strain changes for feedback [16], [17], [18]. FBG sensors provide several advantages over electronic strain sensors, including the capability for dense strain measurements with a single connection and insusceptibility to water submersion and electromagnetic (EM) fields. As a result, FBGs have been investigated in thin surgical tools, such as biopsy needles [19] or even in magnetic resonance imaging environments [20], [21]. Continuous-grating multicore fiber with optical frequency-domain reflectometry (OFDR) interrogation is one form of FBG sensing that is capable of stand-alone 3-D curvature sensing with a single fiber and is typically integrated in manipulators or instruments with strict diameter requirements [22], [23]. For pose estimation of fluid-driven soft robots, single-core optic FBGs using the common wavelength division multiplexing method would be more appropriate considering its advantages of higher sensing sampling rate (100–3000 Hz) and significantly lower cost. When helically wound onto the robot surface [24], [25], the fiber can sensitively detect small deformations at high frequencies enabling reliable closed-loop robot control. Task space control of the soft robot using absolute FBG-detected strain would be more reliable than using IMU feedback, which needs to calculate the integral of relative acceleration/velocity.

The mapping from FBGs measurement to continuum robot configuration can be established using either analytical modeling [21] or machine learning [26] approaches. Sefati et al. [25] had compared their tip positional sensing accuracy of a planar bending continuum manipulator equipped with *three* parallel FBG fibers. The results demonstrated improved sensing performance using the data-driven method without prior information of the FBG allocation. In learning-based methods, positional markers need to be employed as the ground truth to complete the mapping. In our previous work [27], we also proposed a flexible surface sensing system in which only *one* single-core fiber inscribed with FBGs was embedded in a soft substrate. Offline learning was needed to "train" the mapping between FBG strains and the surface morphology detected by motion capture cameras.

Considering the small form factor of FBG fibers and their ease of integration with devices/instruments, researchers have also aimed to leverage them with various camera configurations. In other previous work, we employed a single-core FBG fiber on a continuum robot to enhance the 2-D motion estimation and path tracking in the endoscopic camera view [28]. However, these types of 3-D shape and 2-D motion estimators need to be trained by additional positional sensors in advance, heavily relying on prior data exploration and accurate ground truth data. Alambeigi et al. [29] also proposed a sensor fusion technique to address the shape/position estimation of continuum robots. As an illustration, the intermittent external information provided by an eye-to-hand camera calibrated the continuous imperfect FBG feedback to achieve the accurate 2-D positional sensing in obstructed environments. This work is one example of the few visual-strain fusion combinations for positional sensing, with even fewer examples using cameras integrated into the robot tip for eye-in-hand feedback.

Therefore, our concern in this article is to utilize a self-contained camera to serve as the pose ground truth in ordinary cases, while the online initialized and updated FBG sensor can be fused to settle estimation error caused by poor-quality images. No external sensors would be applied to the algorithm since we would like to simplify the employed devices. A widely adopted sensor may be used in the test but just to prove the accuracy of camera-based pose estimation as the training ground truth. The sensing dimension is also extended from 3-D position/shape to 6-D pose, offering more flexibility in robot applications, such as spatial image stitching.

With real-time learning-based sensing feedback available, the design of the continuum robot controller can also leverage the advantages of data-driven refinement [30], [31], [32]. Our previous work had utilized online-learning locally weighted projection regression and Gaussian process regression (GPR) for orientation control [33] and visual servoing control [34], respectively. Such pure data-driven control has encouraging potential in soft robots but is subject to time-consuming data-exploration procedures. Although analytical kinematics modeling encounters challenges in parameter characterization due to nonlinear fluid and elastomer's dynamics, the convergence of analytical solutions can usually be guaranteed as it is calculated from the inverse kinematics mapping. The combination of kinematics model-based and learning-based approaches could leverage both of their respective advantages. The constant-curvature (CC) assumption can be used to establish a rough kinematics model, saving the time for data collection. Once the robot starts manipulation, the online data exploration could be activated, with which an error compensator is learned/updated to reduce the positional error induced by the model-based control.

In this article, we aim at accurate eye-in-hand pose estimation of soft robots, which is realized by sensing fusion of an integrated single-core FBG fiber and a monocular camera. The fusion result can be further used as feedback for position control. The proposed framework is depicted in Fig. 1. The major contributions of this work are summarized as follows:

1) online-learning-based pose estimation using sparse strain measurement of single-core FBG fiber and sensing fusion with monocamera SLAM;
2) hybrid control combining model-based and data-driven methods for accurate position tracking using soft robots,

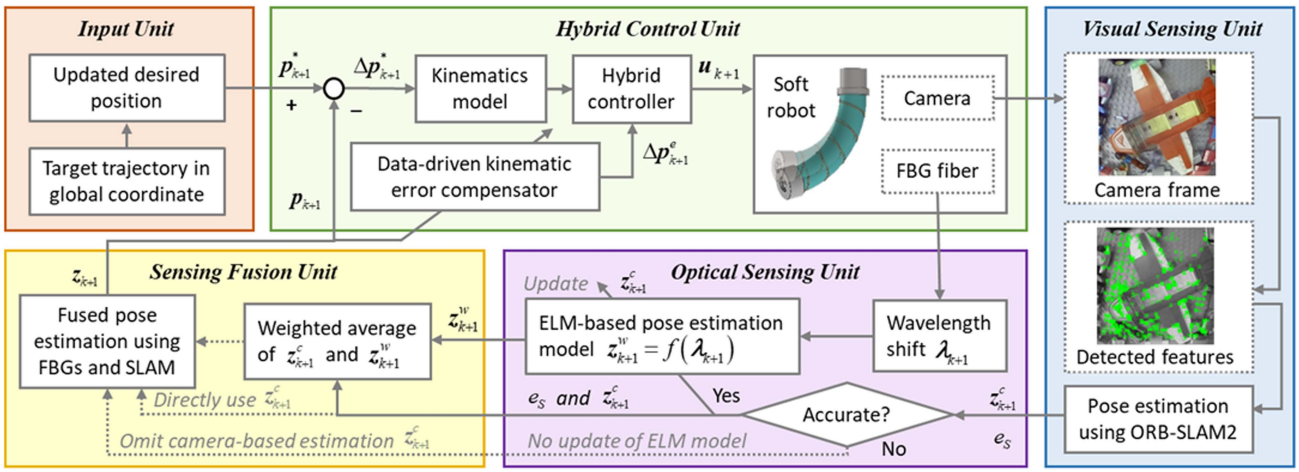


Fig. 1. Robot control architecture. Hybrid controller combining kinematics model and data-driven-trained compensator is implemented, with the pose feedback obtained by sensing fusion of FBG strain measurement and visual feedback. Monocamera ORB-SLAM2 is used, which serves as the ground truth to initialize and update the ELM-based model using FBG strain data. The FBG-estimated portion in the sensing fusion will be more heavily weighted in visual feature-deficient scenarios.

*without* the need for precise parameter tuning and prior data collection;

3) Experimental validation of the proposed sensing fusion modality under poor visual conditions and validation of the robust hybrid controller via target tracking tasks.

The rest of this article is organized as follows. Section II focuses on the pose estimation aspect, which acts as the sensing feedback in the robotic system. It briefly introduces the state-of-the-art camera-based pose estimation method Oriented FAST and Rotated BRIEF (ORB) SLAM2, after which our proposed online-training FBG-empowered (F-emp) pose estimator by extreme learning machine (ELM) and the visual-strain fusion scheme are explained in detail. Section III presents the hybrid kinematics controller comprising the CC-based model and GPR-based error compensator. Experimental validation for both sensing and control is summarized in Section IV, including comparisons between fusion-based and camera-based pose estimations in handling visual obstacles, as well as comparisons between controllers using only a CC model versus combining the model with an error compensator. In addition, the overall performance of path following by integrating our proposed visual-strain fusion sensing and hybrid controller is demonstrated. The effect of physical contacts on our system is also investigated. Finally, Section V concludes this article.

## II. Pose Estimation of Soft Manipulator

To improve the eye-in-hand pose estimation stability by integrating a single-core FBG fiber, the FBGs can be evenly distributed on the robot body. In our case, the fiber is helically wrapped on the cylindrical surface, thus reflecting the robot's overall deformation [see Fig. 2(a)] via wavelength shifts [see Fig. 2(b)].

We hypothesize that the camera-based estimation in feature-abundant scenarios is the primary choice of sensing information

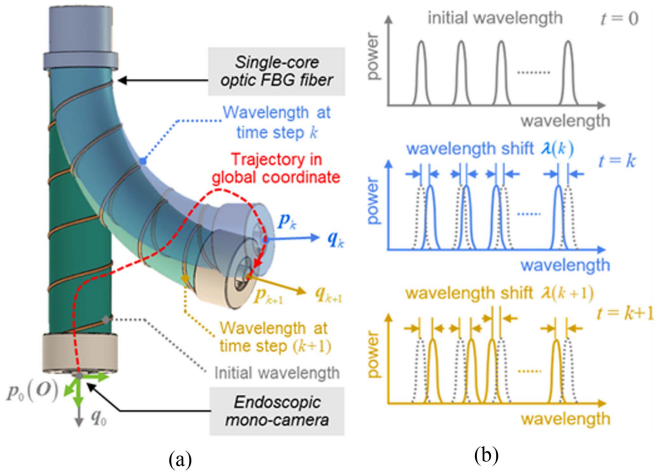


Fig. 2. Soft continuum robot wrapped with a helically wound single-core FBG fiber. (a) Camera poses obtained at each time step $k$ based on the SLAM algorithm. (b) FBG wavelengths shifted correspondingly, i.e., from $\lambda(k)$ to $\lambda(k+1)$.

but may suffer from inadvertent poor image quality. For such circumstances, the F-emp pose estimation model, trained by accurate camera-based estimations, could act as a stable backup and guarantee the operation of the entire framework.

### A. Task Space Definition

The eye-in-hand camera is fixed on the robot end-effector, therefore sharing the same pose with the robot tip. There have been various approaches utilizing visual feedback to analyze the camera pose. We take the initial position of the end-effector without robot actuation as the origin of a global (measuring) coordinate system (green coordinate frame in Fig. 2), and the robot central axis as the $z$-axis with downward as the positive

direction. In the SLAM algorithm, the initial pose is taken by default as the origin of its measuring frame. The task space related to the end-effector position is defined in the 3-D global frame. The camera pose estimated by SLAM at time step $k$ is defined as $\boldsymbol{z}_c = \begin{bmatrix}\boldsymbol{p}_c(k) & \boldsymbol{q}_c(k)\end{bmatrix}$, including the position $\boldsymbol{p}_c(k) \in \mathbb{R}^3$ and quaternion-represented orientation $\boldsymbol{q}_c(k) \in \mathbb{R}^4$. The actual pose becomes $\boldsymbol{z} = \begin{bmatrix}\boldsymbol{p}(k) & \boldsymbol{q}(k)\end{bmatrix}$. Under stable and smooth movement, the SLAM estimation $\boldsymbol{z}_c$ in feature-abundant camera views can be considered as the approximation of robot end-effector pose, i.e., $\boldsymbol{z}_c \approx \boldsymbol{z}$. It is worth noting that SLAM would not be the only option to provide $\boldsymbol{z}_c$; the feedback from other pose measuring approaches will also be valid to train the FBG estimation model, such as EM tracking. Here, the reason to adopt the SLAM-based pose is to utilize the integrated camera without the need for any external sensing devices. The actuator input is represented as $\boldsymbol{u}(k) \in \mathbb{R}^m$ (at equilibrium state), where $m$ denotes the dimension of actuation space. The control objective is to generate an actuation command $\Delta\boldsymbol{u}(k)$, achieving the desired movement $\Delta\boldsymbol{p}^*(k)$ or $\Delta\boldsymbol{q}^*(k)$. The single-core FBG fiber is helically wrapped along with the continuum robot. The multiplexing $l$ units of FBGs inscribed are independent of each other, providing the corresponding $l$ wavelength/strain measurement points. Wavelength shift vector $\boldsymbol{\lambda}(k) \in \mathbb{R}^l$ depicts the difference between wavelength vector at time step $k$ and the original wavelength vector $\boldsymbol{\lambda}_0$ corresponding to the initial robot configuration [see Fig. 1(b)].

### B. Camera Pose Estimation Via ORB-SLAM2

ORB-SLAM2 has three common modules: tracking, local mapping, and loop closing [35]. The camera pose can be obtained at each input image frame by building a perspective-n-point model through the tracking thread. After ORB features in the image are extracted, pose estimation can be conducted by matching features in two consecutive frames and refined by minimizing the reprojection error with motion-only bundle adjustment optimization. This reprojection error, represented as $e_S$ here, is defined as the Euclidean distance between the image projection of 3-D map points and the corresponding observed feature in the image plane. That is to say, the precision of $e_S$ represents the matching accuracy of feature correspondences and then the quality of the pose estimation. The monocular camera was calibrated with OpenCV via robot operating system. It should be noticed that the pose estimation result using SLAM has a different measurement scale than that of common positional sensors. An affine transformation on the raw measurement would be needed to proceed with its usage in robot control. We utilized the EM tracker to calibrate the scaling parameter in this transformation in advance such that the online training of F-emp model and sensor fusion is both in the metric scale.

### C. Learning-Based Pose Estimation Using FBGs

The consideration of optic fiber integration is that the wavelength shift/strain sequence of all FBGs should be mapped uniquely to reflect the end-effector pose but not altering the original soft robot mechanical properties (see Fig. 3). Details about the fiber placement can be found in our previous work [28].

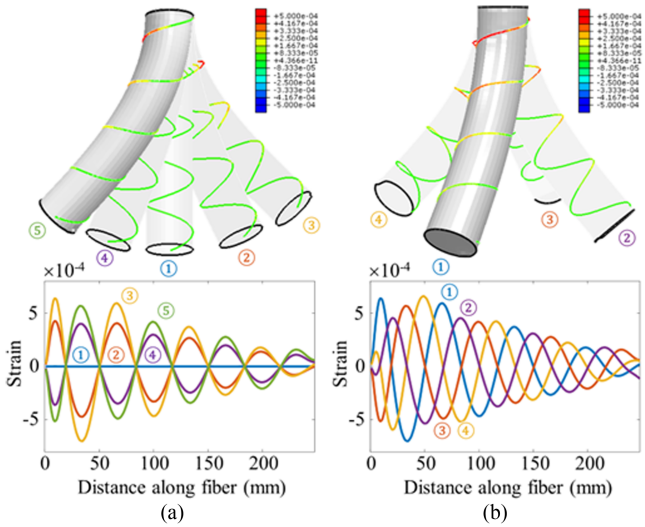


Fig. 3. Finite-element modeling of the strains helically distributed along an elastic continuum manipulator. (a) Strains varying in amplitude when the manipulator bends on the same plane/direction. (b) Strains under four different bending directions distinguished by their phase differences.

Simply, the fiber was wound helically on the robot body (see Fig. 2). No rigorous requirements on the wrapping structure were set as long as the FBGs can be dispersedly distributed. Distances between adjacent turns could also vary without strict consistency. The details of the fiber used in our experiments can be found in Section IV-A. With an appropriate sampling rate of pose estimation provided by the SLAM algorithm (i.e., 20–50 Hz, related to the camera and computer performance), both the wavelength shift $\boldsymbol{\lambda}(k)$ and the pose estimation $\boldsymbol{z}_c(k)$ (see Section II-B) at time step $k$ could be obtained correspondingly. These sensing feedback pairs enable the establishment of a mapping relationship. A pose estimation model using FBG feedback can, thus, be trained.

ELM is an online-updated algorithm employing single-hidden-layer feedforward networks (SLFNs) with randomly assigned input biases and weights. It can facilitate rapid initialization and updates of a trained model, outperforming tuning-based training methods (e.g., fully connected network, FCN for short) in the rapid weight initialization, extremely fast learning speed (thousands of times faster than FCN [36], [37]), and strong generalization performance. Here, we improved an existing adaptive online sequential ELM (FOS-ELM) [38], making it capable of dynamic adaptation as well as outlier exclusion. Different from the standard ELM, our method enables online parameter updates with the adaptive forgetting scheme inherited from FOS-ELM [38] and selectively adopts the newly obtained samples based on the SLAM reprojection error. The main unknown parameter to be tuned during the training procedure is the output weights, which can be automatically calculated by a mathematical transformation. This part is for determining the mapping between wavelength shifts $\boldsymbol{\lambda}(k)$ and end-effector poses $\boldsymbol{z}_c(k)$ of the soft robot.

*Training*: A few sample pairs are collected for the pose estimation model's initialization, where the robot is actuated

by a predefined sequence $\boldsymbol{U} = \begin{bmatrix}\boldsymbol{u}(1) & \boldsymbol{u}(2) & \cdots & \boldsymbol{u}(N_0)\end{bmatrix}$ with $N_0$ steps of exploration. The corresponding sequences of wavelength shift and camera-based pose estimation are as follows:

$$\boldsymbol{\Lambda} = \begin{bmatrix}\boldsymbol{\lambda}(1) & \boldsymbol{\lambda}(2) & \cdots & \boldsymbol{\lambda}(N_0)\end{bmatrix} \in \mathbb{R}^{l\times N_0}$$

and

$$\boldsymbol{Z}_c = \begin{bmatrix}\boldsymbol{z}_c(1) & \boldsymbol{z}_c(2) & \cdots & \boldsymbol{z}_c(N_0)\end{bmatrix} \in \mathbb{R}^{7\times N_0}$$

respectively. The mapping

$$\boldsymbol{z}_c(k) = f(\boldsymbol{\lambda}(k)) \tag{1}$$

is to be learned. Consider the training set with input $\boldsymbol{\Lambda}$ and output $\boldsymbol{Z}_c$, with $N_0$ distinct training samples. The output of an SLFN with $N$ hidden nodes can be represented by [39], [40]

$$\boldsymbol{o}_j = \sum_{i=1}^{N}\boldsymbol{\beta}_i\phi_i(\boldsymbol{\lambda}(j)) = \sum_{i=1}^{N}\boldsymbol{\beta}_i\phi(\boldsymbol{\lambda}(j),\boldsymbol{a}_i,b_i),$$
$$j = 1,2,\ldots,N_0 \tag{2}$$

where for the $i$th hidden node, $\boldsymbol{a}_i = \begin{bmatrix}a_{i1} & a_{i2} & \cdots & a_{il}\end{bmatrix}^{\mathrm{T}}$ and $\boldsymbol{\beta}_i = \begin{bmatrix}\beta_{i1} & \beta_{i2} & \cdots & \beta_{i,7}\end{bmatrix}^{\mathrm{T}}$ are the weighting vectors linking to the input nodes and output nodes, respectively. The activation node function is $\phi(\boldsymbol{\lambda}(j),\boldsymbol{a}_i,b_i)$, where $b_i$ is the threshold of the $i$th node. It is set as radial basis functions here, i.e.,

$$\phi_i(\boldsymbol{\lambda}(j)) = \phi(\boldsymbol{\lambda}(j),\boldsymbol{a}_i,b_i) = \exp\left(\frac{\|\boldsymbol{\lambda}(j)-\boldsymbol{a}_i\|^2}{b_i}\right). \tag{3}$$

Represent the inner product of vectors $\boldsymbol{a}_i$ and $\boldsymbol{\lambda}(j)$ as $\boldsymbol{a}_i\boldsymbol{\lambda}(j)$. Equation (2) can be written compactly as follows:

$$\boldsymbol{\Phi}\boldsymbol{\beta} = \boldsymbol{O} \tag{4}$$

where

$$\boldsymbol{\Phi} = \begin{bmatrix}\phi(\boldsymbol{a}_1\boldsymbol{\lambda}(1)+b_1) & \cdots & \phi(\boldsymbol{a}_N\boldsymbol{\lambda}(1)+b_N)\\ \vdots & \cdots & \vdots\\ \phi(\boldsymbol{a}_1\boldsymbol{\lambda}(N_0)+b_1) & \cdots & \phi(\boldsymbol{a}_N\boldsymbol{\lambda}(N_0)+b_N)\end{bmatrix} \in \mathbb{R}^{N_0\times N}$$

$$\boldsymbol{\beta} = \begin{bmatrix}\boldsymbol{\beta}_1^{\mathrm{T}} & \boldsymbol{\beta}_2^{\mathrm{T}} & \cdots & \boldsymbol{\beta}_N^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{N\times 7},$$

$$\mathbf{O} = \begin{bmatrix}\boldsymbol{o}_1^{\mathrm{T}} & \boldsymbol{o}_2^{\mathrm{T}} & \cdots & \boldsymbol{o}_{N_0}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{N_0\times 7}.$$

Here, $\boldsymbol{\Phi}$ is called the output matrix of the hidden layer. Obviously, proper values of parameters in $\boldsymbol{\Phi}$ will result in

$$\boldsymbol{\Phi}\boldsymbol{\beta} = \boldsymbol{Z}_c. \tag{5}$$

The goal during training is to minimize the network cost function $\|\boldsymbol{O}-\boldsymbol{Z}_c\|$, i.e., to find a solution vector that includes the to-be-tuned $\boldsymbol{a}_i^{\mathrm{T}}$, $\boldsymbol{\beta}_i^{\mathrm{T}}$, and $b_i^{\mathrm{T}}$, $i = 1, 2, \ldots, N$. Several algorithms can be utilized to search this solution, e.g., gradient-based iteration and least-square solution. Here, we use the latter method. An advantage of the ELM algorithm is that the values of input weights $\boldsymbol{a}_i$ and hidden-layer threshold $b_i$ could be assigned randomly without having to consider the input data; thus, the output matrix $\boldsymbol{\Phi}$ could then be obtained. Given an input set $\boldsymbol{\Lambda}$, the least-square solution $\widehat{\boldsymbol{\beta}}$ of linear system (5) could then be determined by

$$\left\|\phi(\boldsymbol{A},\boldsymbol{b})\widehat{\boldsymbol{\beta}} - \boldsymbol{Z}_c\right\| = \min_{\boldsymbol{\beta}}\|\phi(\boldsymbol{A},\boldsymbol{b})\boldsymbol{\beta} - \boldsymbol{Z}_c\| \tag{6}$$

where $\boldsymbol{A} = \{\boldsymbol{a}_1 \quad \boldsymbol{a}_2 \quad \cdots \quad \boldsymbol{a}_N\}$ and $\boldsymbol{b} = \{b_1 \quad b_2 \quad \cdots \quad b_N\}$. Finally, the output weights $\boldsymbol{\beta}$ will be analytically determined as follows:

$$\widehat{\boldsymbol{\beta}} = \boldsymbol{\Phi}^{\dagger}\boldsymbol{Z}_c \tag{7}$$

where $\boldsymbol{\Phi}^{\dagger}$ means the Moore–Penrose (MP) generalized inverse of $\boldsymbol{\Phi}$. After these steps, the global nonlinear mapping model (1) generated by ELM is ready for prediction. The initialization step hereto is the whole procedure of standard ELM, which is an offline training method. Its robustness is determined by the MP inverse, possibly resulting in low overall estimation accuracy. However, the MP inverse is only employed in the initialization phase for network weights calculation, and the subsequent online update can weaken the adverse effects of MP inverse.

*Prediction*: Provided with the wavelength shift $\boldsymbol{\lambda}(k)$ at the $k$th time step obtained, the corresponding pose of robot end-effector can be calculated by

$$\boldsymbol{z}_w(k) = f(\boldsymbol{\lambda}(k)), \quad k = 1,2,\ldots. \tag{8}$$

Since during the prediction procedure, the model is independent of camera-based pose estimation $\boldsymbol{z}_c(k)$ (see Section II-B), it could be regarded as another pose estimator that could be further fused with $\boldsymbol{z}_c(k)$.

*Updating*: Suppose the existing prediction vector $\boldsymbol{\beta}^{(0)}$ is obtained by initial training dataset $\mathbf{D}^{(0)}$ composed of $\boldsymbol{\Lambda}$ and $\boldsymbol{Z}_c$ with $N_0$ distinct sample pairs. The expression of $\boldsymbol{\beta}^{(0)}$ based on (7) could be rewritten as

$$\boldsymbol{\beta}^{(0)} = \left(\left(\boldsymbol{\Phi}^{(0)}\right)^{\mathrm{T}}\boldsymbol{\Phi}^{(0)}\right)^{-1}\left(\boldsymbol{\Phi}^{(0)}\right)^{\mathrm{T}}\boldsymbol{Z}_c = \left(\boldsymbol{K}^{(0)}\right)^{-1}\left(\boldsymbol{\Phi}^{(0)}\right)^{\mathrm{T}}\boldsymbol{Z}_c. \tag{9}$$

When a new set of training data $\mathbf{D}^{(1)}$ with $N_1$ distinct sample pairs is available for ELM, the weighting vector $\boldsymbol{\beta}^{(1)}$ corresponding to both $\mathbf{D}^{(0)}$ and $\mathbf{D}^{(1)}$ can be calculated as follows:

$$\boldsymbol{\beta}^{(1)} = \begin{bmatrix}\boldsymbol{\Phi}^{(0)}\\ \boldsymbol{\Phi}^{(1)}\end{bmatrix}^{+}\begin{bmatrix}\boldsymbol{Z}_c^{(0)}\\ \boldsymbol{Z}_c^{(1)}\end{bmatrix}$$
$$= \left(\boldsymbol{K}^{(1)}\right)^{-1}\left(\boldsymbol{\Phi}^{(1)}\right)^{\mathrm{T}}\left(\boldsymbol{Z}_c^{(1)} - \boldsymbol{\Phi}^{(1)}\boldsymbol{\beta}^{(0)}\right) + \boldsymbol{\beta}^{(0)} \tag{10}$$

where

$$\boldsymbol{K}^{(1)} = \begin{bmatrix}\boldsymbol{\Phi}^{(0)}\\ \boldsymbol{\Phi}^{(1)}\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}\boldsymbol{\Phi}^{(0)}\\ \boldsymbol{\Phi}^{(1)}\end{bmatrix} = \left(\boldsymbol{\Phi}^{(1)}\right)^{\mathrm{T}}\boldsymbol{\Phi}^{(1)} + \boldsymbol{K}^{(0)}.$$

Following this iteration, the ELM model would be updated after the $k$th training dataset $\mathbf{D}^{(k)}$ as follows [36]:

$$\boldsymbol{\beta}^{(k)} = \left(\boldsymbol{K}^{(k)}\right)^{-1}\left(\boldsymbol{\Phi}^{(k)}\right)^{\mathrm{T}}\left(\boldsymbol{Z}_c^{(k)} - \boldsymbol{\Phi}^{(k)}\boldsymbol{\beta}^{(k-1)}\right) + \boldsymbol{\beta}^{(k-1)} \tag{11}$$

where

$$\boldsymbol{K}^{(k)} = \left(\boldsymbol{\Phi}^{(k)}\right)^{\mathrm{T}}\boldsymbol{\Phi}^{(k)} + \boldsymbol{K}^{(k-1)}.$$

In consideration of the possible deteriorated camera-based estimations due to the poor image quality, it is necessary to set an activation threshold of the model updating mechanism. The reprojection error $e_S$ mentioned in Section II-B can be utilized as such an indication to determine whether the newly obtained sample is incorporated for online learning. When $e_S$ is larger than the threshold ($> 1.4$), the matrix $\boldsymbol{\beta}^{(k)}$ will keep the value as in the last iteration step.

The reduction of effects from old data in the update procedure of ELM model can be achieved by introducing and adjusting several weight parameters for the old measurements. Equation (11) can be expressed in the form of

$$\boldsymbol{\beta}^{(k)} = \left( \begin{bmatrix} \boldsymbol{\Phi}^{(k-1)} \\ \boldsymbol{\Phi}^{(k)} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \boldsymbol{\Phi}^{(k-1)} \\ \boldsymbol{\Phi}^{(k)} \end{bmatrix} \right)^{-1} \times \begin{bmatrix} \boldsymbol{\Phi}^{(k-1)} \\ \boldsymbol{\Phi}^{(k)} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \boldsymbol{Z}_c^{(k-1)} \\ \boldsymbol{Z}_c^{(k)} \end{bmatrix} = \mathbf{H}^{(k)} \mathbf{M}^{(k)} \tag{12}$$

where

$$\mathbf{H}^{(k)} = \left[ \left( \boldsymbol{\Phi}^{(k)} \right)^{\mathrm{T}} \boldsymbol{\Phi}^{(k)} + \left( \boldsymbol{\Phi}^{(k-1)} \right)^{\mathrm{T}} \boldsymbol{\Phi}^{(k-1)} \right]^{-1} \tag{13}$$

$$\mathbf{M}^{(k)} = \left( \boldsymbol{\Phi}^{(k)} \right)^{\mathrm{T}} \boldsymbol{Z}_c^{(k)} + \left( \boldsymbol{\Phi}^{(k-1)} \right)^{\mathrm{T}} \boldsymbol{Z}_c^{(k-1)}. \tag{14}$$

Weighting $w$ is added to the variables related to old training samples; thus, the two factors (13) and (14) will be

$$\widehat{\mathbf{H}}^{(k)} = \left[ \left( \boldsymbol{\Phi}^{(k)} \right)^{\mathrm{T}} \boldsymbol{\Phi}^{(k)} + w \left( \boldsymbol{\Phi}^{(k-1)} \right)^{\mathrm{T}} \boldsymbol{\Phi}^{(k-1)} \right]^{-1} \tag{15}$$

$$\widehat{\mathbf{M}}^{(k)} = \left( \boldsymbol{\Phi}^{(k)} \right)^{\mathrm{T}} \boldsymbol{Z}_c^{(k)} + w \left( \boldsymbol{\Phi}^{(k-1)} \right)^{\mathrm{T}} \boldsymbol{Z}_c^{(k-1)}. \tag{16}$$

The recursive expression of (15) can be obtained by Sherman–Morrison formula as follows [38]:

$$\widehat{\mathbf{M}}^{(k)} = \frac{\widehat{\mathbf{M}}^{(k-1)}}{w} - \frac{\mathbf{N}^{(k)} \left( \mathbf{N}^{(k)} \right)^{\mathrm{T}}}{w \left[ w + \boldsymbol{\Phi}^{(k)} \mathbf{N}^{(k)} \right]} \tag{17}$$

where $\mathbf{N}^{(k)} = \boldsymbol{\Phi}^{(k)} \widehat{\mathbf{M}}^{(k-1)}$.

### D. Camera-FBG Sensing Fusion

In the ORB-SLAM2 algorithm, the reprojection error $e_S$ (introduced in Section II-B) can reflect the pose estimation accuracy. The fusion result can be regarded as a combination of SLAM and F-emp portions, with an adjustable weighting that characterizes the visual sensing accuracy. Based on this error $e_S$, the weighting of SLAM portion in sensing fusion can be determined. The final pose estimation can, thus, be obtained by the following criteria (18) shown at the bottom of this page, where $E_L$ and $E_U$ are the error bounds distinguishing whether or not to entirely trust or discard the SLAM estimation, respectively; $K_S$ is an adjusting factor.

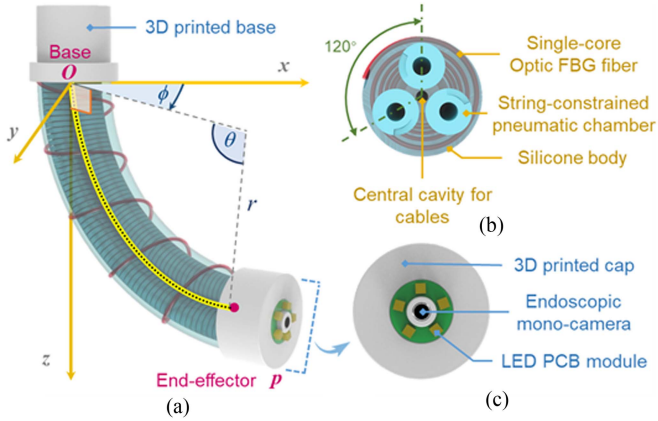


Fig. 4. Structural diagram of the continuum robot mounted with LEDs and a camera at its tip. (a) Configuration parameters $r$, $\theta$, and $\phi$ defined to describe a spatial arc of the CC-based model. (b) Cross section showing three air chambers for robot actuation. (c) Endoscopic camera providing real-time visual feedback to ORB-SLAM2 for camera pose estimation.

## III. Hybrid Position Control of Soft Robots

### A. Kinematics Initialization by CC Model

In consideration of the limited dimensions of the actuation space compared with the task space, only the position $\boldsymbol{p}$ or orientation $\boldsymbol{q}$ can be controlled. In this section, the control objective is illustrated by the end-effector position $\boldsymbol{p}$. As in this article, we utilize such a single-segment soft manipulator to demonstrate the idea of visual-strain fusion and hybrid control, and the controllable number of DoFs would be limited to 2. This results from a curved surface workspace with negligible thickness. Here, positional DoFs $x$ and $y$ are controlled as an example. The robot kinematics can be initialized based on the CC assumption, which is widely applied to continuum robots. The CC model is constructed based on the assumptions of zero torsion and uncoupling among actuation chambers. That is, the bending robot is assumed as no torsion involved, and the length variation of inflated chambers will not affect the other chambers. Two main parts need to be considered during the modeling, namely mappings from configuration space to task space and from joint space to configuration space. The former mapping is robot independent, while the latter mapping is robot specific as it is related to the robot actuation mechanism. In the CC-based model, the three parameters in the configuration space can be interpreted as $r$, $\theta$, and $\phi$, representing the radius, the central angle, and the bending direction (rotation angle) of the arc, respectively [see Fig. 4(a)].

The lengths of the three fluidic chambers [see Fig. 4(b)] are denoted as $\widehat{\boldsymbol{u}} = \begin{bmatrix} l_1 & l_2 & l_3 \end{bmatrix}^T$. The angle $\phi$ representing the

$$\boldsymbol{z} = \begin{cases} \boldsymbol{z}_c, & e_S \le E_L \\ K_S \left( e_S - E_L \right) \boldsymbol{z}_w - \left[ 1 - K_S \left( e_S - E_L \right) \right] \boldsymbol{z}_c, & E_L < e_S < E_U \\ \boldsymbol{z}_w, & e_S \ge E_U \end{cases} \tag{18}$$

bending direction could be obtained as follows:

$$\phi = \tan^{-1}\left(\sqrt{3}\left(l_2 + l_3 - 2l_1\right) \Big/ 3\left(l_2 - l_3\right)\right) \tag{19}$$

as well as the backbone arc curvature $\kappa$ as

$$\kappa = \frac{l_2 + l_3 - 2l_1}{\left(l_1 + l_2 + l_3\right) d \sin\phi}. \tag{20}$$

The central angle of the backbone arc can be obtained by $\theta = \kappa l$, while the axial length $l$ of robot yields $l = (l_1 + l_2 + l_3)/3$. To summarize, the expression of three parameters $r$, $\theta$, and $\phi$ in the configuration space can be represented as follows:

$$\begin{cases} r\left(\widehat{\boldsymbol{u}}\right) = d\left(l_1 + l_2 + l_3\right)/2\delta \\ \theta\left(\widehat{\boldsymbol{u}}\right) = 2\delta/3d \\ \phi\left(\widehat{\boldsymbol{u}}\right) = \tan^{-1}\left(\sqrt{3}\left(l_2 + l_3 - 2l_1\right)/3\left(l_2 - l_3\right)\right) \end{cases} \tag{21}$$

where $\delta = \left(l_1^2 + l_2^2 + l_3^2 - l_1 l_2 - l_1 l_3 - l_2 l_3\right)^{1/2}$.

The position of end-effector $\boldsymbol{p}$ in the coordinate, as shown in Fig. 4(a), can be found as follows:

$$\begin{aligned} &\boldsymbol{p}\left(r\left(\widehat{\boldsymbol{u}}\right), \theta\left(\widehat{\boldsymbol{u}}\right), \phi\left(\widehat{\boldsymbol{u}}\right)\right) \\ &= \begin{bmatrix} r\left(\widehat{\boldsymbol{u}}\right)\left(1 - \cos\left(\theta\left(\widehat{\boldsymbol{u}}\right)\right)\right)\cos\left(\phi\left(\widehat{\boldsymbol{u}}\right)\right) \\ r\left(\widehat{\boldsymbol{u}}\right)\left(1 - \cos\left(\theta\left(\widehat{\boldsymbol{u}}\right)\right)\right)\sin\left(\phi\left(\widehat{\boldsymbol{u}}\right)\right) \\ r\left(\widehat{\boldsymbol{u}}\right)\sin\left(\theta\left(\widehat{\boldsymbol{u}}\right)\right) \end{bmatrix}. \end{aligned} \tag{22}$$

The corresponding differential format can be expressed as follows:

$$\dot{\boldsymbol{p}} = \boldsymbol{J}\dot{\widehat{\boldsymbol{u}}} \tag{23}$$

where the Jacobian matrix $\boldsymbol{J}$ can be calculated by differentiating the position $\boldsymbol{p}$ with respect to the input $\widehat{\boldsymbol{u}}$. Proved with the matrix $\boldsymbol{J}$, we could establish its inverse function as follows:

$$\dot{\widehat{\boldsymbol{u}}} = \boldsymbol{J}^{\dagger}\dot{\boldsymbol{p}} \tag{24}$$

where $\boldsymbol{J}^{\dagger}$ is the generalized inverse of $\boldsymbol{J}$. A singularity point exists on the initial status, that is, when the robot body is straight and aligns with the $z$-axis as in Fig. 4(a). This circumstance could be handled by adding tiny-value (e.g., $10^{-5}$) variations on the chamber lengths during the calculation of $\boldsymbol{J}$ inverse. To involve constraints during generating motion commands for solving redundancy, e.g., to maintain the closest status to a preferred configuration, this scheme could be extended by an additional factor as [41] follows:

$$\Delta\widehat{\boldsymbol{u}}^{(k+1)} = \boldsymbol{J}^{\dagger}\Delta\boldsymbol{p}^{*(k+1)} + \left(\boldsymbol{1} - \boldsymbol{J}^{\dagger}\boldsymbol{J}\right)\cdot\beta\left(\widehat{\boldsymbol{u}}_0 - \widehat{\boldsymbol{u}}^{(k)}\right) \tag{25}$$

where $\beta(\widehat{\boldsymbol{u}}_0 - \widehat{\boldsymbol{u}}^{(k)})$ is used to find a redundant solution approaching the preferred robot configuration, which could be set as the initial configuration $\widehat{\boldsymbol{u}}_0$ without actuation air pressure or only with prepressure. The original chamber length is denoted by $\widehat{\boldsymbol{u}}_0$ and the length in current time step $k$ is $\widehat{\boldsymbol{u}}^{(k)}$.

Therefore, during the runtime, once a command of the desired displacement $\Delta\boldsymbol{p}^{*}$ is given, the corresponding change of three chambers $\Delta\widehat{\boldsymbol{u}}$ could be obtained to calculate the actuation command of stepper motors controlling the chamber air pressure. Thus, for the $k$th time step, the new chamber lengths can be formed as follows:

$$\widehat{\boldsymbol{u}}^{(k+1)} = K_p\Delta\widehat{\boldsymbol{u}}^{(k+1)} + \widehat{\boldsymbol{u}}^{(k)} \tag{26}$$

where $K_p$ is a proportional gain to adjust the change of chamber length. To simplify the modeling for fluid dynamics, we assume that the extension of chamber lengths has a linear positive correlation relation with the stepper motors' output $\boldsymbol{u}$, i.e.,

$$\boldsymbol{u}^{(k+1)} = \boldsymbol{\alpha}\cdot\left(\widehat{\boldsymbol{u}}^{(k+1)} - \widehat{\boldsymbol{u}}_0\right) \tag{27}$$

where $\boldsymbol{\alpha}$ is a diagonal matrix, including the three multiples for three chambers. However, in actuality, this linearization could not describe the transformation well. Nonlinear elongation of elastic chambers and transmission of fluids, as well as other modeling uncertainties, would induce errors in the robot control. Thus, a learning-based component to compensate tracking deviations is investigated in the Section III-B.

### B. Online Data-Driven Error Compensator

The online update of an additional error compensator enables the controller to compensate steady errors and even adapt to mechanical property changes, e.g., material fatigue. Once the actuation change $\Delta\boldsymbol{u}(k)$ is executed in a new step, the corresponding actual motion vector $\Delta\boldsymbol{p}(k)$ could be estimated as in Section II. Thereby, a set of new sample pairs, including input $\mathbf{x} = \left[\boldsymbol{u}(k-1)^{\mathrm{T}}, \Delta\boldsymbol{z}(k)^{\mathrm{T}}\right]^{\mathrm{T}}$ and output $\mathbf{y} = \Delta\boldsymbol{p}_e(k)$ would be produced. Here, the variable $\Delta\boldsymbol{p}_e(k)$ represents the difference between desired motion and actual motion, i.e.,

$$\Delta\boldsymbol{p}_e = \Delta\boldsymbol{p}^{*} - \Delta\boldsymbol{p}_r \tag{28}$$

where $\Delta\boldsymbol{p}_r$ is the actual motion in task space corresponding to the desired $\Delta\boldsymbol{p}^{*}$. The purpose of our proposed error compensator is to predict this error in advance and consider this potential deviation together with the desired robot movement. Thus, $\Delta\boldsymbol{p}^{*}$ would be improved after compensation as follows:

$$\Delta\tilde{\boldsymbol{p}}^{*} = \Delta\boldsymbol{p}^{*} + \Delta\boldsymbol{p}_e\cdot\left|\Delta\boldsymbol{p}^{*}\right|. \tag{29}$$

This newly collected sample could reflect the latest robot mechanical status and is added into the model training dataset, i.e., input matrix $\mathbf{X}$ and output $\mathbf{Y}$. GPR is utilized here for the model training. The working principle of GPR has been introduced in our previous work [34]. For each step of motion, the model would be retrained for updating. A size limitation $N_r^{\max}$ for this dataset is predefined to keep the prediction fast and effective. If the current size $N_r^{(k)} > N_r^{\max}$, the oldest sample $[\mathbf{x}_1, \mathbf{y}_1]$ will be discarded.

No prior data exploration is needed in robot manipulation. The robot can be actuated using the CC model-based controller first, while feedback for compensator initialization is being collected. After several motion steps $N_c$ ($\leq N_r$), the GPR-based compensator will be first trained and updated in the following steps. For time step $k > N_c$, an additional compensated component is added as in (29).

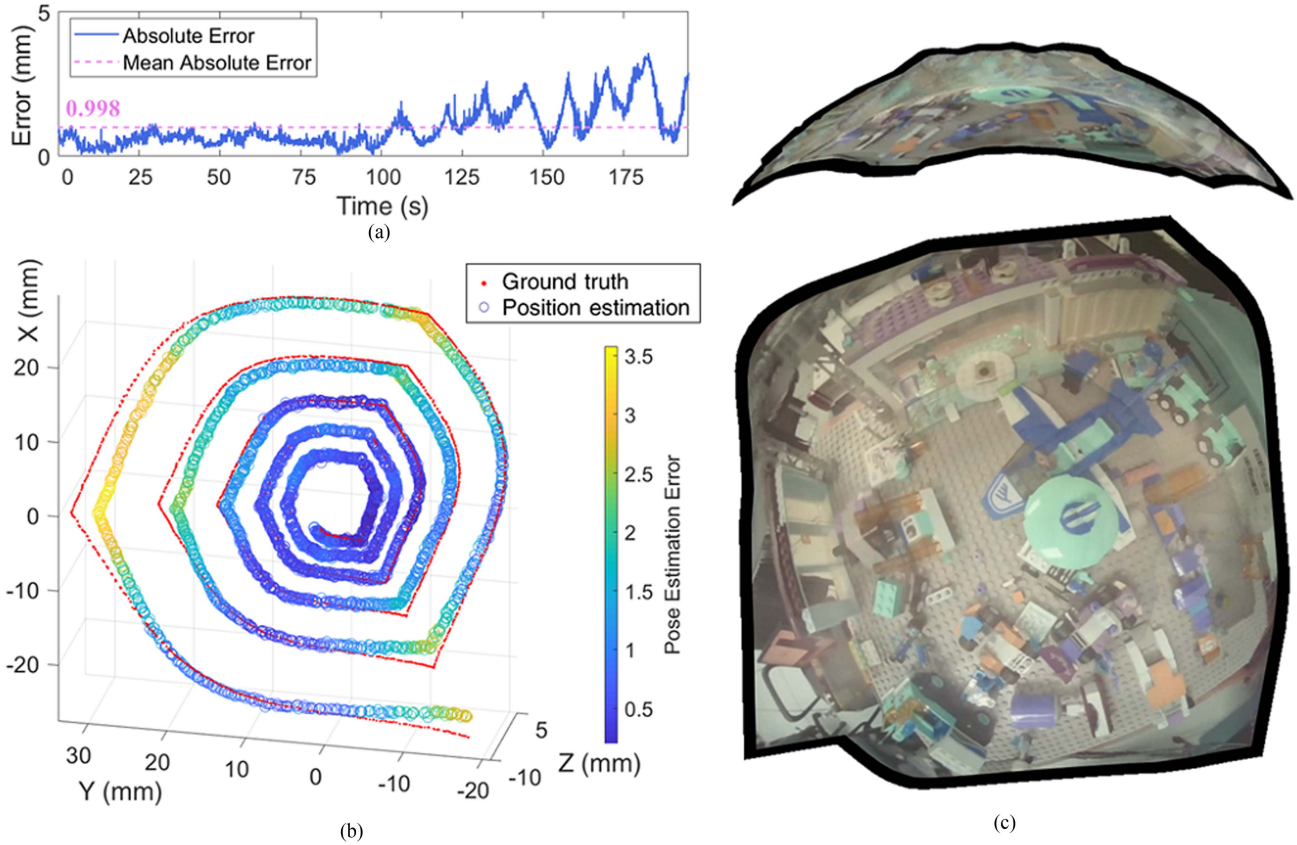


Fig. 5. Camera-based pose estimation results, where the SLAM-based estimation was compared with the EM-tracker measurement (ground truth). (a) Pose estimation errors as well as the mean value. (b) Trajectories recorded by EM trackers (red) and estimated by ORB-SLAM2 ("o"). (c) Front and side views of the stitched images in 3-D, which are reconstructed using the SLAM pose estimation and image feedback.

## IV. Experiments, Results, and Discussion

### A. Soft Robot With Monocamera and FBG Fiber

The continuum robot was molded by silicone rubber (Ecoflex30, Smooth-on Inc.), with a 3-D printed tip cap and a fixation base [see Fig. 4(a)]. Three pneumatic chambers are distributed in a distance of 5.1 mm to the robot's central axis and an angle of 120° between each other [see Fig. 4(b)], providing omnidirectional bending [42]. The chamber inflation was regulated by an actuation unit comprising of three pairs of stepper motors and cylinders. Precise angular position control could be implemented on the motors, thus adjusting the volumes of sealed cylinders connected to chambers. An endoscopic camera (depth of field: 8 to 150 mm) and a LED module were fixed on the tip cap [see Fig. 4(c)]. A single-core optical FBG fiber with 17 FBGs (6-mm long gratings, 20-mm spacing) was helically wrapped and adhered on the silicone continuum body. For the convenience of fabrication, the distances between adjacent turns of fiber were set at approximately 16.5 mm. The robot outer diameter was 20 mm, and the bendable part was 90 mm in length. As the robot base was fixed in the experiment and the twisting is negligible, the roll orientation would not be controllable.

### B. Pose Estimation by ORB-SLAM2

Pose estimation accuracy using ORB-SLAM2 was validated in a LEGO-constructed scenario. The robot was actuated according to a predefined sequence $\boldsymbol{U}$ for three stepper motor sets. This actuation sequence was expected to steer the robot to follow a trajectory that spreads out from the initial position.

A pair of EM trackers were attached on the robot tip to record the actual (ground truth) pose $\boldsymbol{z}$. Intrinsic calibration of the monocamera was performed first. Extrinsic calibration of monocular metric scale was also required in the initialization procedure. Before each time of manipulation, the robot would move slowly along one or two direction(s), until the initialization for visual features was ready. With the robot actuated by sequence $\boldsymbol{U}$ with $N$ steps, the set of SLAM estimation $\widehat{\boldsymbol{Z}}_c = \begin{bmatrix}\boldsymbol{z}_c(1) & \boldsymbol{z}_c(2) & \cdots & \boldsymbol{z}_c(N)\end{bmatrix}$ and EM tracker measurement $\widehat{\boldsymbol{Z}} = \begin{bmatrix}\boldsymbol{z}(1) & \boldsymbol{z}(2) & \cdots & \boldsymbol{z}(N)\end{bmatrix}$ can be obtained. The affine transformation from $\boldsymbol{P}$ to $\widehat{\boldsymbol{P}}_c$ could be calculated; thus, the SLAM position measurement is calibrated as follows:

$$\boldsymbol{p}_c = \mathbf{R}\widehat{\boldsymbol{p}}_c \cdot \mathbf{k} + \widehat{\boldsymbol{p}} \tag{30}$$

where $\mathbf{R}$, $\mathbf{k}$, and $\widehat{\boldsymbol{p}}$ are the rotation matrix, scale factor along all dimensions, and translation vector, respectively. Measurement errors of $\boldsymbol{P}_c$ comparing with $\boldsymbol{P}$ were calculated. The mean absolute errors of ORB-SLAM2 along $x$, $y$, and $z$ axes were, respectively, 0.508 mm, 0.596 mm, and 0.385 mm, while the root-mean-square error (RMSE) was 0.998 mm [see Fig. 5(a)]. The trajectories constructed by $\boldsymbol{P}_c$ and $\boldsymbol{P}$ could be found in Fig. 5(b). As can be seen in conditions of abundant visual

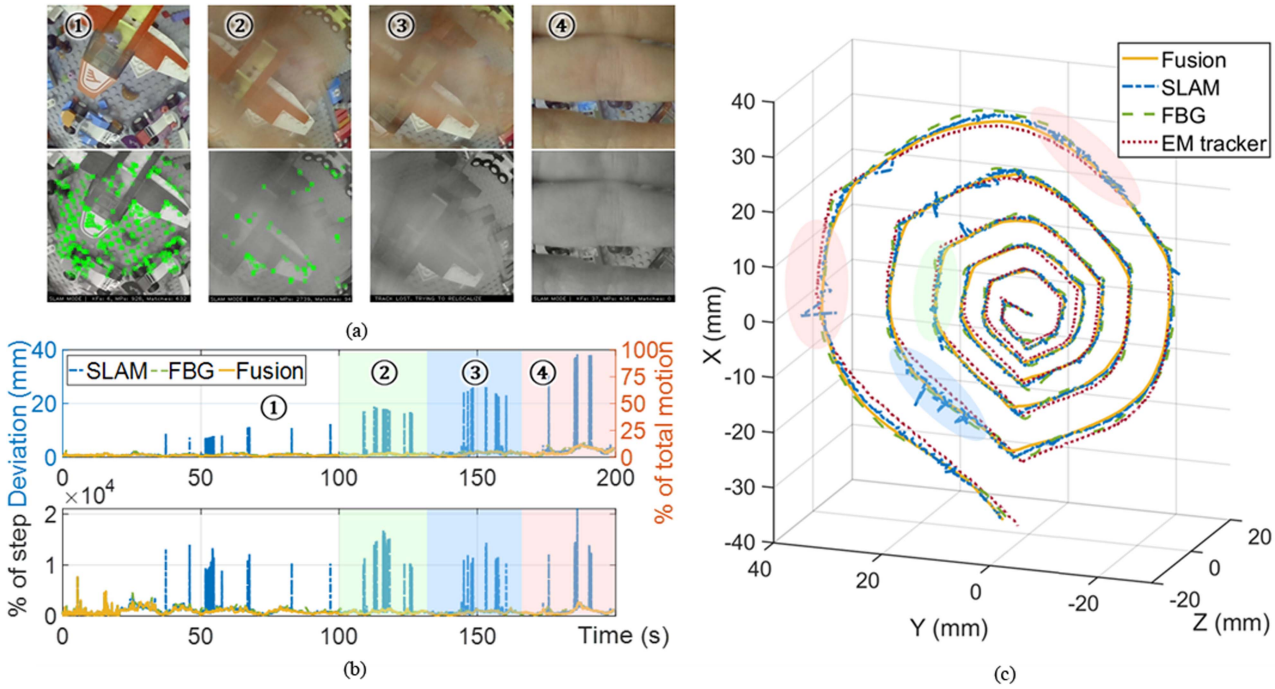


Fig. 6. Sensor fusion performance in the presence of moving obstacles. (a) Examples of camera view and corresponding visual features under ① LEGO-constructed scenario with abundant visual features; ② moving/static hand where features were partly detected; ③ moving hand with no features detected; ④ static hand where all the features were obscured for several seconds. (b) Deviations of SLAM-based and fusion-based pose estimation compared with the ground truth measured by EM trackers. Percentages of error with respect to total motion range and each-step motion are provided, respectively. (c) Trajectories of fusion-, SLAM-, and FBG-based camera positions.

features, the robot manipulation is slow and smooth, and ORB-SLAM2 is reliable, acting as the benchmark of pose estimation for our following training. In soft robot manipulation, this kind of sensing would not rely on any other external position sensors, e.g., EM trackers.

Through the pose estimation and corresponding image views, the 6-D image stitching of scenes is possible [see Fig. 5(c)]. Different from SfM or other feature-based methods, this reconstruction is simplified by positioning the images according to the estimated end-effector/camera pose. This will effectively increase the function of sparse SLAM algorithm as well as the mosaicking efficiency.

### C. Sensor Fusion Pose Estimation

The ELM was chosen to train and update the F-emp pose estimation model in real time. With the same actuation sequence as in Section IV-B, wavelength shifts were collected and trained by ELM to estimate end-effector pose at the same time. We set the numbers of hidden nodes and initialization samples as $N = 200$ and $N_0 = 450$, respectively. These parameters were roughly tuned referring to the ELM estimation accuracy. The interval between adjacent steps was set as 0.05 s. It could be found in our experiments that the prediction result of ELM would be improved with the increment of sample number $N_0$. Once a necessary number to guarantee the convergence of prediction was satisfied, the accuracy would not significantly increase. The ELM model was updated every newly obtained sample. The prediction results were compared with the measurement using ORB-SLAM2, which was the benchmark in training. The trajectory reconstructed by ELM estimation approached that of the SLAM measurement closely. The mean estimation errors along $x$, $y$, and $z$ axes of ELM were $1.82\times10^{-4}$ mm, $3.95\times10^{-4}$ mm, and $4.39\times10^{-4}$ mm, respectively, while the mean spatial error was $8.28\times10^{-4}$ mm. This result demonstrates that ELM is capable of learning the pose information utilizing wavelength feedback. To test the pose estimation effectiveness when SLAM is unable to achieve consistent and stable estimation, we validated the sensing fusion methodology in the following two conditions, which are under moving visual obstacles and under the varying lighting condition, respectively.

*1) Under Moving Obstacles:* The robot was actuated with a similar predefined spiral sequence, as shown in Section IV-B. Moving or stable obstacles would disturb camera view, with examples of the camera view and features, as illustrated in Fig. 6(a). In the first 100 s [i.e., first 2000 time steps, marked with ① in Fig. 6(a)], no disturbances in the camera view were applied. For the testing scenario, this period would guarantee around 600 features in each frame, and the mean error of SLAM and fusion was 0.840 mm and 0.768 mm, respectively. After that, a hand was positioned statically in front of the camera or moving quickly to partly shield the field-of-view (marked with ② and ③). Neither case would consistently cover all features in the camera view. During these periods, the number of features was less than or even reduced to 0 in rare frames, the mean SLAM error was 1.694 mm (4.2% to the largest distance to starting point, 40.23 mm) and the maximum error was 26.272 mm (65.3% to the largest distance), while the

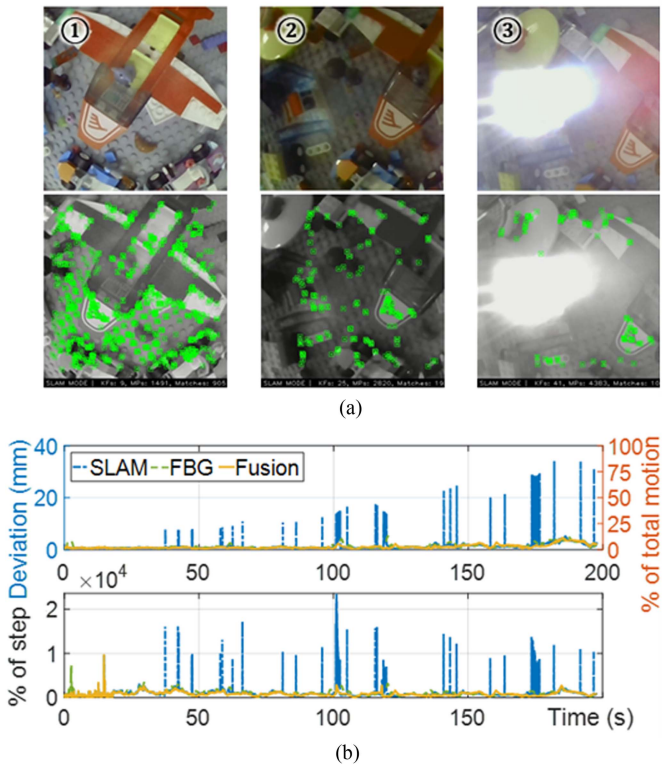


Fig. 7. Sensor fusion under varying lighting conditions. (a) Examples of camera view and corresponding visual features with ① ordinary lab lighting, ② low-level lighting, and ③ moving additional LED lighting. (b) Deviations of SLAM-, FBG-, and fusion-based pose estimation compared with the ground truth obtained by EM tracking. Percentages of error with respect to total motion range and each-step motion are provided, respectively.

fusion errors had a mean of 1.132 mm (2.8% to the largest distance) and max. of 2.573 mm (6.4% to the largest distance). For the last case, the hand would statically cover the whole field-of-view (marked with ④) for ∼2 s each time. During this period, the visual features would be consistently lost; thus, the SLAM procedure would be stagnated (max. error 38.483 mm, 95.7%), but the fusion results could be maintained (max. error 4.747 mm, 11.8%). In the moving-obstacle period, there would also be moments where all the features in the camera view were blurred ③, resulting in the SLAM estimation pausing. However, in our sensing modality, the F-emp estimator would compensate for the lack of visual sensing and guarantee the acquisition of sensing feedback. In the control and plot, if no SLAM estimation was provided, we would set the SLAM pose as the latest valid value to avoid the lack of feedback. As shown in Fig. 6(b) and (c), including the plots of errors, the trajectory of fused pose would not be affected by the moving obstacles, while pure SLAM-based pose would be deteriorated as expected. However, there is still a defect of the SLAM estimation that could not be obviously resolved by the fusion method. As can be seen in the lower subfigure of Fig. 6(b) demonstrating the error percentage to the corresponding motion step, the smaller step lengths (e.g., during first 50 s, mean step size 0.12 mm) will result in poorer estimation (mean errors 0.82 mm, 1074.4% and 0.75 mm, 966.8% for SLAM and fusion results, respectively).

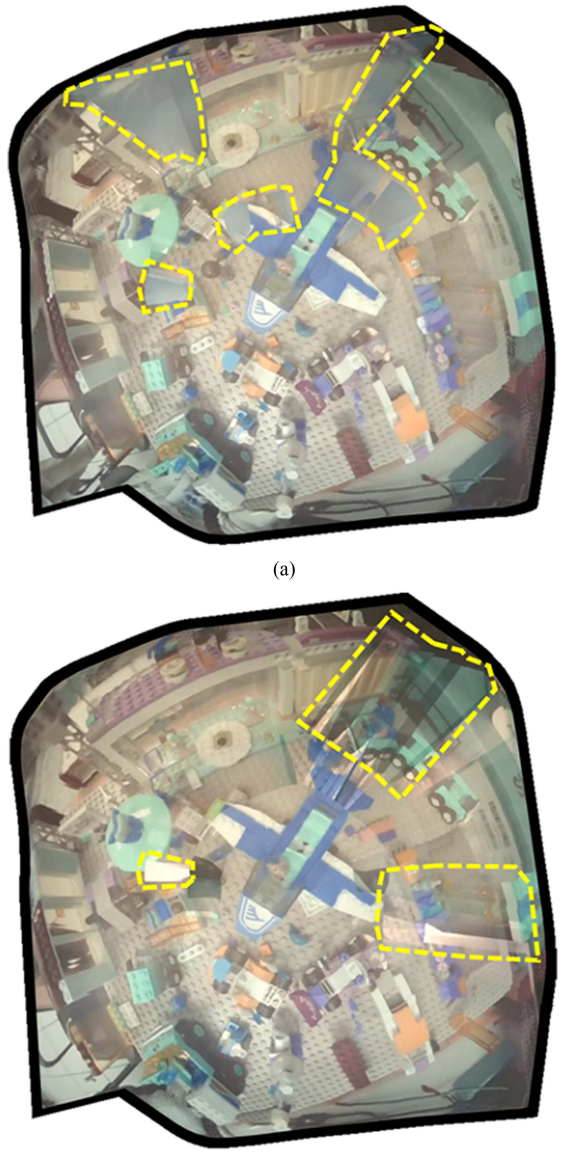


Fig. 8. Scenario reconstructions with disturbances of (a) moving obstacle (see Fig. 6) and (b) varying lighting condition (see Fig. 7). Several blurs due to the obstacle or varying lighting are indicated by dotted outlines.

The noise of SLAM dominated the estimation error in such tiny steps, and this is also a reason for the high percentage throughout the whole procedure. Even in the outer trajectory (e.g., during the last 175–200 s), the step size was only 0.29 mm. Therefore, we may conclude that the advantage of SLAM is the overall localization rather than incremental estimation, and this behavior would also be maintained in the fusion method based on SLAM.

*2) Under Varying Lighting Conditions:* Varying lighting conditions were also tested in the pose estimation experiment (see Fig. 7). Under the lighting provided by incandescent lamp [marked with ① in Fig. 7(a)], the robot started the same series of movement as in Section IV-C1.

During manipulation of the robot, the lighting condition was changed gradually or abruptly to weak lighting (marked

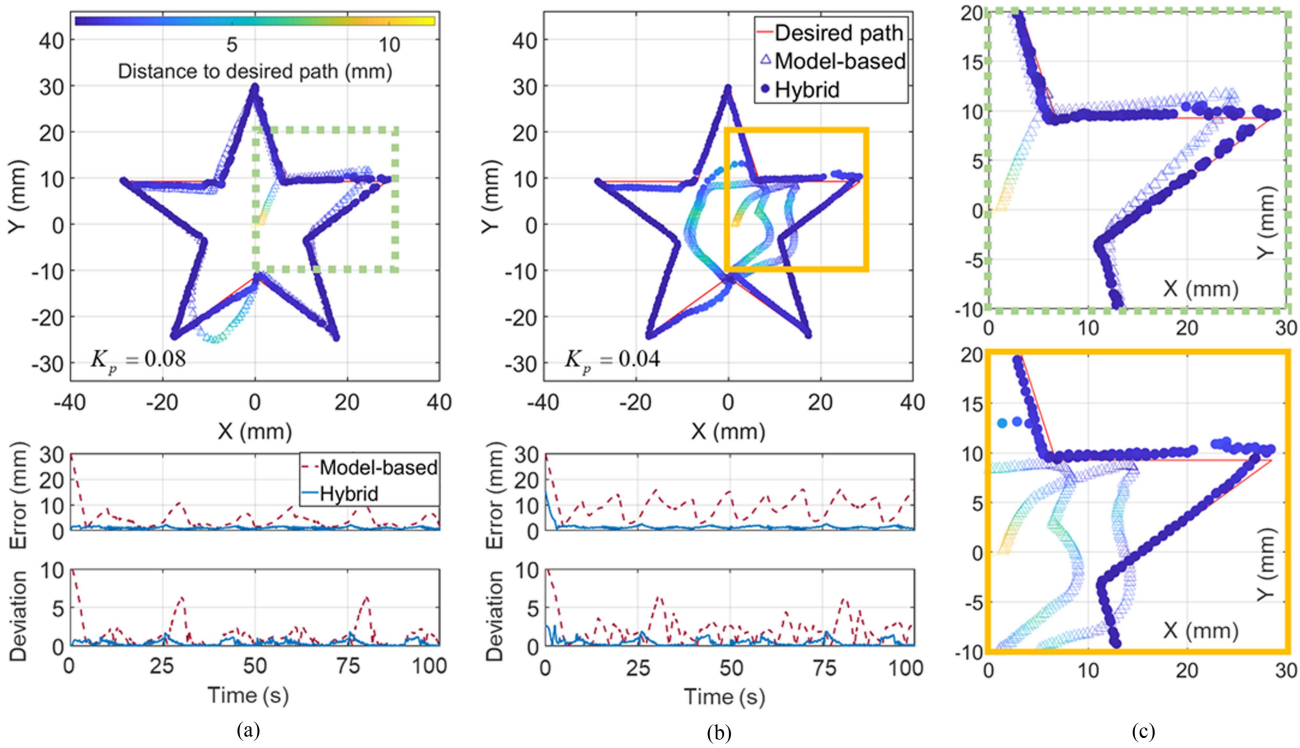


Fig. 9. Comparison of control performance tracking a target moving along a pentagram path. Tracking trajectories and errors with gain factor $K_p$ as (a) 0.08 and (b) 0.04 are plotted. "Error" represents the Euclidean distance between the current target and the actual end-effector position. "Deviation" is calculated as the distance from end-effector position to the closest point on the desired path. The model-based method was validated for the first 100 s (step 1–2000), afterward another 100 s run with the hybrid control. Two series of 100-s period data are plotted and overlaid for ease of comparison. (c) Zoomed-in view of the square blocks in (a) and (b).

with ②) or complete darkness during 75–150 s (i.e., time step 1500–3000), and then returning to the initial lighting. An additional moving LED source was tested from 150 to 200 s (step 3000–4000), which was oriented directly toward the endoscopic camera (③). It could be seen that low-light level would reduce the number of visual features in the camera view and induce minor noise in the SLAM-based pose estimation [see Fig. 7(b)]. However, once the lighting was fully removed in the camera view (which resulted in entire black image feedback), the SLAM procedure would be interrupted due to the absence of features. The moving lighting source would also bring consistent noise to SLAM estimation. When the LED was directly facing the camera, most of the features (especially on the lighting spot area) would be lost and SLAM estimation error increased. The fusion-based estimation could keep a stable estimation level (RMSE: 1.324 mm, 3.4% to the largest distance to starting point, 40.23 mm), largely improving the estimation accuracy compared with SLAM (RMSE: 3.116 mm, 3.4% to the largest distance). A similar limitation on the tiny incremental motion estimation, as discussed at the end of Section IV-C1, also appeared in this varying lighting test.

Scene reconstructions under conditions in Section IV-C1 and C2 were also conducted (see Fig. 8). Poses for image stitching were provided by the sensing fusion result. The blurs caused by the moving hand and varying lighting were reflected in the reconstruction, several of which are marked with a dotted outline. Although the images were blurred, the whole mosaicking could still successfully stitch together, accredited to the stable and consistent feedback of sensing fusion.

### D. Tracking: Hybrid Control Versus Model-Based Control

Control performance comparisons were tested by target tracking along two kinds of paths: a pentagram trajectory comprising of straight lines and sharp angles (see Fig. 9); and a circle trajectory (see Fig. 10). It should be noticed that the control task aims at the positions on the $x$ and $y$ axes. Paths shown in the following figures are projections on the $x$–$y$ plane, while the actual end-effector trajectory is distributed on the spatial curve surface. EM trackers were used to provide the sensing feedback in this section. During this tracking task, the target was moving at a consistent speed. Each iteration of control loop was set as 0.05 s. As we know, model-based control could guarantee stable performance but needs parameter tuning to achieve higher accuracy. Besides those related to robot structures, others, such as the proportional–integral–derivative (PID) factors, would also affect the convergence performance in tracking tasks. The following two experiments are to validate the effects of online-learning-based portion in the hybrid control.

Proportional gain $K_p$ in (26) is to adjust the calculated chamber length change, which plays an important role in the CC model-based control. The smaller its value is, the smaller the actuation change will be. We roughly set several different values of $K_p$ and tested if the learning-based portion could compensate

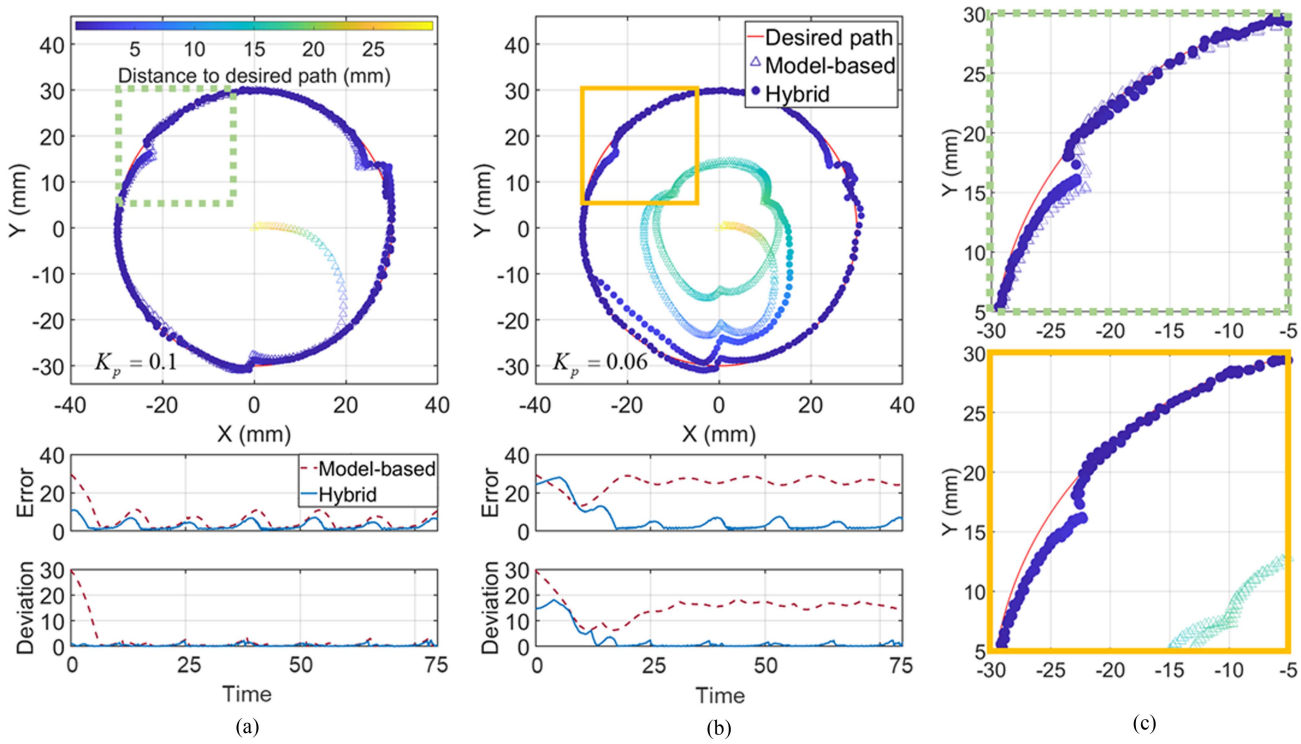


Fig. 10. Comparison of control performance tracking a circle path, where $K_p$ were set as (a) 0.1 and (b) 0.06. Subfigures are arranged in the same way as in Fig. 9.

TABLE I
MEAN TRACKING ERRORS AND PATH DEVIATIONS IN *PENTAGRAM* TRACKING WITH FOUR DIFFERENT GAINS $K_p$

| $K_p$ | Tracking error (mm) | | | | | Path deviation (mm) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Model | | Hybrid | | **Improv.** | Model | | Hybrid | | **Improv.** |
| 0.04 | 16.942 ± 5.421 | 564.2% | 2.305 ± 3.301 | 76.8% | **86.4%** | 3.646 ± 1.917 | 18.9% | 0.696 ± 0.832 | 6.9% | **80.9%** |
| 0.06 | 9.351 ± 3.664 | 311.2% | 1.461 ± 1.320 | 48.7% | **84.4%** | 2.252 ± 1.378 | 12.0% | 0.492 ± 0.477 | 2.6% | **78.2%** |
| 0.08 | 4.541 ± 2.486 | 150.9% | 1.050 ± 0.401 | 34.9% | **76.9%** | 1.461 ± 1.326 | 7.2% | 0.384 ± 0.3353 | 2.1% | **73.7%** |
| 0.10 | 2.986 ± 1.583 | 99.0% | 0.929 ± 0.414 | 30.9% | **68.9%** | 1.057 ± 1.032 | 5.2% | 0.361 ± 0.336 | 2.0% | **65.8%** |

The bold entities emphasize how much the hybrid controller can improve the tracking accuracy compared to conventional model-based controller.

for the tracking deviation under different values of $K_p$. The robot was actuated by model-based control at the beginning 100 or 75 s (2000 or 1500 steps, corresponding to two cycles), after which the initialization of GPR-based error compensator was finished and added. For the following steps, the CC-based model and GPR-based compensator in hybrid controller would work together to correct the tracking performance. To distinguish the errors of pure model-based control and hybrid control, we plotted these errors on the same time range. However, in the actual manipulation, they should be successively arranged as model-based control first (time 0–100 s, or 0–75 s) and then hybrid control (time 100–200 s or 75–150 s).

*1) Path With Sharp Angles:* The performance of tracking the moving target along the pentagram path is shown in Fig. 9. Gain $K_p$ was defined as 0.1, 0.08, 0.06, and 0.04, respectively. As the target would switch on every time step, the curve named with "Error" represents the distance between the current target and actual end-effector position. "Deviation" is calculated as the distance from the actual end-effector position to the closest point on the desired path. When $K_p$ was tuned as a proper value [see Fig. 9(a)], the pure model-based controller's tracking performance was roughly acceptable. However, for the cases that $K_p$ could not be tuned well [see Fig. 9(b)], the model-based method was not capable of adapting to the speed of moving target, i.e., the "Error" obviously increased. Mean tracking errors (i.e., step errors) and path deviations (e.g., the closest distances to the desired path) with different values of $K_p$ are listed in Table I, accompanied with the standard deviation (STD). Both the value and percentage of the errors are provided. The percentage for tracking error was obtained by averaging all tracking (error/step size), where the step size is 3 mm that for the path deviation was the average value of all (deviation/corresponding distance to the starting point). The column "Improv." indicates the percentage that hybrid controller outperforms model-based controller in the

TABLE II
MEAN TRACKING ERRORS AND PATH DEVIATIONS IN *CIRCLE* TRACKING WITH FOUR DIFFERENT GAINS $K_p$

| $K_p$ | Tracking error (mm) | | | | | Path deviation (mm) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Model | | Hybrid | | **Improv.** | Model | | Hybrid | | **Improv.** |
| 0.04 | 28.910 ± 3.330 | 967.1% | 2.864 ± 4.238 | 95.5% | **90.1%** | 22.043 ± 3.537 | 73.4% | 1.311 ± 4.073 | 4.3% | **94.1%** |
| 0.06 | 24.979 ± 3.688 | 832.0% | 6.132 ± 6.301 | 204.3% | **75.5%** | 15.101 ± 3.437 | 50.3% | 2.476 ± 2.688 | 8.2% | **83.6%** |
| 0.08 | 15.868 ± 2.486 | 528.3% | 4.655 ± 0.401 | 155.1% | **70.7%** | 6.211 ± 1.326 | 20.6% | 1.294 ± 0.335 | 4.3% | **79.2%** |
| 0.10 | 5.874 ± 3.217 | 195.2% | 2.803 ± 2.226 | 93.4% | **52.3%** | 2.087 ± 1.541 | 6.9% | 0.445 ± 0.449 | 1.4% | **78.7%** |

The bold entities emphasize how much the hybrid controller can improve the tracking accuracy compared to conventional model-based controller.

value or errors. Such improvements were calculated by $(e_m - e_h)/e_m$, where $e_m$ and $e_h$ are the tracking errors (or path deviation) using model-based controller and hybrid controller, respectively. For these four cases, the hybrid controller could improve the performance (68.9%–86.4% in tracking error, 65.8%–80.9% in path deviation) compared with the model-based one, even under the precondition that the tracking performance of mode was far from an acceptable standard.

*2) Smooth Path:* Similar to the pentagram path tracking, the performance when following a circle (see Fig. 10) is shown according to the same arrangement of Fig. 9. The difference with Section IV-D1 is that this path was constructed with smooth curves. When $K_p$ was not fine-tuned, the compensated part could steer the end-effector to approach the desired path quickly [see Fig. 10(b)]. Both the tracking error and the path deviation in tracing results can be largely reduced (52.3%–90.1% in the tracking error, 78.7%–94.1% in path deviation, Table II) under four values of $K_p$.

Both experiments in Section IV-D demonstrated that the hybrid control scheme enables the tracking convergence and gradually increased tracking accuracy without fine modeling tuning and data exploration. Taking the path deviation in pentagram tracking (see Table I) as an example, 0.1 was the optimal value of $K_p$ among the four values; however, the distinction of deviation was effectively reduced after using the hybrid controller (e.g., path deviation: 0.361–0.696 mm), greatly outperforming the model-based controller (1.057–3.646 mm). It could be noticed that there were deviations at the lower middle of the pentagram path (see Fig. 9(a) and (b), coordinate around [0, −10]) as well as on three areas with 120° interval in the circle path [see Fig. 10(a) and (b)], when $K_p$ was set as both 0.1 and 0.06. This kind of error results from the highly nonlinear mapping from stepper motor positions (i.e., air cylinder volumes) to elastic chamber elongations. The rough linearization (27) to correlate the motor command and chamber elongation could not meet the nonconstant change of factor $\alpha$. However, even with such an insufficiently tuned kinematics for control, the online-learning-based error compensator still enables enhancement of the tracking performance [see Fig. 10(c)].

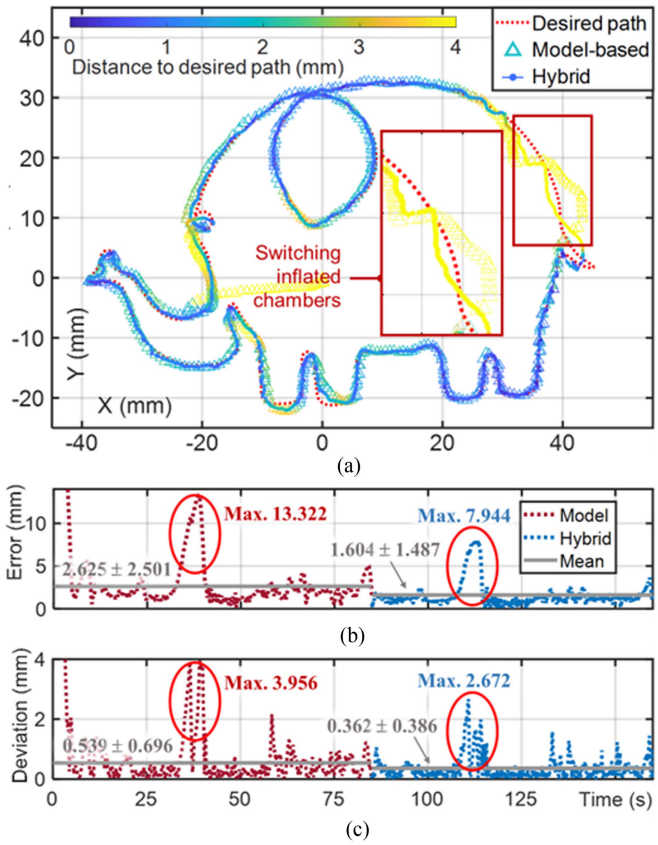


Fig. 11. Tracking performance along an elephant-shaped path. The sensing feedback was provided by visual-strain pose fusion. CC model-based control was conducted for the first-cycle tracking after which the learning-based error compensator was included. (a) Trajectories of the two cycles were plotted, as well as the (b) tracking error and the (c) path deviation with mean values indicated.

## E. Hybrid Control With Sensing Fusion

Experiments integrating the visual-strain fusion-based pose estimation and hybrid controller were also conducted. The robot end-effector was instructed to track a complicated closed path along an elephant-shaped path [see Fig. 11(a)]. The desired path involved 1500 target points, with approximately equal distances. For the pose estimation part, the ELM model was initialized after the first $N_0 = 450$ steps before which the SLAM estimation was acting as the sensing feedback for robot control. After initialization, the pose information would be provided by the fusion result of SLAM and FBG measurement. The ELM model was also updated online. For the closed-loop target tracking, model-based control was used for the first cycle (0–85 s, step 1–1700). Considering the initial end-effector position [0, 0] was not on the path, 200 more time steps after the 1500th step were included in the validation of model-based control to compensate for the initial approaching procedure. During this period, data

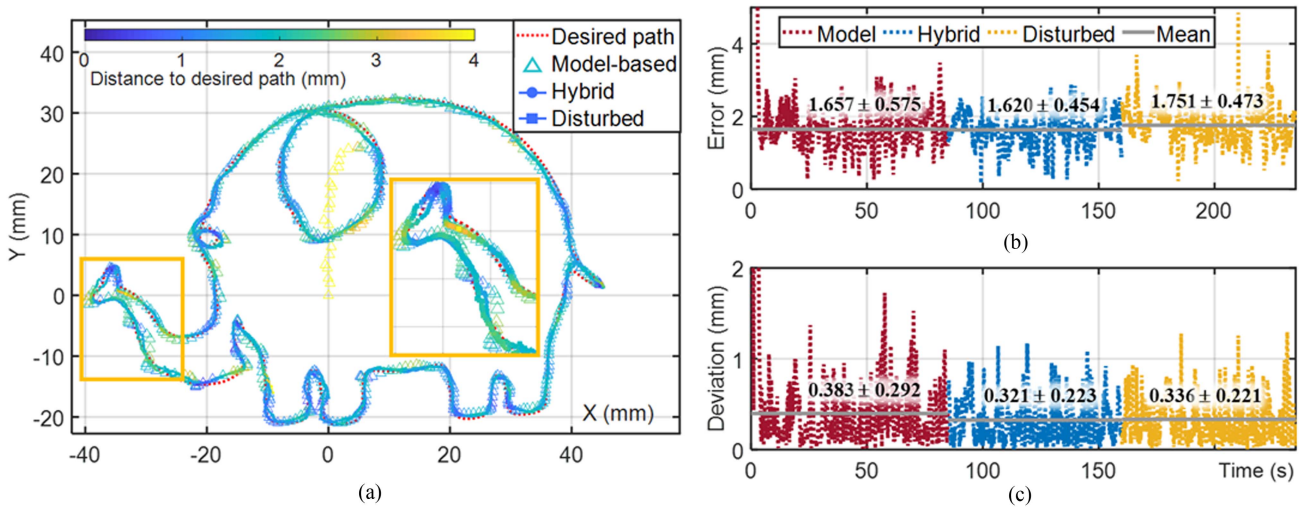


Fig. 12. Tracking performance with the same subfigure meanings as in Fig. 11. The nonlinear relationship between chamber elongation and motor actuation for each chamber was fitted using real-robot data. The third-cycle performance was added, where a moving obstacle created visual disturbances in the camera view.

collection and initialization of the GPR-based error compensator were also finished. Thus, in the second cycle (85–160 s, step 1701–3200), the robot was actuated utilizing hybrid control, which includes model-based control and the online-updated error compensator. The improvement using hybrid controller compared with the model-based controller can then be reflected.

*1) Using Linear Actuation–Elongation Mapping:* We tested the performance using the same hybrid controller as in Section IV-D ($K_p = 0.1$), the result of which is demonstrated in Fig. 11. The path of hybrid controller is marked by dense small filled circles, while that of the model-based method is marked by unfilled triangles. It could be seen that similar to the previous tests, the learning-based part in the hybrid controller can effectively compensate for most of the deviations of modeling uncertainties. However, when switching the inflated chambers [marked red in Fig. 11(a)], the hybrid controller still met difficulties in totally correcting the tracking trajectory to the desired path. Although, if we further fine-tune parameters in the model-based controller, it is feasible to bring a satisfactory tracking result with less path deviation. However, such performance is still valuable to be discussed, which shows that switching chambers is one of the main cases of concern for pneumatic-driven soft actuators, especially in the continuous-path-following tasks. In Fig. 11(b) and (c), the height difference of two controllers' error peaks [e.g., 13.322 mm and 7.944 mm, respectively, for model-based and hybrid controllers in Fig. 11(b)] as well as the mean values [2.625 ± 2.501 mm and 1.604 ± 1.487 mm in Fig. 11(b)] demonstrate that the error compensator can decrease the tracking error under various regions.

*2) Using Nonlinear-Fitted Actuation–Elongation Mapping:* In this experiment, the robot-specific mapping (27) was modified. Instead of a linear magnification, a nonlinear relationship between the chamber length and actuation motor was fitted using cubic spline data interpolation. With this specific-mapping correction, the control performance of model-based method could also be obviously increased with a fine-tuned $K_p$. The three deviations [see Fig. 12(a)] when switching inflated chambers could be eliminated, as no obvious error peaks are found [see Fig. 12(b) and (c)]. This time, we supplemented one more cycle (160–235 s, step 3201–4700), where the hybrid control continued being used, but moving obstacles were applied as visual disturbances in the camera view, similarly in Section IV-C1. In the last cycle with visual disturbances, the tracking error ($1.751 \pm 0.473$ mm) and path deviation ($0.336 \pm 0.221$ mm) [see Fig. 12(b) and (c)] could also maintain in a low level. These two experiments demonstrate that the fusion-based pose estimation could provide valid feedback for the hybrid controller or other controllers.

### F. Effects of Physical Collisions

Considering that the general purposes of soft robot use involved physical interaction with surrounding objects, we intend to investigate how the proposed fused sensing scheme, even under the interaction disturbance, can encounter deteriorated feedback either from visual or strain sensors. As a result, the synergetic use of both sensing approaches would give rise to the overall control performance. We intentionally made the robot even *more susceptible* to the contact interaction such that the FBGs wrapped on the robot cylindrical surface (without outer layer protection) would reflect the disturbance to the measured robot configuration as to which the proper visual sensing is expected to compensate this FBG sensing disturbance in order to maintain the control performance. Besides the LEGO-constructed setting, an abdominal simulator was built using swine viscera, acting as the surrounding for the camera in these experiments.

*1) Slow Push on the Robot:* Statistical results (in the two setups) are summarized in Table III. When the force was applied on the rigid tip [see Fig. 13(a)], its effect on the pose estimation can roughly be compensated by the fusion method (Fig. 13(a) ①, mean error: 1.216 mm). The appearance and removal of

TABLE III
MEAN, MAXIMUM (MAX.), AND STD OF ESTIMATION ERROR (MM) WHEN SLOWLY PUSHING THE ROBOT ON ITS RIGID CAP [SEE FIG. 13(A)] OR SOFT BODY WITH HELICALLY WRAPPED FIBER [SEE FIG. 13(B)]

| Scenes | Errors (mm) | On *rigid cap* | | | | | | On *soft body* with helically wrapped fiber | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Case ① | | | Case ② | | | Case ① | | | Case ② | | |
| | | SLAM | F-emp | Fusion | SLAM | F-emp | Fusion | SLAM | F-emp | Fusion | SLAM | F-emp | Fusion |
| Abdominal simulator | Mean | 1.209 | 1.218 | 1.216 | 2.498 | 2.857 | 2.738 | 1.595 | 1.929 | 2.002 | 1.670 | 1.917 | 1.767 |
| | Max. | 16.109 | 5.256 | 3.180 | 5.747 | 6.546 | 6.483 | 4.445 | 9.850 | 7.039 | 7.970 | 7.745 | 7.240 |
| | STD | 1.194 | 0.926 | 0.644 | 1.097 | 1.188 | 1.212 | 0.751 | 1.712 | 1.527 | 1.317 | 1.377 | 1.347 |
| LEGO® | Mean | 0.689 | 0.950 | 0.747 | 1.003 | 1.346 | 1.054 | 0.516 | 0.732 | 0.703 | 0.489 | 0.698 | 0.634 |
| | Max. | 2.468 | 5.026 | 4.736 | 2.374 | 3.961 | 3.879 | 4.482 | 4.962 | 4.975 | 3.515 | 4.546 | 4.569 |
| | STD | 0.395 | 0.752 | 0.514 | 0.463 | 0.730 | 0.678 | 0.461 | 0.875 | 0.856 | 0.456 | 0.774 | 0.754 |

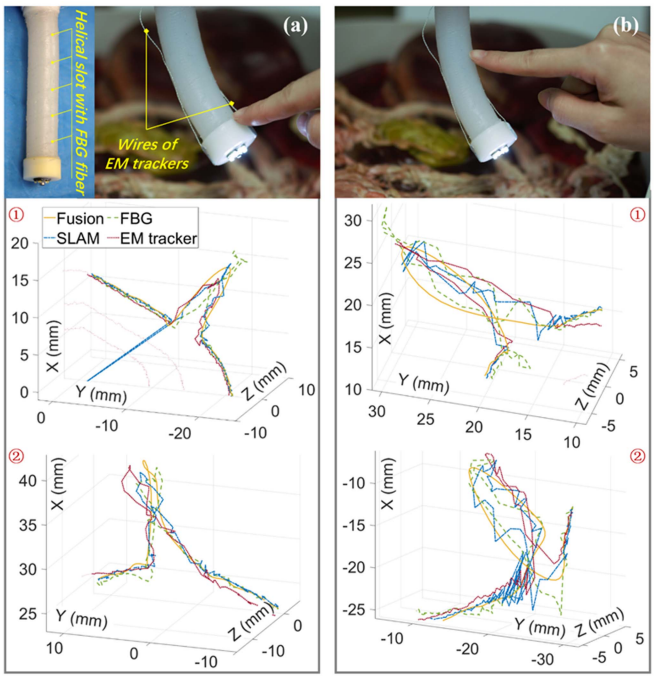


Fig. 13. Pose estimation results when pushing the robot slowly on (a) rigid cap or (b) soft body. The robot was actuated with the same sequential command as in Figs. 5–7. A pair of EM trackers acted as the ground truth (and their connection wires are marked in (a)). Undesired displacements (∼15 mm) were induced by the push. Two randomly selected time points during the path journey (marked as ① and ②) were shown as examples. End-effector trajectories (with four sensing settings) when the force was applied are depicted.

external forces would cause rapid change of robot moving directions. Such changes may result in large occasional errors of SLAM-based estimation (Fig. 13(a) ①, Max. 16.109 mm) but can be resolved by the fusion-based approach (Max. 3.180 mm). During the smooth and continuous pushing period, the difference between SLAM-, FBG-, and fusion-based estimations is not obvious (see Table III). Although sometimes the mean errors using the above three methods are slightly larger (Fig. 13(a) ②, ∼2.5 mm), the fusion result can also reflect the pattern of movement.

Contact tests were also conducted on the soft body [see Fig. 13(b)]. As the local contacts usually resulted in only a small portion of sparse FBGs being affected, the F-emp estimation accuracy did not deviate too much (e.g., Table III LEGO mean error 0.698–0.732 mm). Meanwhile, the image quality could be in a valid level for pose estimation (mean error 0.489–0.516 mm) and guaranteed the fusion result (mean error 0.634–0.703 mm). The fusion sensing performance can be comparable to the baseline that is the EM tracking directly on the robot end-effector (video 3:17–3:30). When the force was intentionally applied to the location wrapped with fiber, noise was observed from the F-emp pose estimations. This correlates with our hypothesis, i.e., if direct contacts exist between the fiber and surroundings, the measured FBG wavelength will involve external-force-caused strain variation and induce estimation error. One method to avoid this is to add a protective sheath/bellow externally in order to isolate or weaken the intensity of forces on fiber gratings. Such protection can be considered when applying the proposed sensor fusion method in specific applications.

*2) Fast Flick on the Robot:* To address more complicated circumstances involving both FBG and visual sensing disturbances, we conducted a set of finger flick tests in which we can observe how the F-emp estimation behaves when the vision-based estimation deteriorated by motion blur (see Tables IV and V). Different from the case with continuous pushing, the SLAM algorithm was unable to stably measure the motion since visual features were lost (video 3:30–3:39). Although the accuracy of the fusion-based estimation was slightly reduced relative to finger push, the fusion-based estimation was still stable (Table IV, error: 1.221 ± 0.860 to 1.769 ± 1.434 mm). Although, due to the robot hyperelasticity, a large deviation (e.g., 10.911 mm) would appear at the moment when an abrupt force was applied, our fusion-based estimation could gradually adapt to the vibration.

Table V took two extreme examples to discuss the limitation of this robot structure. They were chosen when the robot was executing in the abdominal simulator. It can be seen that the error is larger than in Table IV, particularly the error maximum, due to the increased force and corresponding bigger vibration amplitude. Although the fusion result occasionally fell from the optimal estimation, its mean error and STD were still close to the best one (e.g., fusion error 2.654 mm, min. error 2.275 mm). An interesting observation is that, here, the mean error of pushing on the rigid cap was even larger than on the body. One possible reason for this issue is the online-trained ELM model that did not involve enough related samples to accommodate such large range of motions. Aiming at fast model establishment, the

TABLE IV
MEAN, MAX., AND STD OF ESTIMATION ERROR WHEN *RAPIDLY FLICKING THE ROBOT* IN LEGO-CONSTRUCTED SCENES

| Scene | Errors (mm) | On *rigid cap* | | | | | | On *soft body* with helically wrapped fiber | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Case ① | | | Case ② | | | Case ① | | | Case ② | | |
| | | SLAM | F-emp | Fusion | SLAM | F-emp | Fusion | SLAM | F-emp | Fusion | SLAM | F-emp | Fusion |
| LEGO® | Mean | 1.249 | 1.324 | 1.221 | 1.644 | 1.689 | 1.486 | 1.799 | 1.849 | 1.666 | 1.849 | 2.175 | 1.769 |
| | Max. | 5.782 | 5.436 | 5.079 | 12.705 | 11.717 | 11.430 | 11.659 | 11.045 | 11.389 | 16.818 | 10.977 | 10.911 |
| | STD | 0.963 | 0.793 | 0.860 | 1.608 | 1.376 | 1.434 | 1.577 | 1.199 | 1.104 | 1.907 | 1.277 | 1.353 |

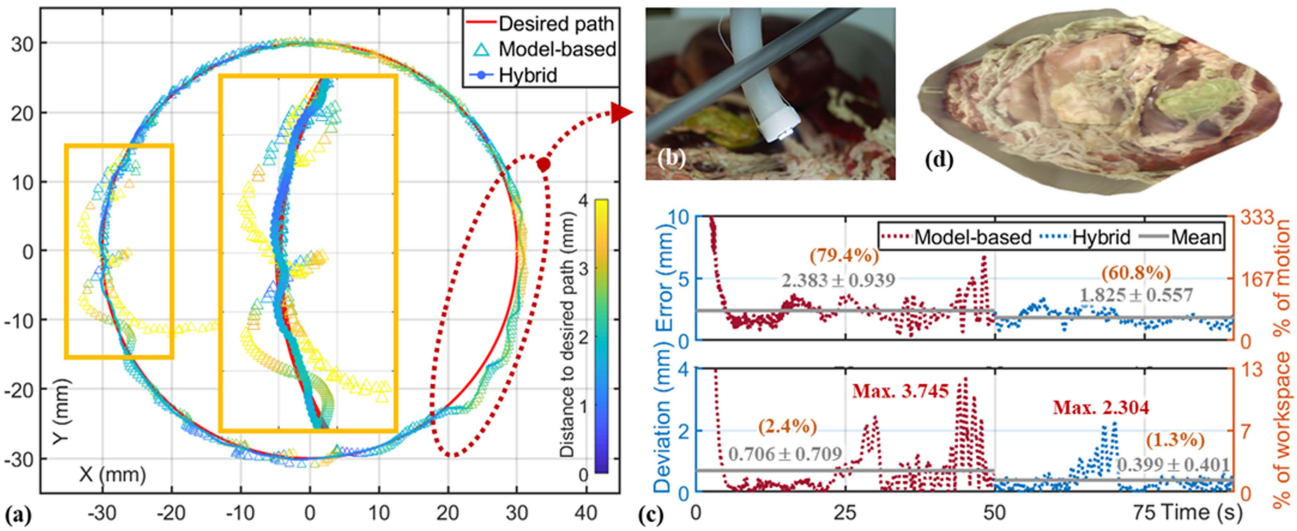


Fig. 14. Tracking performance in the swine-viscera-constructed abdominal simulator. (a) Trajectories of model-based (first cycle) and hybrid (second cycle) controllers are plotted, where a zoomed-in area (yellow contoured) shows the improved tracking accuracy of hybrid controller. Paths marked by an ellipse means the area that (b) fixed aluminum rod limited one-side robot bending. (c) Tracking error and path deviation. (d) Image-stitched figure of the abdominal simulator.

TABLE V
MEAN, MAX., AND STD OF ESTIMATION ERROR WHEN *RAPIDLY FLICKING THE ROBOT* IN ABDOMINAL SIMULATOR

| Error (mm) | On rigid cap | | | On soft body | | |
|---|---|---|---|---|---|---|
| | SLAM | F-emp | Fusion | SLAM | F-emp | Fusion |
| Mean | 4.330 | 4.984 | 5.130 | 2.886 | 2.275 | 2.654 |
| Max. | 48.849 | 31.456 | 38.479 | 19.081 | 9.833 | 25.975 |
| STD | 10.229 | 6.760 | 7.571 | 3.238 | 2.021 | 2.960 |

ELM models in our experiments were initialized online with a limited number of samples, which were in large probability of dense local distribution. Our model focused more on incremental local motions instead of cross-workspace movements. Such a mode was set for fast readiness of the F-emp prediction. The tradeoff between global accuracy and training samples should be adjusted if similar cases (large vibration) take a great proportion in the specific applications. In the *Appendix*, we discussed the robustness of the proposed sensing framework trained by various numbers of training samples to fast flick, where we can see that more and denser initialization samples will benefit to higher accuracy under physical collisions. The effect of protection on the FBG fiber was also simply tested.

*3) Fixed Obstacles in the Workspace:* Besides the validation on sensing, we also tested the controller's performance in the abdominal simulator. The same controller, as shown in Section IV-E2, was used. In Fig. 14(a), a zoomed-in area (yellow outline) demonstrated the improvement of our proposed controller again compared with the model-based controller. Noted that to extend the types of physical collision, an aluminum rod was fixed in the robot workspace, limiting the robot bending on one side [see Fig. 14(b)]. The paths marked by an ellipse indicate the area that the robot body was blocked by such a constraint. It can be observed that the actual paths of robot deviated from the desired path, even using the hybrid controller. This is due to the friction between the hyperelastic robot body and rod. As the target on the desired path switched for each control step, the correction effect of closed-loop controller could not totally compensate for the friction. However, the actual trajectory could keep convergent to the desired. Once the robot left the blocked area, the hybrid controller enables to bring the path back to the desired path in high accuracy again [mean error: 0.399 ± 0.401 mm, Fig. 14(c)], while when using pure model-based control, the robot motion could not return smooth. The image stitching of the abdominal-simulator scene [see Fig. 14(d)] clearly demonstrates the details of the scene.

## V. CONCLUSION

In this article, we proposed an integrated soft robot control system, integrating visual-strain fusion-based pose sensing and online-updated hybrid control. All the data-driven models used in the system could be conducted online, without prior data collection. Sparse strain measurements along a single-core FBG

TABLE VI
COMPARISON OF ESTIMATION ERRORS UNDER *RAPID FLICKS* WHEN MODELS INITIALIZED BY VARIOUS DENSITIES OF TRAINING DATA

| Errors (mm) | 450 samples for initialization | | | | | | 900 samples for initialization | | | | | | 1800 samples for initialization | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Case ① | | | Case ② | | | Case ① | | | Case ② | | | Case ① | | | Case ② | | |
| | SLAM | F-emp | Fusion | SLAM | F-emp | Fusion | SLAM | F-emp | Fusion | SLAM | F-emp | Fusion | SLAM | F-emp | Fusion | SLAM | F-emp | Fusion |
| Mean | 1.490 | 1.972 | 1.497 | 1.434 | 1.582 | 1.422 | 1.224 | 1.939 | 1.177 | 1.212 | 1.179 | 1.121 | 0.715 | 0.615 | 0.712 | 0.890 | 0.899 | 0.808 |
| Max. | 13.709 | 12.346 | 11.502 | 5.470 | 5.414 | 5.331 | 8.323 | 7.455 | 6.280 | 4.289 | 3.276 | 3.430 | 3.156 | 1.137 | 1.113 | 3.522 | 2.732 | 2.629 |
| STD | 2.393 | 2.405 | 1.727 | 0.902 | 0.795 | 0.673 | 1.401 | 1.100 | 1.150 | 0.567 | 0.477 | 0.375 | 0.363 | 0.164 | 0.173 | 0.713 | 0.232 | 0.316 |

fiber wrapped on the robot were trained online as a pose sensor. SLAM estimation using the monocular camera on the robot end-effector was used for the FBG sensor training. The fusion result of SLAM and FBG was able to provide robust feedback of the end-effector pose and accomplish 6-D image stitching. Sensing accuracy and continuity under extreme visual conditions, such as moving obstacles and varying lighting conditions, were resolved, even when encountering full shielding or absolute darkness. The sensing fusion proved immune to failures in SLAM caused by poor feature quality in images. The mean estimation error could be increased and stabilized from RMSE 3.116 to 1.324 mm. For the control scheme, the hybrid controller combining model-based kinematics and learning-based error compensator enabled steady control in target tracking tasks. The learning-based compensator in the hybrid controller reduced the tracking error by >80%. This controller can relax the requirement on modeling accuracy and effectively accommodate unmodeled nonlinearity.

The proposed framework integrated visual-strain fusion sensing modality and hybrid controller could be extended to other robot designs, including multisegment prototypes, although, in this article, we only validate it on the single-segment continuum robot. The application of single-core FBG fiber was not limited by the number of segments, as long as the adjacent segments were connected by a continuous joint that was smooth for wrapping the fiber. However, for the hybrid controller, the kinematics model should be changed or modified according to the specific manipulator used. The learning-based error compensator had the potential to be implemented by the same means as in Section III-B and enhanced the model-based control performance, if the feedback variable and actuation command could be collected and trained.

It is worth noting that our learning-based FBG model incorporates sparse FBGs to predict the robot pose based on its configuration, which has been proved capable of adapting to common local contacts. The further advanced multicore fiber using OFDR technique can even *eliminate* the local/global interaction effect on a similar pose/configuration estimation, such as impulsive or continuous interaction-induced deformation. Examples can be found in bronchoscopy (e.g., ion endoluminal system, Intuitive Surgical, Inc.) and catheterization platforms [22], [23], [43]. The FBG sensing could still be robust against pulsatile liquid flow or sudden contact with the surrounding. In light of the increasing use of FBGs in soft robotics, we can foresee the syngenetic and practical value of using both camera image and FBG strain data as the closed-loop control feedback.

In the aspect of the proposed algorithm, the combination of vision and FBG strain sensing can be further explored. The FBG fiber can be calibrated offline as a position or orientation sensing device and integrated with a monocular camera to compose a visual-FBG soft-robot SLAM framework, similar to the VINS SLAM [15]. Well-calibrated FBGs can take on the role of IMUs in a new enhanced visual SLAM system, therefore recovering the metric scale to enlarge their usage in soft robotic applications. FBGs could resolve challenges in processes, such as estimator initialization, extrinsic calibration, online loop detection, and tightly coupled relocalization, thus generating a new SLAM architecture for continuum robots. This visual-FBG SLAM system would have a great potential to be used in endoscopic robot localization, navigation, and control.

## APPENDIX

As a summarized analysis of Section IV-F1 and F2, the possible factors that affect the sensing accuracy under collisions could be categorized into two types, namely, algorithmic and physical factors. The hypothesis is that more samples for F-emp model training and external protection for FBG fiber will increase the robustness to external contacts. We conducted two simple tests as straightforward validations. First, experiment to test the effect of training samples' number and density under fast flick on the robot was conducted. As mentioned in Section IV-C, the number of samples for ELM model initialization was set as $N_0 = 450$ throughout this article. It is hypothesized that when the number of samples increases with an incremental density, the ELM model can be trained to deal with abrupt and irregular collisions more effectively. Therefore, $N_0 = 450$, 900, and 1800 were used separately under the same scene condition (see Table VI). Their corresponding distribution densities were also constant to, double, and quadruple the original density, respectively. After initialization, the samples for model's incremental training also maintained such densities. It can be found that the model initialized by 1800 samples and updated accordingly has the minimum estimation error ($0.712 \pm 0.173$–$0.808 \pm 0.316$ mm) among the three settings. The maximum error also has a decreasing trend when the density of samples increases. This is consistent with our hypothesis. Note that the time for model initialization and updating in each step is not obviously affected, showing that our proposed model is able to improve the robustness to collisions by increasing the number of samples.

To test FBG fiber's susceptibility to contact, we compared the wavelength shift with and without a silicone sheet cover (∼2-mm thickness) for protection (see Table VII). The unit of wavelength shift is nanometer (nm). The average range of

TABLE VII
PROPORTION OF NOISE IN WAVELENGTH SHIFT

| Error | Bare fiber | | With Silicone cover | |
|---|---|---|---|---|
| | Shift (nm) | % | Shift (nm) | % |
| Static | 0.0040 | 2.53 | 0.0040 | 2.53 |
| Straight | 0.0765 | 48.42 | 0.0387 | 24.49 |
| | 0.0910 | 57.59 | 0.0319 | 20.19 |
| Curve 1 | 0.0922 | 58.35 | 0.0151 | 9.56 |
| | 0.0911 | 57.66 | 0.0145 | 9.18 |
| Curve 2 | 0.0729 | 46.14 | 0.0189 | 11.96 |
| | 0.0903 | 57.15 | 0.0202 | 12.78 |

absolute wavelength shift during the spiral sensing test was 0–0.158 nm (mean 0.024 nm). The noise in wavelength shift under static status (without external force applied) was measured as max. 0.004 nm (2.53% of the max. wavelength change 0.158 nm; the following brackets indicate the same meaning) and mean $9.07\times10^{-4}$ nm (0.57%). The fiber was placed in straight, C-shaped, and S-shaped grooves separately, where the maximum wavelength shift under finger push ($\sim$2 N) was recorded. In Table VII, the "%" represents the proportion of such maximum shift in the sensing range 0.158 nm. It can be seen that the silicone protection can effectively isolate external forces on the fiber, with the sensing noise on wavelength shift reduced from 46.14%–58.35% to 9.18%–24.49%. Other fiber integration methods (e.g., placing a multicore FBG fiber in the inner channel of a continuum robot) will also facilitate the improvement of sensing accuracy under collisions.

## REFERENCES


[1] B. S. Homberg, R. K. Katzschmann, M. R. Dogar, and D. Rus, "Robust proprioceptive grasping with a soft robot hand," *Auton. Robots,* vol. 43, no. 3, pp. 681–696, 2019.

[2] P. Hyatt, D. Kraus, V. Sherrod, L. Rupert, N. Day, and M. D. Killpack, "Configuration estimation for accurate position control of large-scale soft robots," *IEEE/ASME Trans. Mechatronics,* vol. 24, no. 1, pp. 88–99, Feb. 2019.

[3] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu, "Relative camera pose estimation using convolutional neural networks," in *Proc. Int. Conf. Adv. Concepts Intell. Vis. Syst.*, 2017, pp. 675–687.

[4] M. Maidi, F. Ababsa, and M. Mallem, "Vision-inertial tracking system for robust fiducials registration in augmented reality," in *Proc. IEEE Symp. Comput. Intell. Multimedia Signal Vis. Process.*, 2009, pp. 83–90.

[5] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua, "Worldwide pose estimation using 3D point clouds," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 147–163.

[6] T. Nöll, A. Pagani, and D. Stricker, "Markerless camera pose estimation—An overview," in *Proc. Visual. Large Unstructured Data Sets- Appl. Geospatial Planning, Model. Eng.*, 2010, pp. 45–54.

[7] C.-L. Shih and Y. Lee, "A simple robotic eye-in-hand camera positioning and alignment control method based on parallelogram features," *Robotics,* vol. 7, no. 2, 2018, Art. no. 31.

[8] B. H. Yoshimi and P. K. Allen, "Active, uncalibrated visual servoing," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1994, vol. 1, pp. 156–161.

[9] G. Flandin, F. Chaumette, and E. Marchand, "Eye-in-hand/eye-to-hand cooperation for visual servoing," in *Proc. IEEE Int. Conf. Robot. Autom. Symp.*, 2000, vol. 3, pp. 2741–2746.

[10] P. Gemeiner, P. Einramhof, and M. Vincze, "Simultaneous motion and structure estimation by fusion of inertial and vision data," *Int. J. Robot. Res.,* vol. 26, no. 6, pp. 591–605, 2007.

[11] J. R. Rambach, A. Tewari, A. Pagani, and D. Stricker, "Learning to fuse: A deep learning approach to visual-inertial camera pose estimation," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2016, pp. 71–76.

[12] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-DOF camera relocalization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2938–2946.

[13] K. Eckenhoff, P. Geneva, and G. Huang, "Sensor-failure-resilient multi-IMU visual-inertial navigation," in *Proc. Int. Conf. Robot. Autom.*, 2019, pp. 3542–3548.

[14] F. M. Mirzaei and S. I. Roumeliotis, "A Kalman filter-based algorithm for IMU-camera calibration: Observability analysis and performance evaluation," *IEEE Trans. Robot.,* vol. 24, no. 5, pp. 1143–1156, Oct. 2008.

[15] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.,* vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[16] S. C. Ryu and P. E. Dupont, "FBG-based shape sensing tubes for continuum robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 3531–3537.

[17] H. Liu et al., "Shape tracking of a dexterous continuum manipulator utilizing two large deflection shape sensors," *IEEE Sensors J.,* vol. 15, no. 10, pp. 5494–5503, Oct. 2015.

[18] W. Zhuang, G. Sun, H. Li, X. Lou, M. Dong, and L. Zhu, "FBG based shape sensing of a silicone octopus tentacle model for soft robotics," *Optik*, vol. 165, pp. 7–15, 2018.

[19] R. J. Roesthuis, M. Kemp, J. J. van den Dobbelsteen, and S. Misra, "Three-dimensional needle shape reconstruction using an array of fiber Bragg grating sensors," *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 4, pp. 1115–1126, Aug. 2014.

[20] R. Seifabadi, E. E. Gomez, F. Aalamifar, G. Fichtinger, and I. Iordachita, "Real-time tracking of a bevel-tip needle with varying insertion depth: Toward teleoperated MRI-guided needle steering," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 469–476.

[21] C. Shi et al., "Shape sensing techniques for continuum robots in minimally invasive surgery: A survey," *IEEE Trans. Biomed. Eng.,* vol. 64, no. 8, pp. 1665–1678, Aug. 2017.

[22] J. Reisenauer et al., "Ion: Technology and techniques for shape-sensing robotic-assisted bronchoscopy," *Ann. Thoracic Surg.*, vol. 113, no. 1, pp. 308–315, 2022.

[23] X. T. Ha et al., "Robust catheter tracking by fusing electromagnetic tracking, fiber Bragg grating and sparse fluoroscopic images," *IEEE Sensors J.,* vol. 21, no. 20, pp. 23422–23434, Oct. 2021.

[24] R. Xu, A. Yurkewich, and R. V. Patel, "Curvature, torsion, and force sensing in continuum robots using helically wrapped FBG sensors," *IEEE Robot. Autom. Lett.,* vol. 1, no. 2, pp. 1052–1059, Jul. 2016.

[25] S. Sefati, R. Hegeman, F. Alambeigi, I. Iordachita, and M. Armand, "FBG-based position estimation of highly deformable continuum manipulators: Model-dependent vs. data-driven approaches," in *Proc. Int. Symp. Med. Robot.*, 2019, pp. 1–6.

[26] P. Saccomandi et al., "Feedforward neural network for force coding of an MRI-compatible tactile sensor array based on fiber Bragg grating," *J. Sensors*, vol. 2015, 2015, Art. no. 367194.

[27] T. L. T. Lun, K. Wang, J. D. L. Ho, K.-H. Lee, K. Y. Sze, and K.-W. Kwok, "Real-time surface shape sensing for soft and flexible structures using fiber Bragg gratings," *IEEE Robot. Autom. Lett.,* vol. 4, no. 2, pp. 1454–1461, Apr. 2019.

[28] X. Wang et al., "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, Apr. 2020.

[29] F. Alambeigi et al., "SCADE: Simultaneous sensor calibration and deformation estimation of FBG-equipped unmodeled continuum manipulators," *IEEE Trans. Robot.,* vol. 36, no. 1, pp. 222–239, Feb. 2020.

[30] K.-H. Lee et al., "FEM-based soft robotic control framework for intracavitary navigation," in *Proc. IEEE Int. Conf. Real-Time Comput. Robot.*, 2017, pp. 11–16.

[31] J. D. L. Ho et al., "Localized online learning-based control of a soft redundant manipulator under variable loading," *Adv. Robot.,* vol. 32, no. 21, pp. 1168–1183, 2018.

[32] X. Wang, Y. Li, and K.-W. Kwok, "A survey for machine learning-based control of continuum robots," *Front. Robot. AI*, vol. 8, 2021, Art. no. 280.

[33] K.-H. Lee et al., "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.

[34] G. Fang et al., "Vision-based online learning kinematic control for soft robots using local Gaussian process regression," *IEEE Robot. Autom. Lett.,* vol. 4, no. 2, pp. 1194–1201, Apr. 2019.

[35] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source slam system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.,* vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[36] G.-B. Huang, N.-Y. Liang, H.-J. Rong, P. Saratchandran, and N. Sundararajan, "On-line sequential extreme learning machine," in *Proc. IASTED Int. Conf. Comput. Intell.*, 2005, pp. 232–237.
[37] Z. Tian, G. Wang, Y. Ren, S. Li, and Y. Wang, "An adaptive online sequential extreme learning machine for short-term wind speed prediction based on improved artificial bee colony algorithm," *Neural Netw. World,* vol. 28, no. 3, pp. 191–212, 2018.
[38] T. Li, "Research on soft-sensing methods for the size of melt pool in MgO single crystal furnace," Ph.D. dissertation, Sch. Elect. Eng., Fac. Electron. Inf. Elect. Eng., Dalian Univ. Technol., Dalian, China, 2013.
[39] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: A new learning scheme of feedforward neural networks," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2004, vol. 2, pp. 985–990.
[40] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern., B*, vol. 42, no. 2, pp. 513–529, Apr. 2012.
[41] R. F. Reinhart, Z. Shareef, and J. J. Steil, "Hybrid analytical and data-driven modeling for feed-forward robot control," *Sensors,* vol. 17, no. 2, 2017, Art. no. 311.
[42] H.-C. Fu et al., "Interfacing soft and hard: A spring reinforced actuator," *Soft Robot.,* vol. 7, no. 1, pp. 44–58, 2020.
[43] Z. Dong et al., "Shape tracking and feedback control of cardiac catheter using MRI-guided robotic platform—Validation with pulmonary vein isolation simulator in MRI," *IEEE Trans. Robot.*, vol. 38, no. 5, pp. 2781–2798, Oct. 2022.

**Xiaomei Wang** (Member, IEEE) received the B.E. degree in automation from the Harbin Institute of Technology, Harbin, China, in 2014, the M.E. degree in control science and engineering from Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China, in 2016, and the Ph.D. degree in robotics from The University of Hong Kong, Hong Kong, in 2020.

She is currently a Postdoctoral Fellow with the Department of Mechanical Engineering, University of Hong Kong, and the Multi-Scale Medical Robotics Center Limited, Hong Kong. Her research interests include learning-based robot control and sensing, surgical robotics, and continuum robot control.

Dr. Wang serves as an Associate Editor for *ICRA* 2022–2023 and *RoboSoft* 2022–2023, and an Area Chair for IPCAI 2022.

**Jing Dai** (Student Member, IEEE) received the B.Eng. degree in naval architecture and ocean engineering from the Wuhan University of Technology, Wuhan, China, in 2017, and the M.S. degree in mechanical engineering from the Hong Kong University of Science and Technology, Hong Kong, in 2019. He is currently working toward the Ph.D. degree in robotics with the University of Hong Kong, Hong Kong.

His research interests include magnetic resonance imaging-guided robotics system, medical image analysis, and therapeutic ultrasound system.

**Hon-Sing Tong** received the B.Eng. degree in engineering science and the M.Phil. degree in mechanical engineering from the University of Hong Kong, Hong Kong, in 2019 and 2022, respectively.

His research interests include surgical navigation, augmented reality-assisted endoscopic surgery, and minimally invasive surgical robotics.

**Kui Wang** received the B.E. degree in automation and the M.E. degree in control science and engineering from Nankai University, Tianjin, China, in 2013 and 2016, respectively, and the Ph.D. degree in robotics from The University of Hong Kong, Hong Kong, in 2022.

His research interests include flexible shape sensing, soft robotic system, and data-driven robot control.

**Ge Fang** received the bachelor's degree in mechanical engineering and automation from the Huazhong University of Science and Technology, Wuhan, China, in 2014, and the master's degree in mechanical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2017, and the Ph.D. degree in robotics from The University of Hong Kong, Hong Kong, in 2021.

His research interests include magnetic resonance imaging-guided robotics system, soft robotics, and learning-based robot control.

**Xiaochen Xie** (Member, IEEE) received the B.E. degree in automation and the M.E. degree in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2012 and 2014, respectively, and the Ph.D. degree in control engineering from The University of Hong Kong, Hong Kong, in 2018.

She is currently a Full Professor with the Department of Automation, Harbin Institute of Technology, Shenzhen, China. She worked as a Postdoctoral Fellow with the Department of Mechanical Engineering, The University of Hong Kong, from 2018 to 2021. Her research interests include robust synthesis, periodic systems, intelligent systems, cyber–physical systems, robotics, and process monitoring.

Dr. Xie was awarded the Hong Kong Ph.D. Fellowship in 2014. She is an Associate Editor for *IET Control Theory and Applications* and IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN, AND CYBERNETICS.

**Yun-Hui Liu** (Fellow, IEEE) received the B.Eng. degree in applied dynamics from the Beijing Institute of Technlogy, Beijing, China, the M.Eng. degree in mechanical engineering from Osaka University, Osaka, Japan, and the Ph.D. degree in mathematics engineering and information physics from the University of Tokyo, Tokyo, Japan.

After working with the Electrotechnical Laboratory of Japan as a Research Scientist, he joined The Chinese University of Hong Kong in 1995 and is currently Choh-Ming Li Professor of mechanical and automation engineering and the Director of the T Stone Robotics Institute. He also serves as the Director/CEO of Hong Kong Centre for Logistics Robotics sponsored by the InnoHK programme of the HKSAR government. He has authored or coauthored more than 500 papers in refereed journals and refereed conference proceedings and was listed in the Highly Cited Authors (Engineering) by Thomson Reuters in 2013. His research interests include visual servoing, logistics robotics, medical robotics, multifingered grasping, mobile robots, and machine intelligence.

Dr. Liu was a recipient of numerous research awards from international journals and international conferences in robotics and automation and government agencies. He was the Editor-in-Chief for *Robotics and Biomimetics* and served as an Associate Editor for the IEEE TRANSACTION ON ROBOTICS AND AUTOMATION and General Chair of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems.

**Kwok Wai Samuel Au** received the B.Eng. and M.Phil. degrees in mechanical and automation engineering from the Chinese University of Hong Kong (CUHK), Hong Kong, in 1997 and 1999, respectively, and the Ph.D. degree in mechanical engineering from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2007.

He is currently a Professor with the Department of Mechanical and Automation Engineering and the Department of Surgery (by courtesy), CUHK, and the Founding Director of Multiscale Medical Robotics Center, InnoHK, Hong Kong. In September 2019, he found Cornerstone Robotics and has been serving as the President of the company, aiming to create affordable surgical robotic solution. Before joining CUHK in 2016, he was the Manager of Systems Analysis of the New Product Development Department, Intuitive Surgical, Inc. At Intuitive Surgical, he coinvented and was leading the software and control algorithm development for the FDA cleared da Vinci Si Single-Site surgical platform in 2012, Single-Site Wristed Needle Driver in 2014, and da Vinci Xi Single-Site surgical platform in 2016. He was also a founding team member for the early development of Intuitive Surgical's FDA cleared robot-assisted catheter system, da Vinci ION system, from 2008 to 2012. He coauthored more than 60 peer-reviewed manuscripts and conference journals, 17 granted U.S. patents/EP, and 3 pending U.S. patents.

Dr. Au was a recipient of numerous awards, including the first prize in the American Society of Mechanical Engineers Student Mechanism Design Competition in 2007, Intuitive Surgical Problem Solving Award in 2010, and Intuitive Surgical Inventor Award in 2011.

**Ka-Wai Kwok** (Senior Member, IEEE) received the B.Eng. and M.Phil. degrees in automation and computer-aided engineering from the Chinese University of Hong Kong, Hong Kong, in 2003 and 2005, respectively, and the Ph.D. degree in computing from the Hamlyn Center for Robotic Surgery, Department of Computing, Imperial College London, London, U.K., in 2012.

He is currently an Associate Professor with the Department of Mechanical Engineering, University of Hong Kong (HKU), Hong Kong. Prior to joining HKU in 2014, he worked as a Postdoctoral Fellow with Imperial College London in 2012 for surgical robotics research. In 2013, he was awarded the Croucher Foundation Fellowship, which supported his research jointly supervised by advisors from the University of Georgia, Athens, GA, USA, and Brigham and Women's Hospital, Harvard Medical School, Boston, MA, USA. His research focuses on surgical robotics, intraoperative image processing, and their uses of intelligent and control systems. To date, he has coauthored more than 135 publications with $>80$ clinical fellows and $>150$ scientists/engineers, and 6 out of 14 invention patents licensed/transferred to industrial partners in support for their commercialization.

Dr. Kwok multidisciplinary work has been recognized by more than ten international publication awards, mostly under IEEE, particularly in the largest flagship conferences of robotics: e.g., ICRA Best Conference Paper Award in 2018 and IROS Toshio Fukuda Young Professional Award in 2020. He also serves as an Associate Editor for the *Journal of Systems and Control Engineering*, IEEE ROBOTICS AND AUTOMATION MAGAZINE, and *Annals of Biomedical Engineering*. He is the Principal Investigator of research group for Interventional Robotic and Imaging Systems, HKU.